



HAL
open science

Classification Hiérarchique Implicative et Cohésitive selon la mesure MGK - Application en didactique de l'informatique

Hery Frédéric Rakotomalala

► **To cite this version:**

Hery Frédéric Rakotomalala. Classification Hiérarchique Implicative et Cohésitive selon la mesure MGK - Application en didactique de l'informatique. Education. Université d'Antananarivo, 2019. Français. NNT: . tel-02172222

HAL Id: tel-02172222

<https://theses.hal.science/tel-02172222v1>

Submitted on 3 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNIVERSITÉ D'ANTANANARIVO

ÉCOLE DOCTORALE PROBLÉMATIQUES DE L'ÉDUCATION ET DIDACTIQUES DES DISCIPLINES

Équipe d'Accueil Doctorale en Éducation et Didactiques des Mathématiques et de l'Informatique

THÈSE de DOCTORAT

pour l'obtention du grade de

Docteur de l'Université d'Antananarivo

Domaine : « Sciences de l'Éducation »

Thématique : « Problématiques de l'Éducation et Didactiques des Disciplines »

Spécialité : « Didactiques des Mathématiques et de l'Informatique »

par

Hery Frédéric RAKOTOMALALA

Classification Hiérarchique Implicative et Cohésitive selon la mesure M_{GK} - Application en didactique de l'informatique

soutenue publiquement le 30 avril 2019

Membres du Jury

M. Rémi RAKOTONDRAONA,	Maître de Conférences, HDR	Université d'Antananarivo	Président
M. Paul Auguste RANDRIAMITANTSOA,	Professeur Titulaire	Université d'Antananarivo	Rapporteur externe
M. Dominique TOURNÈS,	Professeur des Universités	Université de la Réunion	Rapporteur interne
M. Josvah Paul RAZAFIMANDIMBY,	Professeur Titulaire	Université de Fianarantsoa	Examinateur
M. Feno Daniel RAJAONASY,	Maître de Conférences	Université de Toamasina	Examinateur
M. Jean DIATTA,	Professeur des Universités	Université de la Réunion	Co-Directeur
M. André TOTOHASINA,	Professeur Titulaire	Université d'Antsiranana	Directeur

Remerciements

Je tiens à remercier en premier lieu MM. André Totohasina et Jean Diatta de m'avoir donné l'opportunité de la réalisation de ma thèse sous leur direction, dans l'équipe d'accueil «Éducation et Didactiques des Mathématiques et de l'Informatique d'Antsiranana» de l'École doctorale PEDDI. Grâce à leur encouragement, leur disponibilité, leurs idées, leurs conseils, leur sympathie et leur solide expérience de la recherche, m'ont beaucoup aidé à développer mes aptitudes pour la recherche. Je leur en suis très reconnaissant.

J'adresse aussi mes très sincères remerciements à M. Rémi Rakotondrondona pour l'honneur qui m'a été accordé en présidant le jury. J'exprime ensuite ma plus profonde gratitude à MM. Paul Auguste Randriamitantsoa et Dominique Tournès qui ont accepté de rapporter cette thèse. Je tiens à remercier également MM. Josvah Razafimandimby et Feno Daniel Rajaonasy d'avoir examiné la thèse, et j'ai beaucoup apprécié leur participation au jury de cette thèse.

Mes remerciements vont également à mes trois enfants Njato, Manda et Mihanta, de leur tolérance pendant mes absences fréquentes et régulières pour la réalisation de ce travail. Ils m'ont donné la force de tenir jusqu'au bout.

Enfin, je remercie plus particulièrement ma très chère femme Georgine Rasoanjanahary pour son dévouement, sa patience, et sa confiance durant ces longues années.

Table des matières

Introduction générale	1
État de l'art	9
1 Science des données et Big Data, Processus d'ECD et Data Mining	11
1.1 Introduction	12
1.2 La science des données et Big data	13
1.3 Le processus d'ECD	15
1.3.1 Etapes d'un processus ECD	16
1.4 Les généralités sur le data mining	22
1.4.1 Les méthodes descriptives	22
1.4.2 Les méthodes prédictives	24
1.4.3 Les techniques non supervisées	25
1.4.4 Les techniques supervisées	25
1.5 Conclusion	25
2 Structure de treillis et famille de Moore	27
2.1 Introduction	28
2.2 Historiques	28
2.3 Une introduction aux ensembles ordonnés et aux treillis	29
2.3.1 Les notions mathématiques d'ordre et d'ensemble ordonné	29
2.3.2 La notion mathématique de treillis	31
2.3.3 Treillis et éléments irréductibles	36
2.3.4 Table d'un treillis	38
2.4 Treillis des fermés, système de fermetures	41
2.4.1 Treillis des fermés	42
2.4.2 Système de fermeture	43
2.5 Correspondances de Galois	44
2.5.1 Applications galoisiennes	44
2.5.2 Correspondances de Galois associées à une relation binaire	45
2.6 Système implicatif	47
2.7 Hiérarchies	49
2.7.1 Chaîne de partitions et arbre de classification associée	51
2.7.2 Construction d'un dendrogramme ordinal compatible avec $\mathcal{H}_b(S)$	53

2.7.3	Hierarchie de fourches, hierarchie binaire de parties et arbre binaire de classification	56
2.7.4	Hierarchie de fourches orientees et hierarchie implicative	58
2.7.5	Ultrametricite	60
2.8	Conclusion	64
3	Règle d'association et mesure d'intérêt	65
3.1	Introduction	66
3.2	Modélisation et quelques pratiques pionnières	66
3.2.1	Modélisation mathématique	67
3.2.2	Quelques définitions	68
3.3	Extraction des règles d'association	70
3.3.1	Quelques algorithmes d'extraction des règles d'association	70
3.3.2	Algorithme Apriori	71
3.3.3	Autres algorithmes	72
3.3.4	Test du χ^2	76
3.4	Mesure de qualité des règles d'association	78
3.4.1	Critères de qualité d'une mesure des règles d'association	80
3.4.2	Quelques mesures de qualité	86
3.5	Conclusion	90
 Contributions : Extraction des règles d'association M_{GK}-valides et Classification Hiérarchique Implicative et Cohésitive selon M_{GK} (CHIC-M_{GK})		91
4	Extraction des règles d'association M_{GK}-valides	93
4.1	Introduction	94
4.2	l'Analyse Statistique Implicative	95
4.2.1	Quasi-règle implicative selon l'approche classique de R. Gras	95
4.2.2	Intensité d'implication	98
4.3	Mesure de qualité des règles d'association M_{GK}	100
4.3.1	Construction de la mesure M_{GK}	101
4.3.2	Propriétés de la mesure de qualité des règles d'association M_{GK}	102
4.3.3	Commentaire et plaidoyer sur M_{GK}	109
4.4	Extraction des règles d'association M_{GK} -valides avec contribution du <i>Support</i>	110
4.4.1	Implication entre variables binaires	111
4.4.2	Prise en compte des valeurs critiques de M_{GK}	113
4.4.3	Apport par considération du <i>Support</i> de la règle d'association valide en fonction de M_{GK}	122
4.4.4	Normalisation de la mesure de qualité $supp_{M_{GK}}$	124
4.4.5	Vers un nouvel indice de cohésion implicative selon M_{GK}	133
4.5	Conclusion	142

5	Nouvelle Classification Hiérarchique Implicative et Cohésitive basée sur M_{GK}	143
5.1	Introduction	144
5.2	Construction statistique d'une hiérarchie implicative	144
5.2.1	Implication entre classes de variables	145
5.2.2	Construction de l'arbre hiérarchique et la détermination des niveaux significatifs des classifications	147
5.3	Contributions des individus	154
5.3.1	Paire orientée générique principale associée à une classe	154
5.3.2	Vecteur puissance implicative d'une classe \underline{C}_H	155
5.3.3	Puissance implicative d'un individu et distance sur la classe \underline{C}_H	156
5.4	Conclusion	159
	 Application en didactique de l'informatique : didactique de l'algorithmique	 161
6	Pédagogie, didactique, didactique des disciplines	163
6.1	Pédagogie	164
6.1.1	Pédagogie : une théorie éducative	165
6.1.2	Pédagogie : une pratique éducative	165
6.1.3	Évolution des conceptions pédagogiques	166
6.2	Didactique	169
6.2.1	Didactique générale	169
6.2.2	Le triangle didactique	169
6.2.3	Limites du triangle didactique	170
6.2.4	Trois dimensions de la recherche didactique	171
6.3	Didactique des disciplines	173
6.3.1	Le triangle didactique par rapport à l'introduction des TIC	174
6.3.2	Apport des théories constructivistes en psychologie à la didactique	179
6.3.3	Notion et concept d'activité	187
6.3.4	Modèle adopté dans le système d'activités	197
7	La technologie de l'éducation et les TIC	199
7.1	Introduction	200
7.2	La technologie de l'éducation	200
7.2.1	Technologie	200
7.2.2	Technologie éducative	201
7.2.3	Technologie de l'éducation	202
7.3	Les Technologies de l'Information et de la Communication	205
7.4	Étude sur les stratégies à mettre en œuvre pour l'intégration effective des TIC dans la pratique pédagogique de la région DIANA	206
7.4.1	Matériels et méthodes	208
7.4.2	Résultats et discussions	211
7.4.3	Stratégie à mettre en œuvre	223
7.4.4	Conclusion partielle	224

7.5	Analyse et diagnostic sur la connaissance en informatique des étudiants de l'enseignement supérieur à Madagascar	224
7.5.1	Matériels et méthodes utilisés pour la fouille des données	225
7.5.2	Interprétation de quelques règles	232
7.6	Conclusion	233
8	Didactique de l'informatique/algorithmique - Formation en informatique	235
8.1	Introduction	236
8.2	Didactique de l'informatique	236
8.2.1	Qu'est ce que l'informatique ?	237
8.2.2	La pensée informatique	238
8.3	Formation en informatique dans le modèle anglo-saxon	249
8.3.1	Le modèle anglo-saxon	249
8.3.2	Analyse des thèmes dans CE, CS, IS, IT et SE	251
8.3.3	Analyse des thèmes en CS	256
8.3.4	Analyse des capacités d'aptitudes des étudiants en CE, CS, IS, IT et SE	263
8.3.5	Analyse des capacités d'aptitudes des étudiants en CS	273
8.3.6	Domaines de connaissances du modèle anglosaxon	286
8.3.7	Principes de base du développement logiciel (SDF)	290
8.4	Formation en informatique à l'ENSET	294
8.4.1	Les capacités d'aptitudes des jeunes en licence informatique de l'ENSET	294
8.4.2	Les parcours de licence en informatique	295
8.4.3	Analyse du curriculum informatique à l'ENSET	296
8.5	Sur la programmation	301
8.5.1	Qu'est-ce que la programmation ?	301
8.5.2	Comparaison du domaine de connaissances SDF en CS à celle de l'ENSET	303
8.6	Sur l'algorithme	304
8.6.1	Historique	304
8.6.2	Qu'est-ce que l'algorithme ?	305
8.6.3	La pensée algorithmique	308
8.6.4	Procédure de résolution d'un problème	309
8.6.5	Qu'est-ce que l'algorithmique ?	310
8.7	Curriculum informatique en classe de Seconde du lycée à Madagascar	311
8.7.1	Le cours sur les généralités sur les systèmes informatiques	314
8.7.2	Le cours sur les applications logiciels	315
8.7.3	Le cours en algorithmique et programmation	316
8.8	Conclusion	320
9	Scratch dans l'apprentissage de l'algorithmique en classe de Seconde à Madagascar	321
9.1	Introduction	322
9.2	Choix et utilisation de Scratch	322
9.2.1	L'espace de travail et environnement Scratch	323

9.2.2	La programmation en Scratch	324
9.2.3	Liste des blocs utilisés dans Scratch	325
9.3	Expérimentations	325
9.4	Le déroulement de séances en classe	326
9.4.1	La première séance : Prise de contact	326
9.4.2	La deuxième séance : Activité de découverte de l’algorithme et de la programmation	329
9.4.3	La troisième séance : Activité de programmation - Scratch	340
9.4.4	La quatrième séance : Évaluation finale	348
9.4.5	Analyses des résultats d’évaluation	349
9.5	Conclusion	355
Conclusion générale et perspectives		357
Annexes		397
A Algorithmes d’extraction des RA		397
B Listing des codes sources et résultats		407
B.1	Programme CHIC- M_{GK}	408
B.1.1	Chargement de données	409
B.1.2	Informations sur les variables	410
B.1.3	Croisement 2 à 2 de couples de variables	410
B.1.4	Affichage des tableaux M_{GK} et cohésion à base de M_{GK}	411
B.1.5	Classification Hiérarchique et Cohésitive selon M_{GK}	411
B.2	Classe principale	413
B.3	Classe utilisant pour les calculs en fonction de couple de variables	464
B.4	Classe utilisant pour les calculs occurrence, moyenne et ecart-type	468
B.5	Classe de la classification	469
B.6	Classe pour tracer le dendrogramme	471
B.7	Classe pour dessiner les points du dendrogramme	474
B.8	Classe pour tracer les noeuds du dendrogramme	475
B.9	Organisation des individus	478
B.10	Calcul concernant les individus	480
B.11	Résultats 1 : §4.4.1, tab 4.3, tab 4.4	482
B.12	Résultats 2 : §4.4.2	483
B.13	Résultats 3 : §5.2.2	484
B.14	Résultats 4 : §5.3	485
C Résultats intégration TIC par la méthode AFM		487
C.1	Questionnaire	488
C.2	valeurs propres et corrélation de groupe de variables	490
C.3	Les tables de rapport Classe/Modalité et Valeur-test	491
D Résultats d’évaluation des classes de seconde en algorithmique		493
D.1	Notes des étudiants en langues et maths-info	494

Liste des tableaux

2.1	Table de treillis de la figure 2.4	40
2.2	Table binaire extrait du tableau 2.1	41
2.3	Contexte formael	47
3.1	Exemple de tableau d'un contexte binaire.	68
3.2	Base de données transactionnelles.	69
3.3	Achat de thé et de café	76
3.4	Tableau de contingence de X et Y	78
3.5	Tableaux de contingence associés.	79
3.6	Tableau de probabilités conjointes associée.	80
4.1	Tableau d'exemple pour calculer l'intensité d'implication de la quasi-règle $X \rightarrow Y$	99
4.2	Tableau d'un contexte binaire.	112
4.3	Tableau de contingence.	113
4.4	Extrait de tableaux de contingence et quelques valeurs données par le χ^2 , la <i>confiance</i> , la <i>similarité</i> et la <i>correlation</i> des couples de variables.B.11-TB.1	113
4.5	Exemples comparatifs	114
4.6	Effectifs observés	115
4.7	Effectifs théoriques	115
4.8	Tableau représentant les valeurs de M_{GK} -valides de chaque couple de variables.	122
4.9	Tableau représentant les valeurs de $supp_{M_{GK}}$ de couples de variables.	123
4.10	Tableau représentant $supp_{M_{GK}}$ -normalisées : $supp_{(n)M_{GK}}$	130
4.11	Tableau représentant les cohésions de couples de variables selon $supp_{M_{GK}}$	136
4.12	Tableau représentant les cohésions de couples de variables selon $supp_{(n)M_{GK}}$	138
4.13	Tableau représentant les valeurs de cohésion de couple de variables calculées par le logiciel CHIC de Gras.	140
5.1	Tableau représentant l'occurrence de chaque variable	149
5.2	Tableau représentant les valeurs des PO triées	150
5.3	Tableau montrant les classes constituées, $coh_{supp_{(n)M_{GK}}}(\underline{C})$ et $\psi(\underline{C}, \underline{C}')$	151

5.4	Tableau représentant de l'indice centré et réduit observé (<i>niveau significatif</i>) et l'indice de statistique globale (<i>nœud significatif</i>) de chaque niveau de classification.	154
5.5	Tableau représentant le respect de l'individu x , noté Φ_x vis-à-vis de la classe formée par la paire générique $((v_i, v_j))$	156
7.1	Extrait du questionnaire C.1: partie qui caractérise les individus enquêtés	209
7.2	Répartition des questions selon les parties	209
7.3	Ensemble de k -tableaux ayant en commun n individus	210
7.4	Répartition des questions par rapport aux groupes de variables	211
7.5	Extrait des valeurs propres et pourcentages de variance (ou d'inertie)	212
7.6	Coordonnées de groupes sur les axes	213
7.7	Contributions des groupes sur les 2 axes factoriels	213
7.8	Cosinus carrés des groupes sur les 2 axes factoriels	214
7.9	Indicateur de similarité \mathcal{L}_g	215
7.10	Indicateur RV	215
7.11	Répartition des individus dans la classe	218
7.12	Extrait du tableau binaire de l'enquête sur la connaissance en informatique des étudiants	225
7.13	Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau pour la connaissance en informatique des étudiants.	227
7.14	Information sur les variables	231
8.1	Tableau comparatif des poids des thèmes informatiques dans les cinq types de programmes d'études, tiré du (Shackelford et al., 2005, p.24)	251
8.2	Tableau binaire extrait du tableau 8.1 pour les disciplines CE,CS, IS, IT et SE	252
8.3	Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau.	252
8.4	Les 15 méta-règles concernant les 40 thèmes par rapport aux cinq programmes d'études CE, CS, IS, IT et SE	253
8.5	Tableau binaire extrait du tableau 8.1 pour les disciplines CS	257
8.6	Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau pour CS.	257
8.7	Les 25 méta-règles concernant les 40 thèmes par rapport au programme d'études CS	259
8.8	Capacités d'aptitude des diplômés en informatique par discipline, tiré du (Shackelford et al., 2005, p.28)	265
8.9	Tableau binaire extrait du tableau 8.8 pour les capacités d'aptitudes des étudiants poursuivant les parcours CE,CS, IS, IT et SE	266
8.10	Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau pour les capacités d'aptitudes.	267

8.11	Les 24 méta-règles par rapport aux capacités d'aptitudes des diplômés en informatique	269
8.12	Les 53 méta-règles par rapport aux capacités d'aptitudes des diplômés en CS.	283
8.13	Le parcours licence en informatique EADIMI	296
8.14	Unités d'Enseignements de licence en informatique EADIMI	297
8.15	Les 25 méta-règles par rapport aux UE du parcours informatique de l'ENSET	300
8.16	Tableau représentant l'équivalence de SDF par rapport à l'UE de l'ENSET	303
8.17	Généralités sur les systèmes informatiques	315
8.18	Applications logiciels	316
8.19	Généralités sur l'Algorithmique	317
8.20	Les opérations de lecture et d'écriture	318
8.21	Structures de contrôle conditionnel	318
8.22	Les structures itératives	319
8.23	Les tableaux	319
9.1	Les commandes dans la zone de menu	325
9.2	Langue et dialecte pratiqués dans les deux classes	326
9.3	Caractéristiques des réponses des élèves	327
9.4	Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau pour la question 2 §9.4.1	328
9.5	Sujet d'évaluation	349
9.6	Tableau montrant les valeurs de cohésions des couples de variables formées pour la classe de seconde I	350
9.7	Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau de la classe de seconde I.	350
9.8	Tableau montrant les valeurs de cohésions des couples de variables formées pour la classe de seconde II	353
9.9	Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau de la classe de seconde II.	353
9.10	Moyenne générale des classes en matières littéraires et scientifiques	355
B.1	Tableaux de contingence et les valeurs données par le χ^2 , la <i>confiance</i> , la <i>similarité</i> et la <i>correlation</i> des couples de variables.	482
B.2	Tableau représentant les caractéristiques de toutes les règles positives et négatives du contexte 4.2 donné	483
B.3	Tableau représentant les caractéristiques des règles M_{GK} -valides . .	484
B.4	Tableau des Paires Orientées triées	484
B.5	Tableau montrant les classes constituées, $coh_{supp(n)M_{GK}}(\underline{C})$ et $\psi(\underline{C}, \underline{C}')$	484
B.6	Contributions des individus	485
C.1	Corrélations de groupes de variables sur les deux axes factoriels . .	490

C.2	Tableau représentant les valeurs propres et pourcentages de variance (inertie)	490
C.3	Tableau représentant les valeurs de Cla/Mod, p.value et v.test des variables constituant la classe 1+ à gauche et classe 1- à droite . .	491
C.4	Tableau représentant les valeurs de Cla/Mod, p.value et v.test des variables constituant la classe 2+ à gauche et classe 2- à droite . .	491
C.5	Tableau représentant les valeurs de Cla/Mod, p.value et v.test des variables constituant la classe 3+ à gauche et classe 3- à droite . .	492
D.1	Résultats de malagasy, français, anglais, mathématiques et informatique de la classe de seconde I	494
D.2	Table binaire de la classe de seconde I, tirée du tableau D.1	495
D.3	Résultats de malagasy, français, anglais, mathématiques et informatique de la classe de seconde II	496
D.4	Table binaire de la classe de seconde II, tirée du tableau D.3	497

Liste des figures

1	Plan de la thèse	8
1.1	Diagramme de Venn de la data science	13
1.2	Représentation des données massives	14
1.3	Nettoyage de données	18
1.4	Intégration de données	19
1.5	Reduction de données	19
1.6	Transformation de données	20
1.7	Les étapes d'un processus ECD, (<i>un extrait emprunté à P. A. P. Ruiz (2014, p. 48)</i>).	21
2.1	Trois représentations possibles de la relation binaire \mathcal{R}	31
2.2	Treillis, entre désordre et ordre	33
2.3	Exemple d'ordre	34
2.4	Exemple de treillis	34
2.5	Exemple de treillis distributif	35
2.6	Exemple de treillis booléen	35
2.7	Treillis de la figure 2.4 pour définir les ensembles J_x et M_x	37
2.8	Treillis de la figure 2.4 où sont précisés, pour chaque élément x , les ensembles J_x et M_x	38
2.9	Diagramme de Hasse du treillis de la figure 2.4	41
2.10	Deux familles ordonnées par inclusion	42
2.11	Dendrogramme ordonné	50
2.12	Arbre de classification ordinalement indicé	52
2.13	Fourche non orientée	56
2.14	Fourche orientée	59
2.15	Deux formes implicatives	64
4.1	Représentation par les diagrammes d'Euler	97
4.2	Algorigramme d'extraction des règles M_{GK} -valides	120
4.3	Situations de référence	124
4.4	Courbe représentative des $supp_{M_{GK}}$ et $supp_{(n)M_{GK}}$ en fonction des couples de variables formant les règles valides à un seuil fixé au risque d'erreur $\alpha = 10\%$	131
4.5	Courbes représentatives des $supp_{M_{GK}}$ et $supp_{(n)M_{GK}}$ en fonction des couples de variables formant les règles valides à un seuil fixé au risque d'erreur $\alpha = 10\%$ comparées avec les confiances respectives	132
4.6	Courbe d'entropie : cas de 2 variables formant une classe (v_i, v_j)	135

4.7	Courbes représentatives des $coh_{supp_{M_{GK}}}$ et $coh_{supp_{n_{M_{GK}}}}$ et $conf$ des règles valides	139
4.8	Courbes représentatives des $coh_{supp_{M_{GK}}}$ et $coh_{supp_{n_{M_{GK}}}}$ et $conf$ des règles valides	141
5.1	Dendrogramme représentant les méta-règles du tableau 5.3	154
6.1	Triangle pédagogique de Houssaye	167
6.2	Didactique générale	170
6.3	Triangle didactique de Chevillard	171
6.4	Les trois axes de recherche en didactique	172
6.5	Le triangle didacTIC	177
6.6	Le tétraèdre de Lombard	178
6.7	Le tétraèdre pédagogique de Faerber	179
6.8	Pentaèdre didactique	184
6.9	Les éléments d'un système d'activité dans le milieu scolaire	189
6.10	Structure de base de l'activité humaine d'Engeström (1987)	192
6.11	La contradiction primaire dans chaque constituante du système d'activité, (<i>un extrait emprunté à C Bonneau p.27</i>)	196
6.12	Le modèle didactique associé à l'activité algorithmique en classe de seconde	198
6.13	Le modèle réduit associé à l'activité algorithmique en classe de seconde	198
7.1	Accès et utilisation TIC des élèves/enseignants dans les Lycées/U-niversités	208
7.2	Caractéristiques et graphes respectifs des personnes enquêtées	211
7.3	Eboulis des valeurs propres	212
7.4	Représentation de groupes de variables	216
7.5	Représentation des individus et variables sur le plan factoriel	216
7.6	Représentation des individus sur le plan factoriel	216
7.7	Représentation de groupes de variables sur le plan factoriel	217
7.8	Représentation des variables quantitatives sur le plan factoriel	217
7.9	Classification des individus	219
7.10	Représentation de la classification des individus sur le plan factoriel en 3D	220
7.11	Représentation de la classification des individus sur le plan factoriel en 2D	220
7.12	3 Graphes représentant les nœuds significatifs des 78 classes	230
7.13	Extrait du dendrogramme représentant les méta-règles obtenues.	233
8.1	La pensée informatique	241
8.2	Graphe représentant les nœuds significatifs des 15 classes	254
8.3	Extrait du dendrogramme représentant des 15 classes	254
8.4	Graphe représentant les nœuds significatifs des 25 classes	260
8.5	Extrait du dendrogramme représentant des 25 classes	260
8.6	Graphe représentant les nœuds significatifs des 24 classes	269

8.7	Extrait du dendrogramme représentant des 24 classes	270
8.8	Euclide d'Alexandrie III ^{ime} siècle avant JC.	305
8.9	Muhammad ibn Musa al-Khwarizmi IX ^{ime} siècle.	305
9.1	Fenêtre principale Scratch	324
9.2	Graphe représentant les nœuds significatifs des 5 classes de question 2 §9.4.1	328
9.3	élève 1 - activité 1	332
9.4	élève 2 - activité 1	332
9.5	élève 3 - activité 1	333
9.6	élève 4 - activité 1	333
9.7	Symboles utilisés dans l'algorigramme	334
9.8	Algorigramme	334
9.9	Séquence d'instructions	334
9.10	Recette de cuisine et programme	335
9.11	élève 1 - activité 2	338
9.12	élève 2 - activité 2	339
9.13	élève 3 - activité 2	339
9.14	Structure alternative complète	341
9.15	Structure de choix	342
9.16	Boucle tant que... faire	343
9.17	Boucle répéter... jusqu'à	343
9.18	Boucle répéter... jusqu'à	344
9.19	Boucles imbriquées	345
9.20	Graphe représentant les nœuds significatifs des 5 classes de la seconde I	351
9.21	Extrait du dendrogramme représentant des 5 classes de la seconde I	352
9.22	Graphe représentant les nœuds significatifs des 7 classes de la seconde II	354
9.23	Extrait du dendrogramme représentant des 7 classes de la seconde II	354
B.1	Fenêtre d'interface principale	408
B.2	Chargement des données externes au format .csv	409
B.3	Récupération et affichage des données externes	409
B.4	Déscriptions des variables	410
B.5	Croisement 2 à 2 de couples de variables	411
B.6	Affichage des résultats de M_{GK} et cohésion- M_{GK}	411
B.7	CHIC- M_{GK}	412
C.1	Modèle de questionnaire pour l'étude sur l'intégration du TIC au Lycée	489

Liste des algorithmes

1	Algorithme de Génération de la table d'un treillis	40
2	Algorithme de recherche des motifs fréquents Apriori	71
3	Algorithme de Génération des motifs candidats Apriori-Gen	72
4	Algorithme d'extraction des règles d'association M_{GK} -valides	121
5	Algorithme AprioriTid	398
6	Algorithme SAMPLING	398
7	Algorithme Sampling-Gen	399
8	Algorithme Sampling-Count	399
9	Algorithme PARTITION	400
10	Algorithme Partition-Gen	400
11	Algorithme Partition-Count	401
12	Algorithme ECLAT	401
13	Algorithme FP-Growth(\mathcal{B} ,minsupp)	402
14	Algorithme FP-Tree(\mathcal{B} ,a,minsupp)	402
15	Algorithme Max-Miner	403
16	Algorithme Gen-Initial-Groups	403
17	Algorithme Gen-Sub-Nodes	404
18	Algorithme CLOSE	404
19	Algorithme Gen-Closure	405
20	Algorithme MaxCLQ(G, C, LB), un algorithme de séparation et évaluation de base pour Maxclique	405
21	Algorithme surestimation(G), une surestimation de la cardinalité d'une clique maximum de G	406

Listings

B.1	Classe <i>Fenetre_principale.java</i>	413
B.2	Classe <i>Couple_variable.java</i>	464
B.3	Classe <i>Variable.java</i>	468
B.4	Classe <i>Classification.java</i>	469
B.5	Classe <i>DendrogrammePanel.java</i>	471
B.6	Classe <i>Point.java</i>	474
B.7	Classe <i>Noeud.java</i>	475
B.8	Classe <i>Individu.java</i>	478
B.9	Classe <i>Couple_individu.java</i>	480

Liste des abréviations

AFM	: Analyse Factorielle Multiple
ASI	: Analyse statistique Implicative
CAH	: Classification Ascendante Hiérarchique
CE	: Computer Engineering
CHIC	: Classification Hiérarchique Implicative et Cohésitive
CHIC- M_{GK}	: Classification Hiérarchique Implicative et Cohésitive selon M_{GK}
CPIR	: Conditional Probability Incrementation Ratio
CS	: Computer Science
ECD	: Extraction de Connaissance à partir de Données
ENSET	: École Normale Supérieure pour l'Enseignement Technique
FD	: Fouille de Données
H	: Niveau Hiérarchique d'une classe
ION	: Implication Statistique Orientée Normée
IPT	: Informatique Pour Tous
IS	: Information Systems
IT	: Information Technology
M_{GK}	: Mesure de Guillaume-Kenchaff
μ	: Mesure probabiliste de qualité (MPQ)
PO	: Paire Orientée
RA	: Règle d'Association
SE	: Software Engineering
TIC	: Technologie de l'information et de la Communication

Introduction générale

“ La confiance n'exclut pas le contrôle... ”

Lénine

A Madagascar, la politique de l'État dans le domaine des TIC s'articule sur : le renforcement de l'infrastructure de liaison, le renforcement de l'accès pour tous aux TIC, le développement des TIC en tant que secteur d'activité économique à part entière, et enfin le développement des TIC en tant que secteur d'appui à l'éducation, la santé, la gouvernance, . . . Malgré ceci, les TIC font partie des outils émergents, car la vulgarisation de l'outil informatique, en particulier, les ordinateurs, ne se déroulait effectivement qu'en 2004, par la prise de décision de l'État pour la détaxation totale des achats et ventes de ceux-ci. Récemment, l'Internet à haut débit est opérationnel dans quelques zones de Madagascar. En effet, actuellement Telma (Télécom Malagasy) renforce le réseau en fibre optique dans la capitale et dans les autres chefs lieux de province par la mise en place du réseau national de fibre optique et la liaison internationale par fibre optique sous-marine.

Une étude au préalable a été déjà menée et présentée pendant le colloque international sur les TIC à Antananarivo en 2012, ([Rakotomalala, Totohasina, et Sambatra, 2012](#)) dans le but d'avoir une idée précise du niveau de connaissance, d'intégration et d'utilisation des TIC dans le secteur éducatif de la zone nord de Madagascar, plus particulièrement la Région DIANA. Nous avons entrepris une démarche d'études complète pouvant être généralisée aux cas des autres régions de Madagascar. Cette étude nous a permis de proposer des stratégies à mettre en œuvre pour l'intégration effective des TIC dans la pratique pédagogique de l'enseignement supérieur technique à Madagascar. Les résultats de cette étude sont alarmants en termes de

méconnaissance des TIC aussi bien au niveau des élèves/étudiants que des enseignants que nous détaillons dans cette thèse. Des actions ont été proposées, mais malgré une partie de ces solutions ont été entreprises par quelques établissements de la région en intégrant les TIC dans leur pratique pédagogique, il reste encore un travail de longue haleine pour les chercheurs en problématiques de didactique de **trouver les multitudes failles** constatées, comme l'inexistence de programme scolaire officiel en informatique et des manuels pédagogiques, le manque de formations des enseignants en informatiques, l'insuffisance des infrastructures informatiques et proposer petit à petit des solutions temporelles, car le monde aussi évolue : « *la solution d'hier peut devenir un problème d'aujourd'hui, la solution d'aujourd'hui pourrait poser des problèmes demain, ainsi de suite* ». Aussi, l'État (les décideurs du Pays) doit-il se tourner envers les chercheurs nationaux en didactique de l'informatique pour qu'ils déploient leur réflexion profonde et qu'ils fassent valoir leur esprit visionnaire professionnel et réaliste.

Un constat a été fait que le Ministère de l'Éducation Nationale ait au moins dix sept partenaires internationaux qui financent des projets dans quelques régions de Madagascar. Ces projets se concentrent surtout sur la dotation en matériels informatiques très dispersée (ordinateurs, tableaux numériques interactifs TNI, tablettes, ...), la confection des manuels scolaires en quantité très insuffisantes, la formation des enseignants et les dirigeants d'établissements sans programme, la mise en place des ressources énergétiques renouvelables en campagne (panneau solaire, éolienne,...) dans le but de vulgariser les TIC dans la pratique pédagogique de l'enseignement, mais pour le moment, seulement dans quelques établissements et régions ciblés. Mais ce qui est intrigant, et un grand problème comme nous avons sus-mentionné demeure que le programme de l'informatique ne soit figuré nul part dans les programmes scolaires nationaux du niveau primaire au lycée. En effet, la seule volonté d'intégrer des TIC dans le système éducatif à Madagascar est loin d'être suffisant pour assurer la véritable formation / initiation à l'informatique aux jeunes scolaires et adultes de demain. Des projets d'aide sociale aux élèves aussi sont mis en place en créant des cantines à l'école pour éviter l'abandon scolaire, mais ceci se concentre surtout dans les zones urbaines. En fait, des projets financés par les bailleurs de fonds sont nombreux, mais en général tous sont sectorisés et éparpillés, i.e. « *si on*

veut cuire du riz pour le peuple, une partie de la communauté au nord reçoit de grains de riz, une autre partie à l'ouest reçoit de la marmite, une autre partie au sud reçoit du réchaud et, enfin une autre partie à l'est reçoit de l'eau, ainsi le riz ne serait jamais cuit ».

D'autre part, face à la mondialisation et la globalisation depuis les années 1980, le plan « Informatique Pour Tous » IPT a été adopté. Mais jusqu'où doit-il aller ? Suffit-il de se contenter à manipuler l'ordinateur, ou bien apprendre des logiciels récemment parus ? C'est un investissement perdu, car avant que l'élève ne sorte du lycée, ces produits auront disparu du marché. Et quelle compétence en informatique, va-t-il faire face au niveau supérieur ? L'élève doit-il savoir quelque chose de l'ordinateur et de la façon dont on lui fait réaliser une tâche, ou peut-on considérer matériel et logiciel comme une boîte noire ? Quelle compétence est-elle demandée au professeur ? Ne doit-il pas savoir l'informatique ? Ne doit-il pas être compétent en programmation ? Ne doit-il pas maîtriser l'algorithmique et savoir ensuite l'enseigner ? Le débat est ouvert dicit [Arsac \(1988\)](#), dans son article : quelle informatique pour des futurs utilisateurs ? Quelle sensibilisation pour tous ? La réponse à toutes ces questions ne mène à rien s'il n'y pas de programme officiel d'informatique pour collégiens et lycéens. Voilà les problématiques didactique que nous abordons dans cette thèse.

L'ensemble des travaux synthétisés dans ce document s'insère principalement dans deux domaines de recherche différents et complémentaires : l'Extraction de Connaissances dans les Données, et cette problématique de didactique de l'informatique. L'objectif global est d'avoir un outil informatique basé sur un fondement théorique mathématique pour résoudre des problèmes didactiques.

La construction de connaissance à partir d'observation est une activité quotidienne et naturelle. « *Chaque chose que l'enfant observe, il imite en boucle (apprentissage) et lui forme déjà une connaissance* ». Mais ceci est très complexe, pour la combinaison des nouvelles observations aux connaissances déjà acquises de telle façon que l'ensemble forme un système structuré afin d'oublier les observations et ne garder que la connaissance à laquelle elles ont contribué. Si la connaissance produit les ef-

fets attendus, elle est *validée*, sinon elle est *réajustée*. L'Extraction de Connaissances dans les Données (ECD) est l'automatisation de ce processus et forme un domaine de recherche multidisciplinaire qui s'est appuyée à l'origine sur les bases de données, puis a rapidement nécessité une étroite coopération avec l'apprentissage automatique, les statistiques et l'analyse de données, la visualisation, et l'aide à la décision. L'ECD propose une méthodologie fondée sur un processus de transformation des données vers les connaissances qui se décompose en trois catégories telles que :

1. La localisation, la sélection et le prétraitement des données ;
2. La découverte des connaissances à l'aide des modèles prédictifs ou descriptifs, supervisés ou non supervisés ;
3. Le post-traitement des connaissances découvertes et leur validation par un décideur/expert des données.

Nos travaux de recherche se focalisent sur le point-2 à l'aide des modèles descriptifs du processus d'ECD, au plus proche du décideur. Notamment, sur le modèle des règles d'association (RA) qui permet de découvrir, sans connaissances préalables, des tendances implicatives au sein des données. Alors que les travaux de [Feno \(2007\)](#) se sont déjà attaqués efficacement au problème du nombre généralement pléthorique des règles d'association extraites en construisant une base (i.e. un ensemble minimal des règles valides par l'utilisation d'un ensemble d'axiomes d'inférence). Les travaux de [Ramanantsoa \(2016\)](#) y a déjà apporté quelques améliorations conceptuelles. Cependant, il reste encore un travail informatique important d'implémentation et de construction d'un logiciel opérationnel de fouille de données. Les travaux récents de [Totohasina \(2008\)](#) ont déjà proposé la théorie unificatrice des mesures de qualités normalisées et celles normalisables, i.e. des mesures d'implication statistique prenant ses valeurs dans l'intervalle $[-1, +1]$ et qui s'annulent en cas d'indépendance, valent $+1$ en cas d'implication logique et -1 en cas de situation d'incompatibilité, sont positives en cas d'attraction et négatives en cas de répulsion : de tels indices s'avèreraient plus pertinentes et donc meilleures pour extraire des associations orientées non symétriques et non redondantes répondant à des besoins de recherche de causes ou d'effets. Cette théorie règle déjà partiellement le problème majeur de la fouille des règles d'association dans un contexte binaire, à savoir la non-pertinence et la non-utilité de beaucoup de règles extraites.

Une autre théorie, l'Analyse Statistique Implicative (ASI) (Gras, 1979), qui est une méthode d'analyse des données issues du croisement d'une population d'individus et d'un ensemble de variables quelconques (booléennes, intervalles, floues, vectorielles, . . .). Cette méthode vise à attribuer une valeur d'intensité ou de qualité à des règles d'association (tendancielle ou implicative) du type : « *si la variable x est observée dans la population, alors la variable y a tendance à être également observée* ». Elle fournit pour cela un cadre statistique permettant de traiter à travers leur quantification, des relations causales entre phénomènes, qui s'est appuyé sur une mesure de probabilité l'intensité d'implication de l'in vraisemblance de la liaison entre les variables dans une hypothèse d'indépendance a priori de ces variables. Dans cette théorie, des graphes de règles (selon des chemins) et des hiérarchies orientées de méta-règles (selon des classes ordonnées) sont étudiées et qui conduisent à des interprétations en termes de concepts, typiques de structures comportementales cachées d'individus formant la population ayant participé à l'étude. De plus, la notion de contribution d'individus ou de descripteurs d'individus à un chemin ou à une classe de règles permet, en retour, de définir une structure topologique sur l'ensemble de la population, pour y « *dessiner une topologie du sens* » (Charlionet et al., 2005). L'ASI s'intéresse aussi dans ce cas à la recherche et aux représentations de la structure des connaissances extraites.

De notre point de vue, une solution réside dans l'hybridation de deux théories complémentaires : l'une par rapport aux deux travaux (Feno, 2007; Totohasina, 2008), sus-mentionnés pour l'extraction des règles valides et l'autre, (Gras, 1979) pour la représentation des connaissances :

1. Mesures de qualité : pour sélectionner les règles potentiellement intéressantes à l'aide de mesures adaptées (*interestingness measures*), et par l'élimination des redondances ;
2. Hiérarchies orientées : pour permettre de regrouper de nouveau les règles potentiellement intéressantes en méta-règles ;

La littérature sur l'extraction des connaissances à partir des données volumineuses délivre une variabilité des méthodes d'approche, notamment dans la fouille des règles d'association. Rappelons qu'une règle d'association traduit des relations généralement orientées entre certaines variables ou certains motifs (ensembles de variables)

d'une base de données. Notre objectif dans cette thèse est alors d'obtenir des règles de bonne qualité qui permettent de décrire des relations initialement inconnues, cachées dans la base des données, mais potentiellement utiles, et d'anticiper ainsi des comportements. L'adoption de la classification hiérarchique cohésitive et implicative selon la mesure de qualité M_{GK} , inspirée du CHIC de R. Gras s'avérait pertinente, et cela consiste à extraire un ensemble de formules logiques conditionnelles qui déduisent la valeur d'un ensemble d'attributs à partir des valeurs d'autres classes d'attributs apparaissant simultanément. La constitution des classes se décide de façon automatique et non supervisée selon la valeur de cohésion implicative évaluée à partir de M_{GK} cette fois. Elle fait preuve déjà d'une efficacité notoire dans la science de données, dans le domaine des bases de données relationnelles, en didactique des mathématiques et en bioinformatique. La communauté scientifique spécialisée en fouille des données relève l'inexistence de critère(s) qui serait (seraient) formellement meilleur(s) que d'autres. Le choix de critère dépendrait du contexte.

Cependant, il a été évoqué lors de la 47^{ème} Conférence Internationale de l'Éducation de l'UNESCO sur « l'Éducation de Qualité Pour Tous les Jeunes » EQPTJ qui s'est tenu à Genève en septembre 2004, pour que la qualité des processus éducatifs s'améliore, il faut avoir des enseignants suffisamment nombreux et compétents, (...) « *Les enseignants sont les piliers de la réforme. Je crois qu'il n'y a pas de réforme, que ce soit la qualité ou sur d'autres questions qui puisse être réussie sans une participation, sans un engagement des enseignants* », selon Elie Jouen, Secrétaire Général Adjoint, International de l'Éducation de l'époque. Dans la réalité, la situation est loin d'être satisfaisante. De nombreux pays, y compris Madagascar, souffrent d'une grave pénurie d'enseignants ou d'un manque d'enseignants qualifiés dans un certain nombre de disciplines scientifiques, notamment les mathématiques et l'informatique. La présence d'enseignants compétents et bien formés dans les classes se heurtent souvent à de nombreux obstacles « *Environnement Socio-Economico-Politico-Culturel* » (ESEPC) : bas salaires, statut social précaire, lourde charge de travail, effectifs pléthoriques, perspectives limitées d'évolution professionnelle, etc. Le bon enseignant : un homme-orchestre, un enseignant imbu de leadership pédagogique et de leadership transformationnel ? Celui qui parvient à guider les élèves dans le dédale des connaissances disponibles et réussit à leur montrer comment utiliser

l'information et communiquer.

Pendant même à cette conférence et appuyé par [Pelgrum et Law \(2004\)](#), il a été aussi reconnu que les « *Technologies de l'Information et de la Communication* » (TIC) peuvent contribuer à une multitude de questions liées à l'éducation : l'accès universel, l'équité, la pratique d'un apprentissage et d'un enseignement de qualité, le développement professionnel des enseignants et la mise en œuvre d'une gestion, d'une gouvernance et d'une administration de l'éducation plus efficaces. Les TIC constituent alors un outil d'apprentissage et d'enseignement de plus en plus important et efficace. Dans plusieurs littératures, des questions se posent : Comment faire apprendre des thèmes aux jeunes prédisposés à faire l'informatique ? Comment les motiver à l'apprendre ? *Une condition nécessaire*, et aussi comment les faire comprendre ? *Une condition suffisante*.

L'objectif de cette thèse est alors de créer un outil d'analyse de données permettant d'extraire de nouvelles connaissances à partir des grands volumes de données appliqué à la didactique de disciplines, plus particulièrement à la didactique de l'informatique au niveau du lycée. Il nous permet en effet de découvrir des liens implicatifs des variables qui nous paraissent utiles dans une étude donnée.

Cette thèse est composée de trois grandes parties : La première partie portant sur l'état de l'art est constituée de trois chapitres dont le chapitre 1, concerne un succinct aperçu sur la data science et le big data, ainsi que les notions de base, considérées utiles dans le domaine d'extraction de connaissances. Il comprend des différentes étapes du processus d'ECD et les généralités sur la fouille de données (data mining), suivi de chapitre 2 qui concerne les concepts mathématiques sur les notions de treillis et famille de Moore, utiles pour les méthodes d'extraction des règles d'association que nous abordons dans le chapitre 3, intitulé, règle d'association et mesure d'intérêt.

La deuxième partie est consacrée à nos contributions pour les nouvelles méthodes algorithmiques sur l'extraction des règles d'association M_{GK} -valides dans le chapitre 4, suivi d'une nouvelle classification hiérarchique implicative et cohésitive basée sur la mesure M_{GK} dans le chapitre 5. Ces études nous permettent de créer un nouvel outil d'analyses de données implémenté en langage de programmation Java.

La troisième partie qui est composée de 4 sections se focalise sur l'application en didactique de l'algorithmique de l'outil créé, dont le chapitre 6 parlera un aperçu global sur la théorie de la pédagogie, la didactique en général et de la didactique de l'informatique. Le chapitre 7 porte sur l'alphabétisation des acteurs (enseignants, élèves) par rapport à la technologie de l'éducation et les technologies de l'information et de la communication dans l'apprentissage de l'informatique. Le chapitre 8 oriente vers l'étude des contenus de la formation en informatique qui fait partie de la didactique de l'informatique/algorithmique. Enfin, le dernier chapitre 9 est consacré sur l'expérimentation de curriculum proposé par rapport à des activités simples pour l'apprentissage de la matière - algorithmique et programmation dans un lycée d'Antsiranana.

Classification Hiérarchique Implicative et Cohésitive selon la mesure M_{GK} - Application en didactique de l'informatique		
Introduction générale		
Etat de l'art		
Partie I	Chapitre 1	La Science des Données et Big Data, le processus d'ECD et les généralités sur le data mining
	Chapitre 2	Structure de treillis et famille de Moore
	Chapitre 3	Règle d'association et mesure d'intérêt
Contributions : Extraction des règles d'association M_{GK}-valides et Classification Hiérarchique Implicative et Cohésitive selon M_{GK}		
Partie II	Chapitre 4	Extraction des règles d'association M_{GK}-valides
	Chapitre 5	Nouvelle Classification Hiérarchique Implicative et Cohésitive basée sur M_{GK}
Application en didactique de l'informatique : didactique de l'algorithmique		
Partie III	Chapitre 6	Pédagogie, didactique, didactique des disciplines
	Chapitre 7	La technologie de l'éducation et les TIC
	Chapitre 8	Didactique de l'informatique/algorithmique - Formation en informatique
	Chapitre 9	Utilisation de Scratch dans l'apprentissage de l'algorithmique en classe de Seconde à Madagascar
Conclusion générale et perspectives		

Figure 1 – Plan de la thèse

Première partie

État de l'art

Chapitre 1

La Science des Données et Big Data, le processus d'ECD et les généralités sur le data mining

“ La connaissance s’acquiert par l’expérience, tout le reste n’est que de l’information. ”

Albert Einstein
Mathématicien, Physicien,
Scientifique (1879 - 1955)

Sommaire

1.1 Introduction	12
1.2 La science des données et Big data	13
1.3 Le processus d'ECD	15
1.3.1 Etapes d'un processus ECD	16
1.4 Les généralités sur le data mining	22
1.4.1 Les méthodes descriptives	22
1.4.2 Les méthodes prédictives	24
1.4.3 Les techniques non supervisées	25
1.4.4 Les techniques supervisées	25
1.5 Conclusion	25

1.1 Introduction

Notre vie quotidienne est submergée d'informations, mais notre capacité d'analyse est très limitée face au grand volume de données qui ne cesse de croître que l'on appelle actuellement données massives (angl. Big Data). Les progrès des technologies de l'information et de la technologie de stockage ont poussé les travaux de recherche de connaissances à partir de bases de données disponibles, et qui sont généralement peu ou pas exploitées au sein des organisations (Bendoly, 2003; Marban, Segovia, Menasalvas, et Fernandez-Baizan, 2009; Kamsu-Foguem et Mathieu, 2014). Aussi, des nouvelles disciplines connexes sont nées, tels que la science des données (angl. Data Science) qui est l'extraction d'ensembles de données (Dhar, 2012), utilisant des théories et techniques issues de plusieurs domaines tels que les mathématiques, la statistique et l'informatique. Les méthodes qui s'adaptent aux données massives (big data) sont surtout intéressantes dans la science des données.

Un des processus de la science de données est l'Extraction de Connaissance à partir de Données (ECD) ou KDD¹ en anglais. Plusieurs méthodes de traitement statistique des données sont appliquées sur les informations recueillies, mais celles-ci ont montré leurs limites quant à l'interprétation des résultats obtenus (Renaud, Bonjour, Chebel-Morello, Fuchs, et Matta, 2008). Des méthodes issues de l'intelligence artificielle relatives à la fouille de données ont aussi été développées afin d'extraire de nouvelles connaissances en utilisant des techniques avancées d'analyse de données (Fayyad, Piatetsky-Shapiro, et Smyth, 1996b).

L'E.C.D. a deux buts différents : la description et la prédiction. L'objectif descriptif vise à donner une vue globale des données en les regroupant dans des ensembles les plus homogènes possibles ; ainsi, les données deviennent plus faciles à appréhender. L'objectif prédictif consiste à estimer les informations a priori inconnues. En général, l'ECD vise à extraire des informations potentiellement utiles à partir de grands volumes de données en tant que sources de nouvelles connaissances (Harding, Shahbaz, et Kusiak, 2005). De même, les connaissances mises en évidence pourrait venir enrichir la base de connaissances et conduirait à identifier les informations majeures contenues dans ladite base de données.

1. Knowledge Discovery in Databases

1.2 La science des données et Big data

La science des données est l'étude de l'extraction généralisable de connaissances à partir de données (objet), mais le mot clé est science. Il incorpore divers éléments et s'appuie sur des techniques et théories de nombreux domaines, notamment le traitement du signal, les mathématiques, les modèles de probabilité, l'apprentissage automatique, l'apprentissage statistique, la programmation informatique, l'ingénierie de données, la reconnaissance et l'apprentissage de modèles, la visualisation, la modélisation des incertitudes, la calcul de performance... Bien que l'utilisation du terme science des données ait explosé dans les environnements d'affaires, de nombreux universitaires et les journalistes ne font aucune distinction entre la science des données et les statistiques.

Gil Press écrit, dans Forbes, que « (...)la science des données est un mot à la mode sans définition claire et qu'elle a simplement remplacé l'analyse commerciale dans des contextes tels que les programmes d'études supérieures » (Press, 2013). La science des données fait référence à un nouveau domaine d'activités lié à la collecte, la préparation, l'analyse, la visualisation, la gestion et la conservation d'importantes collections d'informations. Bien que le terme science des données ou (Data Science) semble être le plus en contact avec des domaines tels que les bases de données et l'informatique, de nombreux types de compétences, y compris des compétences non mathématiques, sont nécessaires comme expert en la matière. Dans notre cas, le domaine d'expertise est la didactique de disciplines, plus précisément la didactique de l'informatique (Cf. Figure 1.1).

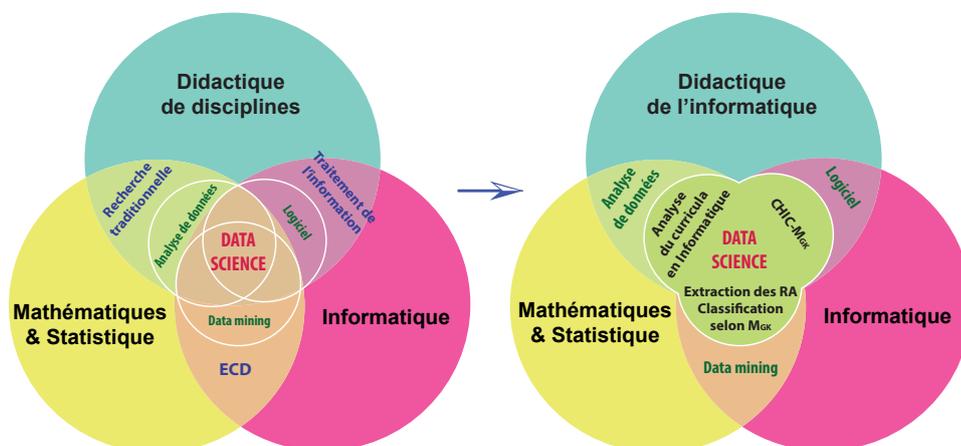


Figure 1.1 – Diagramme de Venn de la data science

La data science, une discipline très récente est maintenant en pleine expansion grâce aux grands volumes de données stockées et fournies par des entreprises, aux données publiques par des multitudes réseaux sociaux sous différentes formes que nous appelons données massives (big data) et les avancés techniques et technologiques pour le stockage, la gestion et le traitement de ces données avec des langages de programmation afin de produire la valeur des jeux de données (datasets) prêts à exploiter en temps réel.

En 1997, big data était né lorsque Michael Cox et David Ellsworth utilisaient ce terme pour la première fois dans leur article : « (...)challenge for computer systems : data sets are generally quite large, taxing the capacities of main memory, local disk, and even remote disk. We call this the problem of big data(...) »² et seulement en 2001, Doug Laney analyste et stratège du Groupe Gartner évoquait la problématique du big data selon le principe de 3V - **Volume** : très très grande taille de données à stocker (à l'ordre de 10 à 20 zettaoctets par an), **Variété** : différents types d'informations (structurées, non-structurées et semi-structurées) de diverses sources disponibles et prêtes à manipuler et à traiter et **Vélocité** : fréquence à laquelle les données sont en même temps générées, capturées, partagées, mises à jour et aussi analysées de suite pour répondre les besoins de tout le monde en temps réel. Big data atteint sa maturité en 2013, lorsque le terme a été introduit dans Oxford English Dictionary pour la première fois.

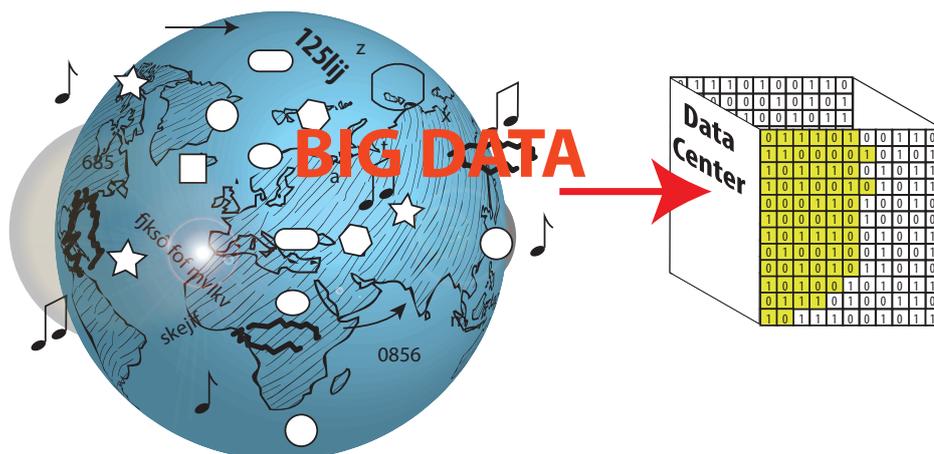


Figure 1.2 – Représentation des données massives

2. « Application-controlled demand paging for out-of-core visualization » in the Proceedings of the 8th conference on Visualisation, IEEE Computer Society Press Los Alamitos, CA, USA, ACM Digital Library, 1997

Big data est un ensemble titanesque d'informations collectées. Chaque seconde, environ 29000 Go d'informations sont publiées dans le monde, soit 2.5 Eo par jour / 912.5 Eo par an³. En 2017, Google par exemple collecte 30 Eo - 30 milliards de Go d'informations. La distribution et le traitement de ces données exigent des techniques performantes et des méthodes efficaces. La répartition des données est assurée par des plusieurs millions d'ordinateurs travaillés en parallèle dans des *Data Center* (Cf. Figure 1.2), sous forme d'un système de fichiers spécial appelé *Hadoop File System* (HDFS), et le traitement se fait selon le principe *map-reduce* qui est un ensemble d'algorithmes faciles à écrire et à exécuter aussi en parallèle appelé *Yet Another Resource Negotiation* (YARN) (*map*: extraction d'informations à partir d'un jeu de données; *reduce*: regroupement de ces informations extraites). L'outil Hadoop  est un système de gestion de données massives et de traitements distribués dont les composantes principales sont le HDFS et le YARN.

1.3 Le processus d'ECD

Les approches d'extraction de connaissances ont développé depuis quelques années grâce à de nouveaux outils « *intelligents* » pour améliorer la découvertes de connaissances dans différents domaines tels que la médecine, l'économie, le commerce, la maintenance, l'industrie qui inclue la conception produits et de processus, la planification des ressources, le contrôle qualité, les systèmes d'aide à la décision, etc. . . , car susceptible de procurer des avantages concurrentiels importants (Zighed et Rakotomalala, 2000).

L'ECD désigne le processus non-trivial d'extraction d'informations implicites non connues à l'avance et potentiellement utiles concernant les données stockées dans les bases de données (Piatetsky-Shapiro et Frawley, 1991). Pourtant, le processus d'ECD n'est pas un processus monolithique et univoque au cours duquel il s'agirait d'appliquer un principe général à tous les types de données stockées ou récupérées. Mais, c'est un processus interactif et itératif d'analyse d'un grand ensemble de données brutes afin d'en extraire des connaissances exploitables par l'utilisateur (Kodratoff, Napoli, et Zighed, 2001). Aussi, c'est un processus non trivial d'identification de motifs ou modèles valides, nouveaux potentiellement utiles et compréhensibles à par-

3. <https://www.planetoscope.com/Internet-/1523-informations-publiees-dans-le-monde-sur-le-net-en-gigaoctets-.html>. Consulté le 04/01/2019

tir de données, composé de plusieurs étapes (Fayyad, Piatetsky-Shapiro, et Smyth, 1996c). C'est un processus interdisciplinaire qui combine différentes approches issues de l'intelligence artificielle, de l'informatique, de l'apprentissage automatique, de la gestion de bases de données, de l'Interaction Homme-Machine (IHM) et surtout et à ne pas négliger des algorithmes mathématiques et statistiques (Liao, 2003).

Nous devons considérer les composantes basiques pour le développement d'un tel système d'extraction de connaissances. Les techniques de « *data mining* » sont au cœur de ces systèmes, puisqu'elles jouent le rôle de moteur d'extraction de ces nouvelles connaissances. Les entrées du système sont généralement des informations issues des bases de données, les sorties étant les connaissances extraites qui devraient être interprétées et évaluées.

1.3.1 Etapes d'un processus ECD

Parmi les différents étapes du processus d'ECD, le « *data mining* » est considéré comme une étape crucial, consistant à appliquer des algorithmes d'analyse de données afin de générer de nouvelles connaissances (Köksal, Batmaz, et Testik, 2011). Le processus classique d'ECD comporte trois étapes : la préparation des données (pré-traitement), l'application des algorithmes d'extraction (*data mining*) et l'évaluation des résultats obtenus (post-traitement).

1.3.1.1 Compréhension du problème

La compréhension de la problématique permet de cibler les buts et guide ainsi la suite du processus. Pour ce faire, on a souvent besoin de bien discuter intensivement avec l'expert utilisateur des données et des connaissances extraites (Lui, Hsy, Chen, et Ma, 2000; Brisson, 2004).

1.3.1.2 Sélection des données

Le choix de données dans la base de données est primordial, car la totalité de données contenant la base de données ne correspond pas forcément au type de données utiles par rapport aux buts fixés. Le jeu de données et les attributs pertinents sont extraits des données brutes. La sélection nous permet de former un échantillon représentatif pour la suite de l'étude.

1.3.1.3 Prétraitement

Cette étape est indispensable dans le processus d'ECD et requiert une attention importante afin de disposer de données fiables avant l'application des algorithmes d'extraction, garantissant ainsi dans une certaine mesure la qualité des résultats. On peut rencontrer différentes difficultés dans l'exploitation des bases de données disponibles pour l'extraction de connaissances. Ces difficultés proviennent du fait que les bases de données du monde réel sont généralement dynamiques, incomplètes, bruitées et contiennent de plus en plus d'informations (Frawley, Piatetsky-Shapiro, et Matheus, 1992). Les valeurs aberrantes ou manquantes sont traitées par l'élimination complète de la ligne ou par interpolation. D'autres préoccupations sont relatives à la question de savoir si la base de données contient des informations adéquates et pertinentes pour l'exploitation. Pour cela, considérer les techniques de nettoyage, de discrétisation, de transformation ou de réduction des données avant l'étape de data mining devient essentiel afin d'améliorer la qualité des données disponibles, puis des connaissances générées. Il s'agit d'une manière générale de comprendre les propriétés des données, de supprimer les données peu intéressantes, d'enrichir les données par des compléments d'informations, de fractionner ou créer de nouveaux attributs, et/ou d'en combiner d'autres.

Nous présentons ci-dessous certains techniques de prétraitement des données susceptibles d'être utilisées en fonction des données disponibles et des formats des données d'entrée demandés par les algorithmes de data mining. Afin d'essayer d'améliorer leur qualité en termes de consistance, complétude, précision, homogénéité, etc., on doit pré traiter le sous-ensemble représentatif sélectionné. En effet, plus les données sont bruitées, moins le résultat final sera pertinent. Ce pré-traitement se compose de diverse phases : le nettoyage permettant de compléter les données manquantes et d'éliminer les valeurs aberrantes et les doublons, l'homogénéisation conduisant à l'intégration des données provenant des sources diverses, la standardisation amenant à une normalisation et à une mise à l'échelle des données.

- **Nettoyage des données** (data cleaning) : consiste à détecter, corriger et éliminer les erreurs, les inexactitudes et/ou les incohérences des données (Rahm et Do, 2000). Certaines incohérences dans les données pourraient être corrigées manuellement, cependant, quand il s'agit de grandes quantités de données, il devient nécessaire d'utiliser des systèmes automatiques ou semi-automatiques

afin de détecter des difficultés et de corriger certains problèmes comme les données manquantes ou aberrantes, les erreurs lors de saisie des informations par les opérateurs ou les données non valides. Le principal objectif de cette étape est donc de générer des bases de données modifiées (soit en inférant des données soit en corrigeant d'autres) avant l'application des algorithmes d'analyse (Cf. Figure 1.3). Il est nécessaire de définir des règles pour gérer ou pour corriger les problèmes présents dans les données. Plusieurs solutions ont été proposées. Par exemple dans le cas des données numériques, des données manquantes peuvent être remplacées par la valeur la plus fréquente de l'attribut concerné, ou elles peuvent être estimées par des méthodes d'induction (Zighed et Rakotomalala, 2002). D'autre part, pour le traitement des données aberrantes, des règles ou stratégies de traitement doivent être préétablies afin de détecter les données qui peuvent gêner l'analyse. Des solutions ont été aussi proposées dans le cas des données qualitatives (Hellerstein, 2008); toutefois, la plupart des méthodes de nettoyage se concentrent surtout sur les erreurs dans les attributs de type quantitatif.

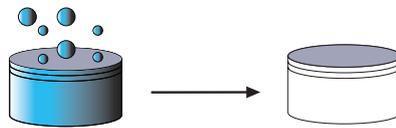


Figure 1.3 – Nettoyage de données

- **Intégration des données:** un autre contexte qui demande un prétraitement de données provient du fait d'avoir plusieurs sources de données, i.e. des données localisées sur des sites différents (bases de données relationnelles, entrepôts de données, sources externes, etc.) (J. Han et Kamber, 2006). Dans ce contexte, l'objectif est d'intégrer les bases de données concernées dans une nouvelle base qui regroupe toute l'information pertinente pour faciliter l'application des techniques de data mining (Cf. Figure 1.4).

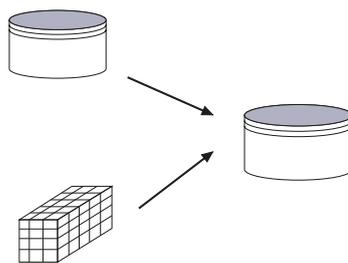


Figure 1.4 – Intégration de données

- **Réduction des données:** en outre, il y a généralement dans les bases de données des attributs ou même des transactions qui ne sont pas intéressantes ou significatives dans un contexte donné. De ce fait, il est possible de réduire le volume de données afin de ne considérer que celles qui sont les plus utiles et intéressantes à analyser en fonction de l'objectif du processus d'extraction. On peut ainsi optimiser la mémoire et le temps d'analyse des algorithmes tout en maintenant l'intégrité des données d'origine (Cf. Figure 1.5).

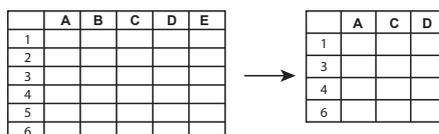


Figure 1.5 – Réduction de données

Dans ce but, des méthodes statistiques d'échantillonnage, de sélection de transactions ou de sélection d'attributs peuvent être considérées selon des conditions exprimées par l'utilisateur (Zighed et Rakotomalala, 2002). Parmi les différentes techniques de réduction de données, la discrétisation est une forme de réduction consistant à transformer des attributs (en les séparant), ou à réduire le nombre de niveaux d'un attribut (en remplaçant les concepts de bas niveau par des concepts de haut niveau d'une hiérarchie de concepts).

1.3.1.4 Transformation des données

Généralement, les données issues des bases de données ne sont pas exploitables directement par les techniques de data mining, c'est pourquoi une transformation des données initiales est en général nécessaire afin de garantir que celles-ci soient dans le format d'entrée demandé par l'algorithme de data mining utilisé (Cf. Figure

1.6). Tout dépend donc de la technique choisie, certaines étant plus contraignantes que d'autres.



Figure 1.6 – Transformation de données

1.3.1.5 Fouille de données

La ou les méthodes de fouille de données, traduction de « data mining », doivent être choisies en fonction des objectifs visés (classification, segmentation, association) et/ou de leur disponibilité. La complémentarité des techniques amènent avantageusement à les combiner pour un même jeu de données. Insistons ainsi que la fouille de données n'est qu'une étape d'ECD. Malheureusement, il y a souvent une confusion entre ces deux termes, au point que certains auteurs parlent indifféremment de fouille de données et d'extraction de connaissances à partir de bases de données. Cette étape consiste à investir/appliquer les techniques aux données transformées (Pasquier, 2000; Azé, Lucas, et Sebag, 2003). Certaines méthodes nécessitent d'ajuster des paramètres et de faire des essais pour optimiser les solutions. Nos travaux de recherche concernent essentiellement la fouille de données.

1.3.1.6 Post-traitement

Le post-traitement, appelé également post-mining, concerne la dernière étape du processus d'ECD. En effet, les modèles ou motifs extraits ne devraient pas être utilisés directement en toute confiance, il faut d'abord une étape d'analyse et d'interprétation de l'information découverte afin d'estimer sa qualité (Giudici, 2003). Il est essentiel que les modèles soient évalués en termes de qualité et/ou d'efficacité avant leur utilisation possible sur le terrain (Larose, 2005). Plus particulièrement, cette dernière étape vise l'évaluation, la visualisation et la validation par l'utilisateur des résultats issus de l'étape du data mining, afin de pouvoir garantir les résultats extraits comme de nouvelles « connaissances » dans le domaine. Dans de nombreux cas, il est possible de définir des mesures statistiques, objectives ou d'utilité pour définir l'intérêt d'un modèle extrait.

Les modèles issus de la fouille de données sont ensuite interprétés, avec prise en compte de l'avis de l'expert (eu égard à ses problématiques et objectifs). Leur interprétation conduit à la validation ou à la réfutation qui pourrait remettre en cause tout le processus ou une partie du processus d'ECD (Lallich et Teytaud, 2004). Diverses méthodes de validation sont envisageables. Les modèles issus d'une classification pourront être vérifiés en premier lieu par un expert, puis la validation sera complétée par des tests statistiques sur des bases de cas existantes. Pour des techniques d'apprentissage non supervisées telles que la segmentation et l'association, la détermination de la pertinence des modèles obtenus est essentiellement une affaire d'expertise. Pour ce qui est de la classification supervisée, une validation croisée est recommandable à partir de trois ensembles de données, c'est-à-dire un ensemble d'apprentissage, un ensemble de tests et un ensemble de validations. Il est même possible d'effectuer une validation croisée permettant de calculer l'erreur d'un modèle construit sur un ensemble par rapport aux entités de ce même ensemble. On retrouve aussi ce problème de validation en apprentissage automatique.

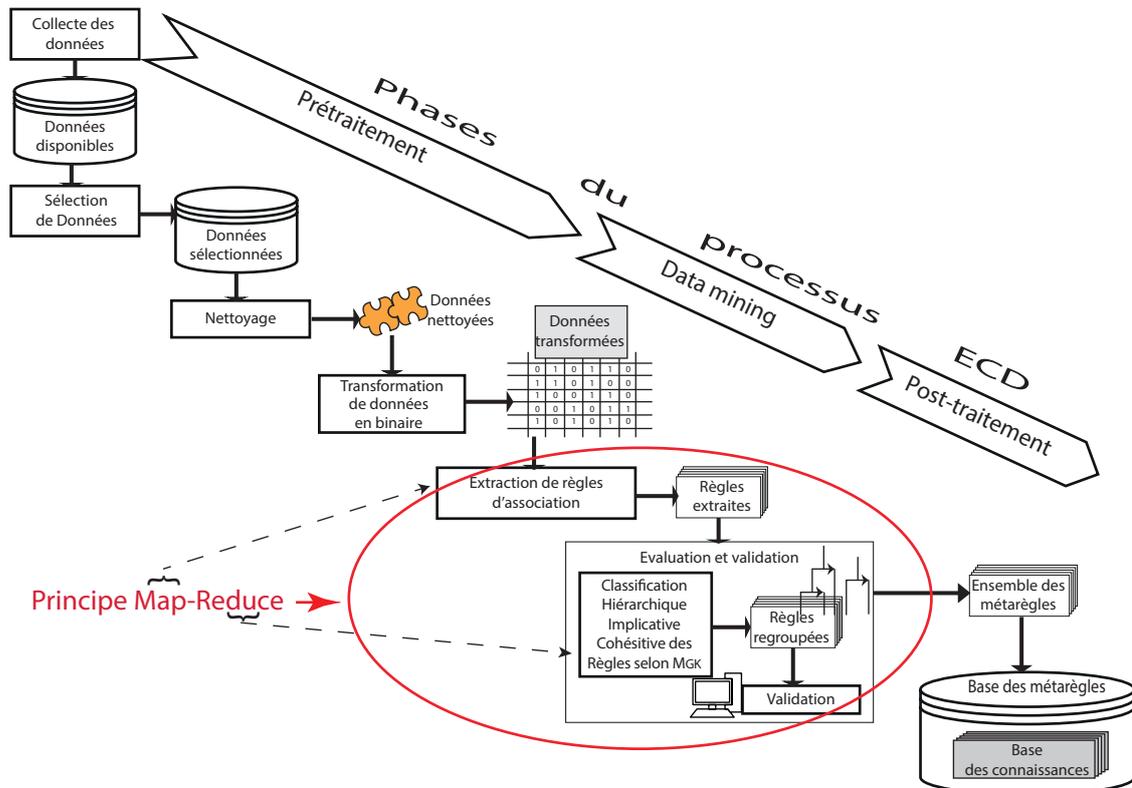


Figure 1.7 – Les étapes d'un processus ECD, (un extrait emprunté à P. A. P. Ruiz (2014, p. 48)).

1.4 Les généralités sur le data mining

Le data mining est une étape fondamentale du processus d'extraction de connaissances. Les systèmes de data mining sont classifiés suivant différents critères tels que le type de bases de données exploitées, le type de connaissances extraites à partir des données, le type de technique utilisée ou le domaine d'application (J. Han et Kamber, 2001). Ces systèmes sont souvent classés en deux groupes principaux (J. Han et Kamber, 2006) en fonction de la nature des données considérées ainsi que du type de connaissances extraites : data mining descriptif et data mining prédictif.

- Le data mining descriptif selon Giudici (2003) ; Dasu et Johnson (2003) se concentre principalement sur la découverte de modèles ou de relations afin de décrire les données.
- Le data mining prédictif se concentre surtout sur la prédiction du comportement d'un modèle, et détermine les valeurs futures des variables à partir de l'information issue des bases de données (?).

1.4.1 Les méthodes descriptives

Les méthodes descriptives ont pour objectif d'organiser, de simplifier et d'aider à comprendre les phénomènes existant dans un ensemble important de données non étiquetées. Cet ensemble est organisé en instances constituées de plusieurs variables descriptives, où aucune des variables n'a d'importance particulière par rapport aux autres. Toutes les variables sont donc prises en compte au même niveau.

- La « **description** », (Rencher, 2003) consiste à dégager les aspects les plus intéressants de la structure des données. Les techniques d'analyse factorielles consiste à dégager des variables cachées dites « facteur » à partir d'un ensemble de mesures. L'utilité de ces facteurs réside dans le fait qu'un nombre réduit de ces derniers explique aussi bien les données que l'ensemble des variables descriptives. Parmi, les techniques factoriels, on a : Analyse en Composantes Principale (ACP) pour les variables quantitatives, Analyse des Correspondances Multiples (ACM) pour les variables qualitatives, Analyse Factorielle des Correspondances (AFC) pour les variables qualitatives et Analyse Factorielle Multiple (AFM) pour des groupes de variables quantitatives et/ou

qualitatives.

- **Le « clustering »**, appelé également regroupement ou classification non supervisée, cherche à identifier et à créer des sous-ensembles homogènes d'objets, catégories ou groupes (i.e. des clusters) pour décrire les données, à partir d'un ensemble total hétérogène d'objets dans la base de données. Contrairement à la technique de classification présentée précédemment, le clustering analyse les données sans connaître a priori les classes ou groupes. Cette technique de type descriptif permet en effet de fractionner la base de données en groupes d'objets ayant des caractéristiques similaires, e.g. en utilisant le principe de la similarité, qui est généralement mesurée par des fonctions de distance entre les objets. La technique a pour objectif de maximiser la similarité intra-groupe tout en minimisant la similarité entre groupes différents ([J. Han et Kamber, 2001](#)). Ces techniques sont souvent classées en plusieurs catégories : les méthodes de partitionnement, les méthodes hiérarchiques, les méthodes basées sur la densité, les méthodes basées sur la grille, et les méthodes fondées sur un modèle (?). Un exemple de clustering dans le domaine du marketing inclut la découverte de différentes catégories homogènes pour représenter des groupes cibles de consommateurs ([Huang, Tzeng, et Ong, 2007](#)).
- **La « summarization »**, appelée également caractérisation, considère l'identification et la présentation des caractéristiques générales ou propriétés typiques d'un sous-ensemble de données, en essayant de trouver une description concise pour chaque sous-ensemble ([Fayyad et al., 1996c](#); [Lee et Kim, 1997](#)). Un exemple simple de summarization concerne le calcul de la moyenne et des écarts-types pour un ensemble de données, souvent utile pour l'analyse ou la visualisation des données.
- **Les « règles d'association »** concernent une technique de data mining de type descriptif qui cherche à identifier des groupes d'items (attributs et leur valeur) qui se produisent ensemble. Le problème d'extraction de règles d'association a été introduit par [Agrawal, Imielinski, et Swami \(1993b\)](#) dans le domaine des transactions de ventes, chacune constituée d'une liste d'articles achetés, afin de découvrir des relations entre les articles le plus fréquemment achetés ensemble. Ces relations ne sont pas basées sur des propriétés inhérentes aux données (comme les dépendances fonctionnelles), elles sont plutôt basées

sur la co-occurrence des items dans les données de la base (?). Une règle dans ce domaine sera par exemple : « les clients qui achètent du lait ont une tendance à acheter des céréales ».

- « **La découverte de motifs séquentiels** » est une technique de type descriptif d'identification d'associations ou de modèles séquentiels au fil du temps (Agrawal et Srikant, 1995). Il s'agit de la détection de motifs séquentiels fréquents ou périodiques à partir d'une base de séquences, dans laquelle chaque séquence est décrite par une liste ordonnée d'ensembles d'items. Un exemple classique de cette méthode concerne l'identification de la séquence d'achat d'un client afin de prédire la suite de cette séquence.

1.4.2 Les méthodes prédictives

Les méthodes prédictives permettent de prévoir et d'expliquer à partir d'un ensemble de données étiquetées un ou plusieurs phénomènes observables.

- **La « classification »** est une technique de type prédictif qui affecte les objets d'une base de données à une des classes prédéfinies (Fayyad et al., 1996c) selon un modèle (ou fonction) de classification préétabli. Le principal objectif est donc ici d'utiliser le modèle (règles de classification, arbres de décision, etc.) pour prédire la classe de chaque objet et/ou concept analysé (J. Han et Kamber, 2006) dans la base de données, i.e. chaque objet va être associé à un nom de classe. De nombreuses applications relèvent de cette problématique, comme le diagnostic médical, la gestion des stocks ou le ciblage de clientèle (Ngai, Xiu, et Chau, 2009). Un exemple classique de cette technique concerne l'étude des marchés financiers, afin de classifier de manière automatique les futurs demandeurs de prêts et ainsi aider la prise de décision dans l'entreprise.
- « **La régression** » est une technique de type prédictif qui associe un élément des données à une variable de prédiction. Dans ce contexte, la régression linéaire concerne la méthodologie statistique la plus utilisée pour la prévision numérique (J. Han et Kamber, 2006). En revanche, la régression logistique est la principale méthode de prédiction pour les variables qualitatives (Giudici, 2003). D'autres techniques associées à la prédiction sont entre autres les arbres de décision, les réseaux bayésiens ou les réseaux de neurones.

Par rapport à ces deux méthodes sus mentionnées, les techniques d'extraction des connaissances dans les Données se groupent en deux classes ([Alessandri, 2010](#)): les techniques non supervisées et les techniques supervisées.

1.4.3 Les techniques non supervisées

Les techniques non supervisées (*clustering*) cherchent à détecter les régularités dans les données et à les classer sans fixer, à priori, d'éléments à découvrir. L'utilisation de ces techniques peut se baser sur les statistiques, les réseaux de neurones, les règles d'association.

1.4.4 Les techniques supervisées

Les techniques supervisées (*classification*) consistent à choisir au préalable un ou plusieurs attributs ou items connus, appelé(s) attribut(s) endogène(s). A partir de ce choix, à déterminer au sein des autres attributs, les conditions associées aux valeurs particulières présent par les attributs endogènes. Ces techniques comprennent les arbres de décisions, les réseaux bayésiens, les réseaux de neurones et quelques techniques statistiques.

1.5 Conclusion

Nous constatons que les réseaux de neurones sont des techniques qu'on peut utiliser dans les deux méthodes d'extraction de connaissances. Selon les cas, la nature de ces méthodes d'extraction peut être supervisée ou non. Par exemple, pour la recherche de règles d'association classée parmi la méthode supervisée comme l'arbre de décision, utilisée pour l'extraction des règles d'association, génère par essence une arborescence dans le processus de recherche comme l'algorithme CHARM ([Zaki et Hsiao, 2002](#)). Une autre méthode non supervisée, et c'est dans le cadre de cette thèse, la combinaison de la méthode des **règles d'association** et du **clustering**, d'utiliser des variables cibles, correspondant à des seuils appliqués aux mesures d'intérêt qui décrivent des relations généralement orientées initialement inconnues entre certaines variables ou certains motifs, cachées dans la base de données, mais potentiellement utiles, et ainsi d'anticiper des comportements. La **classification hiérarchique co-**

hésitive et implicative, inspirée du CHIC de [Gras \(1979\)](#), qui consiste à extraire un ensemble de formules logiques conditionnelles déduisant la valeur d'un ensemble d'attributs à partir des valeurs d'autres classes d'attributs apparaissant simultanément. Mais cette fois-ci, **selon la mesure de qualité** M_{GK} , appelée mesure de Guillaume-Kentchaff ([Guillaume, 2000](#)). La constitution des classes se décide de façon automatique et non supervisée selon la valeur de cohésion implicative évaluée toujours à partir de la mesure d'intérêt M_{GK} . Mais tout d'abord, nous allons présenter dans le chapitre 2 suivant des notions mathématiques, à savoir les Treillis, les Familles de Moore, les Correspondances de Galois, qui seront utiles dans la théorie de la fouille des règles d'association. Ainsi que le système implicatif et les hiérarchies, utiles dans la classification hiérarchique des règles extraites.

Chapitre 2

Structure de treillis et famille de Moore

“ La structure narcissique a un caractère irréductible. ”

Jacques Lacan
Psychanalyste, Scientifique
(1901 - 1981)

Sommaire

2.1	Introduction	28
2.2	Historiques	28
2.3	Une introduction aux ensembles ordonnés et aux treillis	29
2.3.1	Les notions mathématiques d'ordre et d'ensemble ordonné	29
2.3.2	La notion mathématique de treillis	31
2.3.3	Treillis et éléments irréductibles	36
2.3.4	Table d'un treillis	38
2.4	Treillis des fermés, système de fermetures	41
2.4.1	Treillis des fermés	42
2.4.2	Système de fermeture	43
2.5	Correspondances de Galois	44
2.5.1	Applications galoisiennes	44
2.5.2	Correspondances de Galois associées à une relation binaire	45
2.6	Système implicatif	47
2.7	Hierarchies	49
2.7.1	Chaîne de partitions et arbre de classification associée	51
2.7.2	Construction d'un dendrogramme ordinal compatible avec $\mathcal{H}_b(S)$	53

2.7.3	Hiérarchie de fourches, hiérarchie binaire de parties et arbre binaire de classification	56
2.7.4	Hiérarchie de fourches orientées et hiérarchie implicative	58
2.7.5	Ultramétrie	60
2.8	Conclusion	64

2.1 Introduction

Dans un cadre applicatif, la structure de treillis des concepts ou de treillis des fermés fournit une représentation très intuitive des données lorsqu'elles sont décrites par une table binaire ou bien des règles d'implication. Depuis les années 2000, l'émergence de l'Analyse Formelle de Concepts (AFC) dans divers domaines de l'informatique, que ce soit en extraction qu'en représentation de connaissances, dans les domaines des ontologies ou encore des bases de données, a ainsi mis en avant les structures de treillis des concepts. En vue de leurs applications dans l'administration d'une société, ces notions mathématiques sont utiles dans la fouille des règles d'association et à la classification hiérarchique orientée, que nous présentons dans ce chapitre, à savoir des treillis et ses évolutions dans le temps dans les sections 2.2, des notions de bases sur les ensembles ordonnés et quelques définitions sur les treillis 2.3, des treillis des fermés et système de fermés (Familles de Moore) dans la section 2.4, des correspondances de Galois dans la section 2.5, des systèmes implicatifs dans la section 2.6 et des hiérarchies dans la section 2.7.

2.2 Historiques

Autour des années 1890, R. Dedekind a basé ses travaux sur la distributivité en considérant la question : *combien de sous-groupes différents peut-on avoir avec trois sous-groupes G_1, G_2, G_3 d'un groupe commutatif G , en utilisant seulement la somme et l'intersection ?* On a par exemple, comme résultats les sous-groupes, $G_1 + G_2, G_1 + G_3, \dots, (G_1 + G_2) \cap G_3, \dots$ (Szasz, 1971). Il constata en même temps que quelques éléments sur les treillis avaient été déjà étudiés par Ernst Schöler dans son livre *Die Algebrader Logik* et conclut qu'il y a vraiment un lien entre la théorie de treillis et l'algèbre moderne. Vers les années 1930, les travaux de R. Dedekind ont

été découverts par Birkhoff (1933) et Öre (1944) pour le développement moderne de la théorie du réseau, « *il a été supposé que, dans un réseau modulaire fini, le nombre d'inf-irréductibles est égal au nombre de sup-irréductibles* », et ils utilisaient le terme *structure* pour la première fois. Ce théorème a été prouvé dans les travaux de Dilworth (1990). Le terme *treillis* a été proposé par Birkhoff lors du premier symposium sur la structure de treillis en 1938 et a été inséré dans (Birkhoff, 1940). Plusieurs ouvrages portent sur la définition ordinaire sur la théorie des treillis (Szasz, 1971; Grätzer, 1978; Davey et Priestley, 1991). Quelques nouveaux termes ont été introduits comme *correspondance de Galois* par Öre (1944), *treillis de Galois* par Barbut et Monjaret, car un résultat principal a été établi que « *tout treillis fini est isomorphe au treillis de Galois ou treillis de concepts de sa table* » (Barbut et Monjaret, 1970). Le *treillis des concepts* a été introduit par Wille dans les années 1980 dans le cadre de l'Analyse Formelle des Concepts (Wille, 1982; Ganter et Wille, 1999), qui définit le treillis de concepts à partir des données binaires (ou contexte) de types *individus* \times *variables*. Le nœud du treillis, appelé *concept*, est un regroupement maximal d'individus possédant des variables en commun. Le treillis composé de concepts reliés par inclusion, fournit une représentation des données très intuitive.

2.3 Une introduction aux ensembles ordonnés et aux treillis

En mathématiques, entre les anti-chaînes où les éléments d'aucun couple d'un ensemble ne sont comparables, aussi les ensembles totalement ordonnés où tous les éléments sont comparables, existe la famille des ensembles partiellement ordonnés. Les ensembles partiellement ordonnés sont omniprésents, mais généralement l'ordre partiel est antisymétrique.

2.3.1 Les notions mathématiques d'ordre et d'ensemble ordonné

Définition 2.1. Une relation binaire \mathcal{R} sur un ensemble X est une propriété portant sur les couples d'éléments de X . On notera $x\mathcal{R}y$ le fait que la propriété est vraie pour le couple $(x, y) \in X \times X$.

Définition 2.2. Soit \mathcal{R} une relation binaire sur un ensemble X .

- \mathcal{R} est réflexive si $\forall x \in X$, on a $x\mathcal{R}x$.
- \mathcal{R} est symétrique si $\forall x, y \in X$, on a $x\mathcal{R}y \Rightarrow y\mathcal{R}x$
- \mathcal{R} est antisymétrique si $\forall x, y \in X$, $x\mathcal{R}y$ et $y\mathcal{R}x \Rightarrow x = y$.
- \mathcal{R} est transitive si $\forall x, y, z \in X$, $x\mathcal{R}y$ et $y\mathcal{R}z \Rightarrow x\mathcal{R}z$

Terminologie et notation

- Si \mathcal{R} est une relation d'ordre sur X , alors le couple (X, \mathcal{R}) est dit un ensemble ordonné.
- Si \mathcal{R} est réflexive et transitive, alors on dit que (X, \mathcal{R}) est un ensemble pré-ordonné ou quasi-ordonné; on dit que \mathcal{R} est un pré-ordre.

On notera \mathcal{R} par \leq , et l'on parlera alors d'ensemble ordonné (X, \leq) pour faire allusion à la relation d'ordre total \leq dans l'ensemble des réels \mathbb{R} , sauf que (X, \leq) n'est pas nécessairement un ensemble totalement ordonné.

Définition 2.3. Une relation d'équivalence sur un ensemble X est une relation binaire \mathcal{R}_{\approx} de $X \times X$ qui est à la fois réflexive, symétrique et transitive.

Étant donnée une relation d'équivalence \mathcal{R}_{\approx} , on définit la classe d'équivalence, pour \mathcal{R}_{\approx} , d'un élément $x \in X$, comme l'ensemble $P_{\mathcal{R}_{\approx}}(x) = \{y/y \in X \text{ et } y\mathcal{R}_{\approx}x\}$. c'est l'ensemble des éléments y de X qui sont liés à x par la relation d'équivalence \mathcal{R}_{\approx} .

Définition 2.4. L'ensemble des classes d'équivalence pour \mathcal{R}_{\approx} est :

$X_{/\mathcal{R}_{\approx}} = \{C/C \subset X = P_{\mathcal{R}_{\approx}}(x) \text{ pour un } x \in X\}$. $X_{/\mathcal{R}_{\approx}}$ est appelé l'ensemble quotient de X par \mathcal{R}_{\approx} .

La fonction $P_{\mathcal{R}_{\approx}} : X \rightarrow X_{/\mathcal{R}_{\approx}}$ associe à tout élément x de X sa classe d'équivalence pour \mathcal{R}_{\approx} est appelée projection de X sur son ensemble quotient $X_{/\mathcal{R}_{\approx}}$.

Propriété 2.1. $P_{\mathcal{R}_{\approx}}(X)$ est une partition de X , i.e. :

- $P_{\mathcal{R}_{\approx}}$ est une surjection,
- Chaque élément de X appartient à exactement une et une classe d'équivalence pour \mathcal{R}_{\approx} .

Définition 2.5. On dit qu'un ensemble ordonné (C, \leq) est une chaîne si ses éléments vérifient :

$\forall x, y \in C$, on a $x \leq y$ ou $y \leq x$, i.e. si (C, \leq) est totalement ordonné.

Si $C = \{c_0, c_1, \dots, c_n\}$ est une chaîne finie d'un ensemble ordonné P avec $\text{card}(C) = n + 1$, alors on dit que la longueur de C est n .

Définition 2.6. Soit (X, \leq) un ensemble ordonné et x, y deux éléments de X .

- Un élément $b \in X$ est un *minorant* de x et y si $b \leq x$ et $b \leq y$.
- Un élément $u \in X$ est un *majorant* de x et y si $x \leq u$ et $y \leq u$.
- Le plus grand des minorants m de x et y et se note $x \wedge y$ est appelé *borne inférieure*. Il vérifie alors : $m \leq x; m \leq y$; si $(b \leq x \text{ et } b \leq y)$ alors $b \leq m$.
- Le plus petit des majorants l de x et y et se note $x \vee y$ est appelé *borne supérieure*. Il vérifie alors : $x \leq l; y \leq l$; si $(x \leq u \text{ et } y \leq u)$ alors $l \leq u$.

Propriété 2.2. Soient $m = x \wedge y$ et $l = x \vee y$ de X

- Une telle borne inférieure, si elle existe, est unique ;
- Une telle borne supérieure, si elle existe, est unique.

Exemple 2.3.1. Soit une relation binaire \mathcal{R} sur X .

$\{\text{ordres}\} \subset \{\text{relations binaires}\}$. Considérons $X = \{a, b, c, d, e\}$

et $\mathcal{R} = \{(a, b), (a, e), (c, b), (c, d), (c, e), (d, e), (a, a), (b, b), (c, c), (d, d), (e, e)\}$.

Alors \mathcal{R} peut se représenter sous l'une des 3 figures ci dessous (Cf. Figure 2.1).

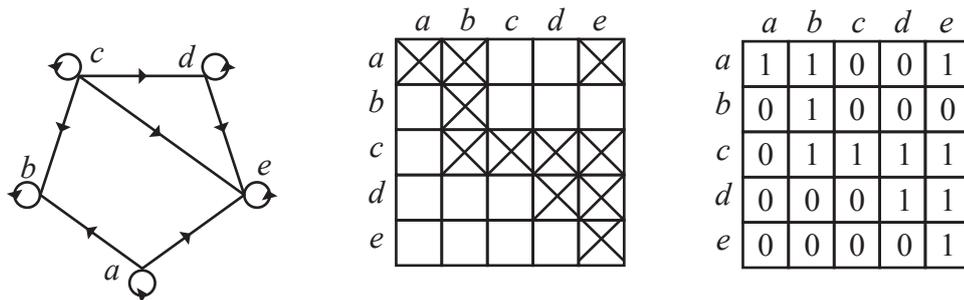


Figure 2.1 – Trois représentations possibles de la relation binaire \mathcal{R}

2.3.2 La notion mathématique de treillis

Définition 2.7 (Définition algébrique (Birkhoff, 1940)). Un treillis ou une algèbre de treillis est un triplet $L = (X, \vee, \wedge)$, où X un ensemble non vide, \vee et \wedge sont deux

opérateurs binaires sur l'ensemble X qui vérifient les propriétés suivantes :

- associativité : $\forall x, y, z \in X, (x \vee y) \vee z = x \vee (y \vee z)$ et $(x \wedge y) \wedge z = x \wedge (y \wedge z)$.
- commutativité : $\forall x, y \in X, x \vee y = y \vee x$ et $x \wedge y = y \wedge x$.
- idempotence : $\forall x \in X, x \vee x = x = x \wedge x$.
- loi d'absorption : $\forall x, y \in X, x \vee (x \wedge y) = x = x \wedge (x \vee y)$.

Définition 2.8 (Définition ordinale (Barbut et Monjardet, 1970)). *Un treillis est une paire $L = (X, \leq)$ où X est un ensemble non vide et \leq est une relation d'ordre sur l'ensemble X , telle que toute paire d'éléments $\{x, y\}$ de X admet à la fois une borne inférieure et une borne supérieure.*

Remarque 1. *La borne inférieure de x et y , notée $x \wedge y$, est l'unique élément maximal (plus grand élément) de l'ensemble des prédécesseurs (ou minorants) de x et y (ensemble des éléments $z \in X$ tels que $z \leq x$ et $z \leq y$).*

Dualement, la borne supérieure de x et y , notée $x \vee y$, est l'unique élément minimal (plus petit élément) de l'ensemble des successeurs (ou majorants) de x et y (ensemble des éléments de $z \in X$ tels que $z \geq x$ et $z \geq y$) (Cf. Propriété 2.2).

Pour une partie $A \subseteq X$: la borne inférieure de A , notée $\bigwedge_{x \in A} x$, est l'unique élément maximal de l'ensemble des minorants de X , alors que la borne supérieure de A , notée $\bigvee_{x \in A} x$, est l'unique élément minimal de l'ensemble des majorants de X . Les treillis se trouvent parmi les ensembles partiellement ordonnés ayant la propriété suivante : pour tout couple d'éléments, il existe un autre élément qui leur est inférieur et un qui leur est supérieur vis-à-vis de la relation d'ordre partiel.

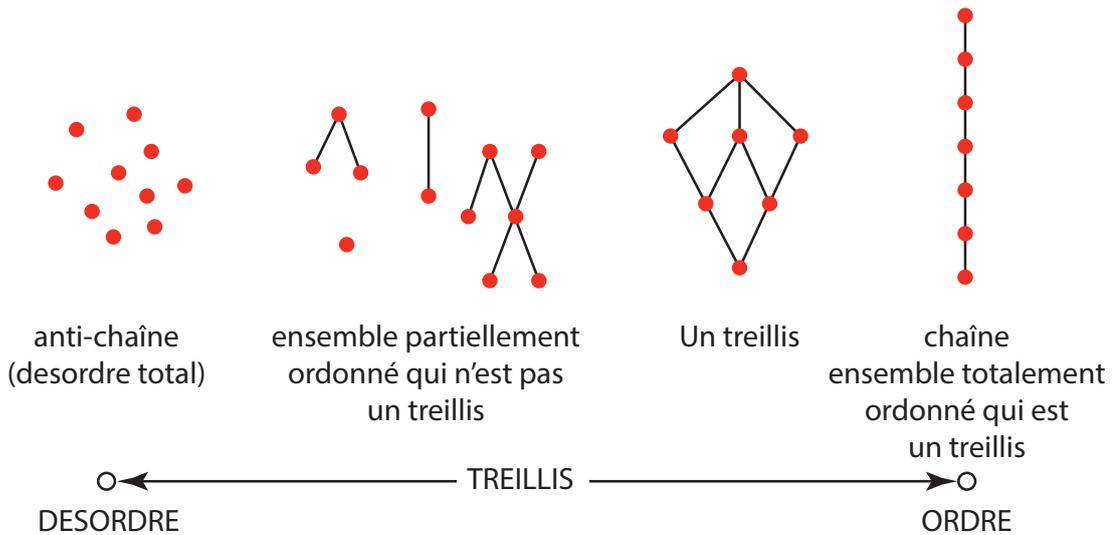


Figure 2.2 – Treillis, entre désordre et ordre

Lorsque seule l'existence de la borne inférieure est vérifiée, on parle d'*inf-demi-treillis*. Un *sup-demi-treillis* est défini dans le cas dual où seule l'existence d'une borne supérieure est vérifiée. Un treillis est donc à la fois un inf-demi-treillis et un sup-demi-treillis. Un treillis admet uniquement un élément maximal noté \top , ainsi qu'un unique élément minimal noté \perp .

A toute relation d'ordre \leq , on associe sa relation d'*ordre strict*: notée $<$, c'est une relation asymétrique, transitive et irréflexive définie par $x < y$, si $x \leq y$ et $x \neq y$.

On associe également à une relation d'ordre \leq sa *relation de couverture* notée \prec : c'est une relation antisymétrique définie par $x \prec y$, si $x < y$, et il n'existe pas d'élément z tel que $x < z < y$. On dit alors que y *couvre* x . Elle se déduit de la relation \leq en supprimant les relations de réflexivité et de la transitivité.

Exemple 2.3.2. *Un ordre est représenté par la figure ci-dessous (Cf. Figure 2.3) avec les relations de transitivité (continues) et de réflexivité (pointillées). Les éléments b et c possèdent e pour borne supérieure et a pour borne inférieure, les éléments e et d n'admettant pas de borne supérieure, et les éléments f et g n'admettant pas de borne inférieure. Ainsi, cet ordre n'est pas un treillis. Il possède a pour unique élément minimal, et f et g pour éléments maximaux. La figure ci-dessous (Cf. Figure 2.4) représente un exemple de treillis. Toute paire d'éléments admet une borne supérieure ainsi qu'une borne inférieure. De plus ce treillis possède v_7 pour unique élément minimal \perp , et v_{10} pour unique élément maximal \top .*

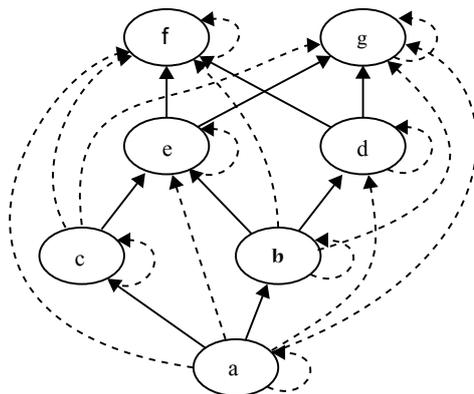


Figure 2.3 – Exemple d'ordre

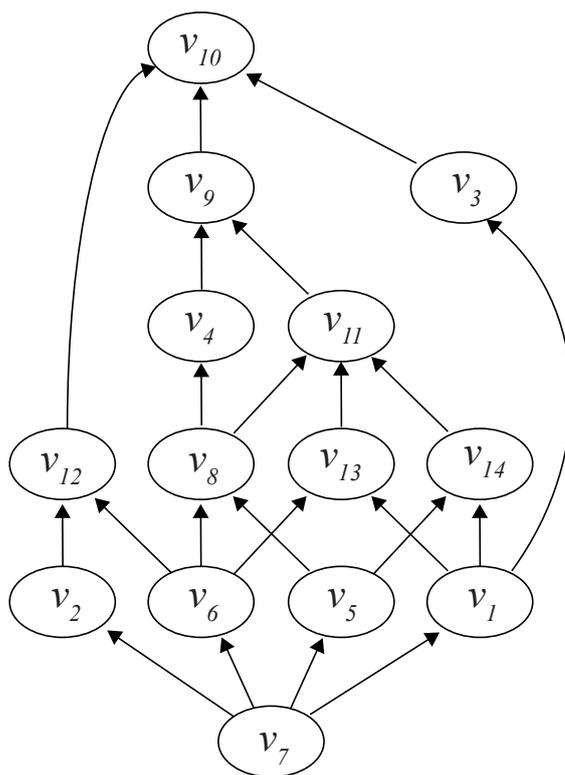


Figure 2.4 – Exemple de treillis

Définition 2.9 (Treillis distributif). *Un treillis est distributif, si \vee et \wedge vérifient les propriétés de distributivité $\forall x, y, z \in X, x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ et $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.*

Définition 2.10 (Treillis complémenté). *Un treillis est complémenté, si tout élément de $x \in X$ admet au moins un complément, i.e. un élément $x' \in X$ tel que $x \vee x' = \top$ (treillis sup-complémenté) et $x \wedge x' = \perp$ son dual (treillis inf-complémenté).*

Définition 2.11 (Treillis booléen). *Un treillis est booléen, s'il est à la fois distributif et complémenté.*

Exemple 2.3.3. *Un exemple classique de treillis distributif est un ensemble fini de nombres muni des deux opérateurs $\wedge = \text{pgcd}$ et $\vee = \text{ppcm}$ (Cf. Figure 2.5). La relation d'ordre est alors la relation « divise ». L'ensemble $\mathcal{P}(X)$ des parties d'un ensemble S muni de la relation d'inclusion $(\mathcal{P}(X), \subseteq)$ est quant à lui un exemple classique de **treillis booléen**, i.e. de treillis distributif et complémenté; la relation d'inclusion \subseteq est un ordre, $\vee = \cup$ et $\wedge = \cap$ (Cf. Figure 2.6).*

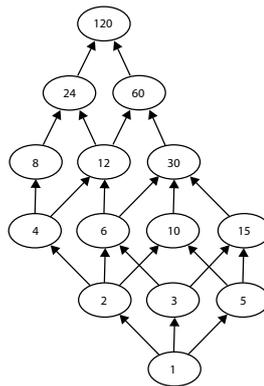


Figure 2.5 – Exemple de treillis distributif

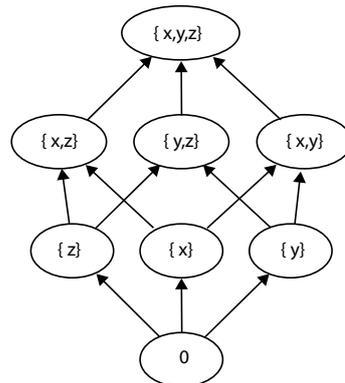


Figure 2.6 – Exemple de treillis booléen

2.3.3 Treillis et éléments irréductibles

Définition 2.12 (Éléments irréductibles). *Soit $L = (X, \leq)$ un treillis.*

- Un élément $j \in X$ est appelé sup-irréductible si, pour tout sous-ensemble A de X , $j = \vee A \Rightarrow j \in A$.
- Un élément $m \in X$ est appelé inf-irréductible si, pour tout sous-ensemble A de X , $m = \wedge A \Rightarrow m \in A$.

L'ensemble des sup-irréductibles du treillis L est noté J_L , et celui des inf-irréductibles M_L .

Proposition 2.1. *Pour un treillis fini, et à partir de la relation de couverture \prec du treillis, on a les propriétés suivantes :*

- Un élément est sup-irréductible, si et seulement si il couvre un seul élément, noté j^- .
- Un élément est inf-irréductible, si et seulement si il est couvert d'un seul élément, noté m^+ .

Un *atome* est un élément couvrant l'élément minimal \perp , tandis qu'un *coatome* est un élément couvert par l'élément maximal \top .

Définition 2.13 (Treillis atomistique). *Le treillis L est un treillis atomistique si tous les sup-irréductibles sont des atomes.*

Définition 2.14 (Treillis coatomistique). *Le treillis L est un treillis coatomistique si tous les inf-irréductibles sont des coatomes.*

Tout élément $x \in X$ d'un treillis $L = (X, \leq)$ est la borne supérieure de l'ensemble de ses prédécesseurs, ainsi que la borne inférieure de l'ensemble de ses successeurs. La définition latticielle permet d'établir facilement qu'il est possible de réduire ces ensembles aux seuls prédécesseurs sup-irréductibles, et successeurs inf-irréductibles :

$$x = \vee J_x = \vee \{y \text{ sup-irréductible tel que } y \leq x\}$$

$$x = \wedge M_x = \wedge \{y \text{ inf-irréductible tel que } y \geq x\}$$

Les ensembles d'éléments irréductibles portent l'information minimale permettant de reconstruire le treillis dans sa globalité par reconstructions successives de bornes

supérieures (en utilisant les sup-irréductibles) ou de bornes inférieures (en utilisant les inf-irréductibles).

Définition 2.15. Soit $L = (X, \leq)$ un treillis et $x \in X$ un élément du treillis. Un générateur minimal de x est un sous-ensemble B de J_x tel que $x = \vee B$ et qui soit minimal au sens de l'inclusion, i.e. pour tout $A \subset B$, on a $x < \vee A$. La famille \mathcal{B}_x des générateurs minimaux de x se définit par :

$$\mathcal{B}_x = \{B \subseteq J_x : x = \vee B \text{ et } x < \vee A, \forall A \subset B\} \quad (2.1)$$

L'observation duale vaut pour la reconstruction de x comme borne inférieure à partir de M_x . Le nombre de générateurs minimaux d'un élément x peut se réduire à 1 (le seul ensemble J_x), mais peut également être élevé, au pire exponentiel en la cardinalité de J_x , d'où une explosion combinatoire de la cardinalité de \mathcal{B}_x .

Exemple 2.3.4. Considérons l'exemple du treillis de la figure 2.4. Nous avons six éléments qui possèdent un seul arc entrant (flèche bleue), et forment l'ensemble des sup-irréductibles. Huit éléments possèdent un seul arc sortant (flèche rouge), et forment l'ensemble des inf-irréductibles : $J = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, $M = \{v_2, v_3, v_4, v_9, v_{11}, v_{12}, v_{13}, v_{14}\}$.

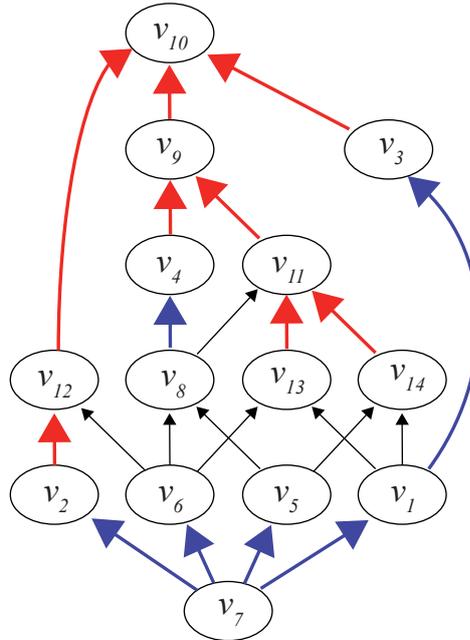


Figure 2.7 – Treillis de la figure 2.4 pour définir les ensembles J_x et M_x

Outre cela, nous avons quatre éléments sup-irréductibles qu'on appelle des co-atomes $\{v_1, v_2, v_5, v_6\}$ (caractère bleu), alors que trois inf-irréductibles appelés atomes $\{v_3, v_9, v_{12}\}$ (caractère rouge). Ces éléments irréductibles sont utilisés dans le treillis de la figure 2.8 où, dans le nœud de chaque élément x sont précisés l'ensemble J_x des sup-irréductibles inférieurs ou égaux à x ainsi que l'ensemble M_x des inf-irréductibles supérieurs ou égaux à x .

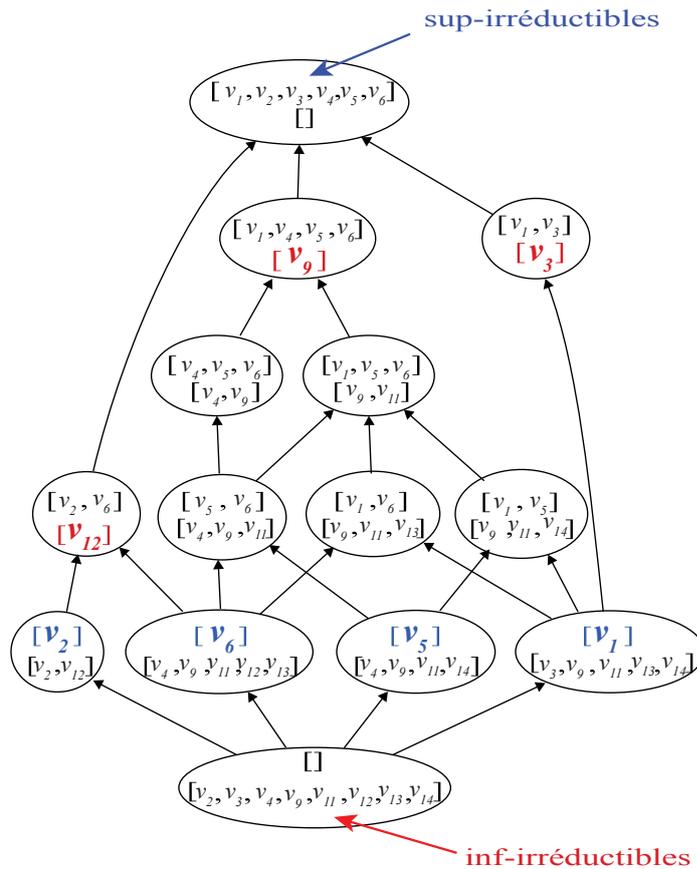


Figure 2.8 – Treillis de la figure 2.4 où sont précisés, pour chaque élément x , les ensembles J_x et M_x

2.3.4 Table d'un treillis

La notion d'éléments irréductibles d'un treillis permet de concevoir des représentations compactes du treillis qui contiennent l'information nécessaire à sa reconstruction. L'une de ces représentations est sous forme de table composée en colonne des sup-irréductibles et en ligne des inf-irréductibles.

La *table* d'un treillis se définit à partir des *relations flèches* des différents liens possibles entre un sup-irréductible et un inf-irréductible à l'aide des cinq relations binaires définies par $J_L \times M_L$. La relation de comparabilité \leq tout d'abord permet d'établir une première partition de $J_L \times M_L$ en deux parties notées P_{\leq} et $P_{\not\leq}$.

$$P_{\leq} = \{(j, m) \in J_L \times M_L : j \leq m\} \text{ et } P_{\not\leq} = \{(j, m) \in J_L \times M_L : j \not\leq m\}. \quad (2.2)$$

Les deux relations flèches \uparrow et \downarrow permettent d'affiner les liens entre les paires $(j, m) \in P_{\not\leq}$.

Définition 2.16 (Relations flèches (Wille, 1983)). *Soit $L = (S, \leq)$ un treillis, $j \in J_L$ et $m \in M_L$.*

- $j \uparrow m$ si $j \not\leq m$ et $j < m^+$
- $j \downarrow m$ si $j \not\leq m$ et $j^- < m$

Comme ces deux relations ne sont pas disjointes, on peut partitionner l'ensemble des paires $(j, m) \in P_{\not\leq}$ en quatre parties :

$$P_{\not\leq} = \begin{cases} P_{\uparrow} = \{(j, m) \in J_L \times M_L : j \uparrow m \text{ et } j \not\downarrow m\} \\ P_{\downarrow} = \{(j, m) \in J_L \times M_L : j \not\uparrow m \text{ et } j \downarrow m\} \\ P_{\updownarrow} = \{(j, m) \in J_L \times M_L : j \uparrow m \text{ et } j \downarrow m\} \\ P_{\circ} = \{(j, m) \in J_L \times M_L : j \not\uparrow m \text{ et } j \not\downarrow m\} \end{cases} \quad (2.3)$$

Définition 2.17 (Table de treillis). *La table T d'un treillis $L = (S, \leq)$ est composée d'une colonne par sup-irréductible, d'une ligne par inf-irréductible. Pour $m \in M_L$ et $j \in J_L$, $T[m, j]$ contient $\times, \uparrow, \downarrow, \updownarrow$ ou \circ selon que (m, j) appartient à $P_{\leq}, P_{\uparrow}, P_{\downarrow}, P_{\updownarrow}$ ou P_{\circ} .*

$$T[m, j] = \begin{cases} \times & \text{si } (m, j) \in P_{\leq} \\ \uparrow & \text{si } (m, j) \in P_{\uparrow} \\ \downarrow & \text{si } (m, j) \in P_{\downarrow} \\ \updownarrow & \text{si } (m, j) \in P_{\updownarrow} \\ \circ & \text{si } (m, j) \in P_{\circ} \end{cases} \quad (2.4)$$

	v_1	v_2	v_3	v_4	v_5	v_6
v_2	↓	×	○	○	↓	↕
v_3	×	↕	×	↑	↕	↕
v_4	↕	↓	○	×	×	×
v_9	×	↕	↕	×	×	×
v_{11}	×	↓	↓	↕	×	×
v_{12}	↕	×	↑	↑	↕	×
v_{13}	×	↓	↕	○	↕	×
v_{14}	×	↓	↕	○	×	↕

Tableau 2.1 – Table de treillis de la figure 2.4

Algorithme 1 Algorithme de Génération de la table d'un treillis

Entrée : Un treillis $L = (S, \leq)$ et son diagramme de Hasse \prec

Sortie : La table T du treillis L

1 : soit J l'ensemble des sup-irréductibles (un seul arc entrant) ;
2 : soit M l'ensemble des inf-irréductibles (un seul arc sortant) ;
3 : initialiser une table T avec J en lignes et M en colonnes ;
4 : **tant que** $(j, m) \in J \times M$ **faire**
5 : **si** $j \leq m$ **alors**
6 : $T[m, j] = \times$;
7 : **sinon**
8 : soit j^- l'unique prédécesseur de j selon \prec ;
9 : soit m^+ l'unique successeur de m selon \prec ;
10 : **si** $j \leq m^+$ et $j^- \leq m$ **alors**
11 : $T[m, j] = \updownarrow$;
12 : **sinon**
13 : **si** $j \leq m^+$ et $j^- \not\leq m$ **alors**
14 : $T[m, j] = \uparrow$;
15 : **sinon**
16 : **si** $j \not\leq m^+$ et $j^- \leq m$ **alors**
17 : $T[m, j] = \downarrow$;
18 : **sinon**
19 : **si** $j \not\leq m^+$ et $j^- \not\leq m$ **alors**
20 : $T[m, j] = \circ$;
21 : **fin si**
22 : **fin tant que**
23 : **Retourner** T

Définition 2.18 (Table binaire). *La table T d'un treillis $L = (S, \leq)$ est une table binaire $T[m, j]$ contenant \times , si $(m, j) \in P_{\leq}$, et rien sinon.*

$$T[m, j] = \begin{cases} \times & \text{si } (m, j) \in P_{\leq} \\ \text{rien} & \text{si } (m, j) \in P_{\not\leq} \end{cases} \quad (2.5)$$

	v_1	v_2	v_3	v_4	v_5	v_6
v_2		×				
v_3	×		×			
v_4				×	×	×
v_9	×			×	×	×
v_{11}	×				×	×
v_{12}		×				×
v_{13}	×					×
v_{14}	×				×	

Tableau 2.2 – Table binaire extrait du tableau 2.1

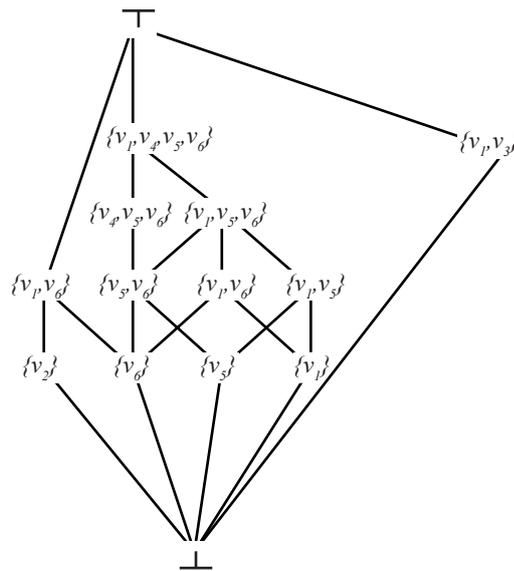


Figure 2.9 – Diagramme de Hasse du treillis de la figure 2.4

2.4 Treillis des fermés, système de fermetures

Une *famille de Moore* est une famille \mathcal{F} de parties - dites *fermés* - d'un ensemble S stable par intersection et contenant S lui-même. (\mathcal{F}, \subseteq) est un treillis.

2.4.1 Treillis des fermés

Soit S un ensemble non vide, toute famille $\mathcal{F} \in \mathcal{P}(S)$ munie de la relation \subseteq est ordonnée, et aussi \subseteq est *transitive*, *réflexive* et *antisymétrique* (CF. Figures 2.10). Le treillis possède les propriétés d'une famille de Moore (Moore, 1909).

Définition 2.19 (Treillis des fermés). *Un treillis des fermés sur un ensemble S est une paire (\mathcal{F}, \subseteq) où \mathcal{F} est une famille sur S possédant les propriétés d'une famille de Moore, encore appelée famille des fermés :*

- \mathcal{F} contient S
- \mathcal{F} est stable par intersection : pour tous $F, F' \in \mathcal{F}$, on a $F \cap F' \in \mathcal{F}$

En particulier, $\mathcal{F} = \mathcal{P}(S)$ est une famille de Moore, et la structure de treillis induite $(\mathcal{P}(S), \subseteq)$ est un treillis booléen.

Une hiérarchie est une famille de Moore \mathcal{F} sur S contenant S lui-même, ainsi que tous les singletons, et telle que, pour F et F' parties de la famille, la propriété suivante est vérifiée : $F \cap F' = \emptyset$ ou bien $F \subseteq F'$ ou bien $F' \subseteq F$.

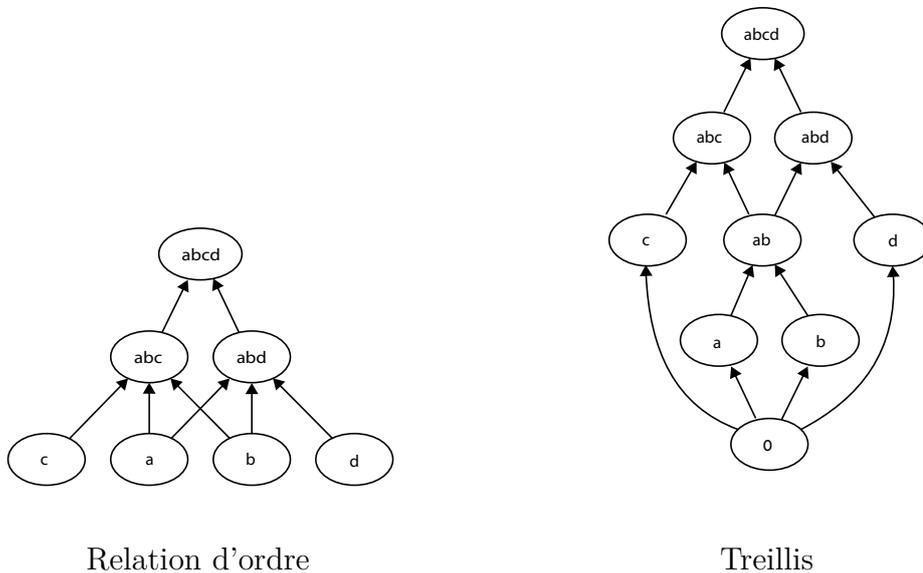


Figure 2.10 – Deux familles ordonnées par inclusion

Les propriétés d'une famille de Moore sur un ensemble S sont purement ensemblistes (stabilité par intersection et appartenance de l'ensemble S lui-même), ses propriétés latticielles permettent d'introduire les opérateurs de bornes inférieures \wedge et supérieures \vee , pour F et F' parties de \mathcal{F} :

$$F \wedge F' = F \cap F'.$$

$$F \vee F' = \bigcap \{F' \in \mathcal{F} : F \cup F' \subseteq F'\}$$

Treillis et treillis des fermés sont en correspondances bijectives.

2.4.2 Système de fermeture

Définition 2.20 (Système de fermeture). *Un système de fermeture (closure) est une paire $C = (S, \varphi)$, avec S un ensemble et φ un opérateur de fermeture, i.e. une application définie sur $\mathcal{P}(S)$ à la fois isotone, extensive et idempotente*

- *isotonie* : $X \subseteq Y \Rightarrow \varphi(X) \subseteq \varphi(Y)$.
- *extensivité* : $X \subseteq \varphi(X)$.
- *idempotence* : $\varphi(\varphi(X)) = \varphi(X)$.

Pour un système de fermeture (S, φ) , pour une partie $X \subseteq S$ quelconque, $\varphi(X)$ est appelée la fermeture de X , ou encore un fermé de C . Il est facile de vérifier que l'ensemble des fermés forme une famille de Moore \mathcal{F}_C :

$$\mathcal{F}_C = \{\varphi(X) : X \subseteq S\}.$$

La famille des fermés \mathcal{F}_C possédant la propriété d'une famille de Moore, à tout système de fermeture C , on associe son treillis des fermés (\mathcal{F}, \subseteq) . Dualelement, à tout treillis de fermés (\mathcal{F}, \subseteq) sur S , on associe un système de fermeture dont l'opérateur de fermeture φ est défini, pour $X \subseteq S$, par : $\varphi(X) = \bigcap \{F \in \mathcal{F} : X \subseteq F\}$.

Les deux bornes \vee et \wedge se redéfinissent via l'opérateur de fermeture, pour X et Y parties de S , par :

$$\varphi(X) \wedge \varphi(Y) = \varphi(X) \cap \varphi(Y).$$

$$\varphi(X) \vee \varphi(Y) = \varphi(\varphi(X) \cup \varphi(Y)).$$

Treillis et systèmes de fermeture sont en correspondances bijectives.

A tout treillis $L = (S, \leq)$, on associe le système de fermeture $C = (J_L, \varphi)$, où pour $x \in S$, $\varphi(x) = J_x$. La minimalité de la famille de Moore sous-jacente $(J_x)_{x \in S}$ permet de définir la notion de système de fermeture réduit.

Définition 2.21 (Système de fermeture réduit). *Un système de fermeture $C = (S, \varphi)$ est réduit lorsque l'ensemble S sur lequel il est défini ne contient que les éléments dont les fermés sont des sup-irréductibles de son treillis fermés :*

$$\forall x \in S, \forall Y \subseteq S \text{ tel que } x \notin Y, \text{ on a } \varphi(x) \neq \varphi(Y)$$

2.5 Correspondances de Galois

Les opérateurs de fermeture se trouvent dans de nombreux domaines, comme en logique, en topologie, en bases de données, en combinatoire, ou encore en analyse de données. Lorsque les données s'organisent sous forme d'une table binaire, appelée *contexte*, ou d'un système de règles d'implications du type « *si...alors...* ». Dans la suite, nous avons choisi de présenter le treillis des concepts d'un contexte, et le treillis de fermés d'un système d'implications comme deux cas d'usage des systèmes de fermeture, car leur existence repose sur celle d'opérateurs de fermeture. En effet, la correspondance de Galois entre les objets et les attributs qui définit tout contexte induit l'existence de deux opérateurs de fermeture, le premier défini sur l'espace des objets (individus) et le second sur l'espace des attributs (variables).

[Birkhoff \(1940\)](#) a mis en évidence la théorie des ensembles ordonnés en 1940 sous le nom d'applications polarisées. Les correspondances de Galois sont un objet fondamental dans cette théorie. Le terme correspondances de Galois fut apparu en 1944 par [Öre \(1944\)](#). La correspondance de Galois est la base de la construction des éléments de treillis de Galois. Ces éléments sont aussi appelés les couples complets du treillis ou les concepts ([Wille, 1982](#)).

2.5.1 Applications galoisiennes

Définition 2.22 (Partition). $\mathcal{P}(S)$ l'ensemble des partitions de S , i.e. $\mathcal{P}(S)$ c'est l'ensemble de tous les sous ensembles de S .

Définition 2.23 (Correspondance de Galois). *Soient S et S' deux ensembles ordonnés et $f : S \rightarrow S'$ et $g : S' \rightarrow S$ deux applications. Le couple d'application (f, g) sera dit correspondance de Galois entre S et S' si, pour tous $x, x' \in S$, $y, y' \in S'$, les trois conditions suivantes sont vérifiées :*

(G1) f antitone, i.e. $x \leq x'$ implique $f(x) \geq f(x')$;

(G2) g antitone, i.e. $y \leq y'$ implique $g(y) \geq g(y')$;

(G3) $f \circ g$ et $g \circ f$ extensive, i.e. $x \leq g \circ f(x)$ et $y \leq f \circ g(y)$.

L'application f (resp. g) est appelée application galoisienne de S vers S' (resp. de S' vers S).

Proposition 2.2. Soient (S, \leq) , (S', \leq) deux ensembles ordonnés et $f : S \rightarrow S'$ et $g : S' \rightarrow S$ deux applications. Le couple d'application (f, g) est une correspondance de Galois, si et seulement si :

(G4) pour tous $x \in S$ et $y \in S'$, $x \leq g(y) \Leftrightarrow y \leq f(x)$

Proposition 2.3. Une correspondance de Galois (f, g) est un couple involutif, c'est-à-dire : $f = f \circ g \circ f$ et $g = g \circ f \circ g$

Démonstration. Par (G3), on a $f(x) \leq fgf(x')$ et $x \leq fg(x)$ donc, par antitonie de f , $f(x) \geq fgf(x)$. La preuve pour $gfg(x') = g(x')$ est similaire. \square

Proposition 2.4. Soit (f, g) une correspondance de Galois entre deux ensembles S et S' . Notons $\phi = g \circ f$ et $\phi' = f \circ g$. Les applications ϕ et ϕ' sont des opérateurs de fermetures respectivement sur S et S' .

Définition 2.24 (Application résiduée/résiduelle). Soient S et S' deux ensembles ordonnés et $f : S \rightarrow S'$ et $g : S' \rightarrow S$ deux applications. L'application f (resp. g) sera dite application résiduée (resp. résiduelle) si pour tous $x, x' \in S$, $y, y' \in S'$, les trois conditions suivantes sont vérifiées :

(R1) f isotone, i.e. $x \leq x'$ implique $f(x) \leq f(x')$;

(R2) g isotone, i.e. $y \leq y'$ implique $g(y) \leq g(y')$;

(R3) $f \circ g$ extensive i.e. $x \leq g \circ f(x)$ et $g \circ f$ contractée i.e. $y \geq f \circ g(y)$.

Proposition 2.5. Une application f de S vers S' (resp. g de S' vers S) est résiduée (resp. résiduelle) si et seulement si f (resp. g) est une application galoisienne de S dans S'^d (resp. S^d vers S), où S^d est le dual de S' .

2.5.2 Correspondances de Galois associées à une relation binaire

Soient S et S' deux ensembles finis et \mathcal{R} une relation binaire de S vers S' .

Soient les deux fonctions $f_{\mathcal{R}}$ et $g_{\mathcal{R}}$ définies comme suit :

$$f_{\mathcal{R}} : \mathcal{P}(S) \longrightarrow \mathcal{P}(S')$$

$$X \longmapsto f_{\mathcal{R}}(X) = \bigcap_{x \in X} \{y \in S' : x \mathcal{R} y\} = \{y \in S' : \text{pour tout } x \in X, x \mathcal{R} y\}$$

$$g_{\mathcal{R}} : \mathcal{P}(S') \longrightarrow \mathcal{P}(S)$$

$$Y \longmapsto g_{\mathcal{R}}(Y) = \bigcap_{y \in Y} \{x \in S : x \mathcal{R} y\} = \{x \in S : \text{pour tout } y \in Y, x \mathcal{R} y\}$$

Théorème 2.1. (Ganter et Wille, 1999)

- Le couple $(f_{\mathcal{R}}, g_{\mathcal{R}})$ est une correspondance de Galois.
- Réciproquement, si (f, g) est une correspondance de Galois entre les ensembles $\mathcal{P}(S)$ et $\mathcal{P}(S')$ alors $\mathcal{R}_{(f,g)} = \{(x, y) \in S \times S' : x \in g(\{y\})\} = \{(x, y) \in S \times S' : y \in f(\{x\})\}$ est une relation binaire de S vers S' .
- Par ailleurs, nous avons les égalités suivantes :
 $f_{\mathcal{R}_{(f,g)}} = f$, $g_{\mathcal{R}_{(f,g)}} = g$ et $\mathcal{R}_{(f_{\mathcal{R}}, g_{\mathcal{R}})} = \mathcal{R}$.

Selon la Proposition 2.4, on en déduit le corollaire suivant :

Corollaire 2.1. Les applications $\phi = g_{\mathcal{R}} \circ f_{\mathcal{R}}$ et $\phi' = f_{\mathcal{R}} \circ g_{\mathcal{R}}$ sont des opérateurs de fermetures respectivement sur $\mathcal{P}(S)$ et $\mathcal{P}(S')$.

Ce type de correspondance de Galois est utile en analyse formelle de concepts (AFC). L'AFC fournit un cadre théorique pour la fouille des règles d'association dans un contexte binaire. De ce fait, l'ensemble S désigne un ensemble fini d'individus (entité) et S' un ensemble fini de variables (attributs).

Définition 2.25. Un contexte formel est un triplet $\mathbb{K} = (S, S', \mathcal{R})$, où S est un ensemble fini d'individus, S' un ensemble fini de variables et \mathcal{R} est une relation binaire de S vers S' .

- Soit $(X, Y) \in \mathcal{R}$. Le couple (X, Y) sera dit concept formel si X est un fermé de ϕ (i.e. $\phi(X) = X$) et $Y = f_{\mathcal{R}}(X)$ avec $\phi = g_{\mathcal{R}} \circ f_{\mathcal{R}}$.

Exemple 2.5.1. Le Tableau 2.3 présente un contexte formel avec $\mathcal{S} = \{X_1, X_2, X_3, X_4, X_5, X_6\}$ et $\mathcal{S}' = \{Y_1, Y_2, Y_3, Y_4, Y_5\}$. Dans ce tableau, on écrit $(X, Y) \in \mathcal{R}$ lorsqu'une case contient \times , par exemple, $(X_1, Y_1) \in \mathcal{R}$, $(X_1, Y_2) \notin \mathcal{R}$, $(X_1, Y_3) \in \mathcal{R}$, $(X_1, Y_4) \in \mathcal{R}$, $(X_1, Y_5) \notin \mathcal{R}$, $(X_2, Y_1) \notin \mathcal{R}$, $(X_2, Y_2) \in \mathcal{R}, \dots$

\mathcal{S}	\mathcal{S}'	Y_1	Y_2	Y_3	Y_4	Y_5
X_1		×		×	×	
X_2			×	×		×
X_3		×	×	×		×
X_4			×			×
X_5		×	×	×		×
X_6			×	×		×

Tableau 2.3 – Contexte formael

2.6 Système implicatif

En logique, ou aussi en fouille de données, les *systèmes implicatifs*, ou *systèmes de règles d'implication*, sont utilisés pour exprimer des implications entre des données.

Définition 2.26 (Système implicatif). *Un système implicatif Σ sur un ensemble d'éléments S est une relation binaire définie sur $\mathcal{P}(S)$. Une règle d'implication est un couple $(A, B) \in \Sigma$ notée $A \rightarrow B$ et signifiant «A implique B». Dans ce sens, A est la prémisses de la règle, et B sa conclusion.*

A tout système implicatif Σ , on associe la famille des parties de S fermées pour les règles d'implication : $\mathcal{F}_\Sigma = \{X \subseteq S : A \subseteq X \text{ et } A \rightarrow B \text{ impliquent } B \subseteq X \text{ pour tout } A \rightarrow B \in \Sigma\}$. La famille \mathcal{F}_Σ possède les propriétés d'une famille de Moore (stabilité par intersection, et appartenance de l'ensemble S lui-même), alors $(\mathcal{F}_\Sigma, \subseteq)$ forme un treillis.

Deux systèmes implicatifs différents Σ et Σ' peuvent engendrer deux familles de Moore isomorphes, i.e $\mathcal{F}_\Sigma \cong \mathcal{F}_{\Sigma'}$.

Définition 2.27 (Système implicatif réduit). *Un système implicatif est dit réduit lorsque la famille de Moore qu'il engendre est minimale, i.e. la cardinalité de l'ensemble S correspond aux sup-irréductibles du treillis de fermés.*

Réduire un système implicatif, c'est remplacer dans chaque règle, chaque élément réductible par ses irréductibles équivalents.

Définition 2.28 (Système implicatif équivalent). *Deux systèmes implicatifs Σ et Σ' sont dits équivalents lorsqu'ils engendrent deux familles de Moore identiques, i.e. $\mathcal{F}_\Sigma = \mathcal{F}_{\Sigma'}$.*

Définition 2.29 (Système implicatif unaire). *Un système implicatif Σ est dit unaire lorsque chacune de ses règles possède un singleton en conclusion. On peut définir Σ sur $\mathcal{P}(S) \times S$, et écrire $B \rightarrow x$ au lieu de $B \rightarrow \{x\}$, avec $B \subseteq S$, et $x \in S$.*

Rendre un système **implicatif Σ unaire**, c'est remplacer toute règle $B \rightarrow X = \{x_1, x_2, \dots, x_n\} \in \Sigma$ par les n règles unaires $B \rightarrow x_i$ pour $i \leq n$.

Définition 2.30 (Système implicatif compact). *Un système implicatif Σ est dit compact lorsque chacune de ses règles est composée d'une conclusion de cardinalité maximale, ou de façon équivalente, s'il n'existe pas deux règles distinctes de même prémisse.*

Rendre **compact un système implicatif Σ** , c'est remplacer les n règles de même prémisse B par une seule règle dont la prémisse est B , et la conclusion est l'union des conclusions des n règles.

Définition 2.31 (Système implicatif propre). *Un système implicatif Σ est propre lorsque la prémisse et la conclusion de chacune de ses règles ne s'intersectent pas.*

Rendre **propre un système implicatif Σ** , c'est remplacer chaque règle $B \rightarrow X$ par la règle propre $B \rightarrow X \setminus B$.

Définition 2.32 (Système implicatif maximal). *Un système implicatif Σ est dit maximal à droite lorsqu'il est compact, et que la conclusion de chacune de ses règles de prémisse X est égale à $\varphi(X) \setminus X$.*

Définition 2.33 (Système implicatif minimal). *Un système implicatif Σ est dit minimal à gauche lorsqu'il est compact, et qu'il n'existe pas deux règles de même prémisse.*

Définition 2.34 (Système implicatif non redondant). *Un système implicatif Σ est non redondant si aucune règle ne peut être supprimée, i.e. pour toute règle $X \rightarrow Y \in \Sigma$, $\Sigma \setminus \{X \rightarrow Y\}$ n'est pas équivalent à Σ .*

Définition 2.35 (Système implicatif complet). *Un système implicatif complet Σ est un système implicatif satisfaisant les conditions suivantes :*

- (S1) $B \subseteq A$ implique $A \rightarrow B$.
- (S2) Pour tous $A, B, C \subseteq S$, $A \rightarrow B$ et $B \rightarrow C$ impliquent $A \rightarrow C$ (transitivité).

- (S3) Pour tous $A, B, C, D \subseteq S$, $A \rightarrow B$ et $C \rightarrow D$ impliquent $A \cup C \rightarrow B \cup D$.

Un système implicatif complet peut être vu comme un préordre sur $\mathcal{P}(S)$, compatible avec \supseteq et union stable.

2.7 Hiérarchies

Plusieurs méthodes issues de l'AFC sont proposées dans un objectif de classification. Les hiérarchies ont une place importante en classification, car elles correspondent aux arbres de classifications. Dans cette partie, nous nous référons aux travaux de [Lerman \(1981\)](#); [Lerman et Ghazzali \(1991\)](#)

Définition 2.36 (Hiérarchie). Une hiérarchie \mathcal{H} sur S est un ensemble de sous-ensemble de S (clusters) satisfaisant les conditions :

- (H1) $\emptyset \notin \mathcal{H}$
- (H2) $S \in \mathcal{H}$
- (H3) pour tout $s \in S$, $\{s\} \in \mathcal{H}$; (fermeture des singletons)
- (H4) pour $h, h' \in \mathcal{H}$, $h \cap h' \in \{\emptyset, h, h'\}$. (arborescence)

Définition 2.37. Une hiérarchie de parties $\mathcal{H}(S)$ de S est un ensemble de parties de S telles que :

- $S \in \mathcal{H}(S)$
- $\forall x \in S, \{x\} \in \mathcal{H}(S)$
- $\forall (X \in \mathcal{H}(S), Y \in \mathcal{H}(S)), X \cap Y = X$ ou $X \cap Y = Y$ ou $X \cap Y = \emptyset$

Cette hiérarchie de partie est *binnaire* si toute partie non réduite à singleton peut être obtenue comme la réunion de deux parties disjointes de la hiérarchie, et est notée $\mathcal{H}_b(S)$.

Exemple 2.7.1. Posons $S = \{1, 2, 3, 4, 5, 6, 7, 8\}$, un exemple d'une hiérarchie binnaire de partie est donné par :

$$\begin{aligned} \mathcal{H}_b(S) = \{ & \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}, \{3, 5\}, \{4, 8\}, \{1, 3, 5\}, \\ & \{2, 4, 8\}, \{1, 3, 5, 6\}, \{1, 2, 3, 4, 5, 6, 8\}, \{1, 2, 3, 4, 5, 6, 7, 8\} \end{aligned}$$

Soit un ensemble totalement ordonné de valeurs quelconque qui comprend une échelle d'intervalle, où les différences numériquement, et peuvent être comparées. Généralement, cet ensemble est \mathbb{R}_+ ou \mathbb{N} . Une application ν de $\mathcal{H}(S)$ dans \mathbb{N} , strictement croissante par rapport à la relation d'inclusion dans $\mathcal{H}(S)$, définit une indexation de $\mathcal{H}(S)$. Le couple $(\mathcal{H}(S), \nu)$ est une hiérarchie indicée strictement croissante, où $\nu = 0$ lorsque chacune des parties contenant un seul élément (singleton).

Sur l'ensemble $\mathcal{H}(S)$ on définit un graphe orienté et valué, correspondant à la relation d'inclusion entre parties de $\mathcal{H}(S)$. On note $(\mathcal{H}(S), \Gamma, \mu)$ un tel graphe, où Γ définit l'ensemble des arcs du graphe et μ est la valuation (à partir d'une mesure). Si $(X, Y) \in \mathcal{H}(S)$, on a $\Gamma(X, Y)$ si et seulement si :

1. $X \subset Y$ strictement
2. Il n'existe pas Z de $\mathcal{H}(S)$, tel qu'on ait strictement $X \subset Z \subset Y$

La valuation $\mu(X, Y)$ de l'arc (X, Y) est définie par la différence $\nu(X) - \nu(Y)$, où ν représente l'indexation définie de $\mathcal{H}(S)$. La mise en niveaux de ce graphe permet de le représenter par un *dendrogramme*.

Exemple 2.7.2. Par rapport à $\mathcal{H}_b(S)$ de l'exemple 2.7.1, on a une indexation ordinaire ν définie par : $\forall x, \nu(\{x\}) = 0$.

$$\begin{aligned} \nu(\{3, 5\}) &= 1, \nu(\{4, 8\}) = 4, \nu(\{1, 3, 5\}) = 2, \\ \nu(\{2, 4, 8\}) &= 5, \nu(\{1, 3, 5, 6\}) = 3, \nu(\{1, 2, 3, 4, 5, 6, 8\}) = 6, \\ \nu(\{1, 2, 3, 4, 5, 6, 7, 8\}) &= 7 \end{aligned}$$

Nous avons le dendrogramme (Cf. Figure 2.11) ci-dessous :

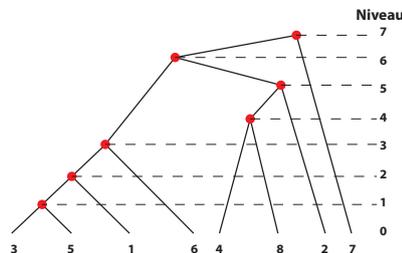


Figure 2.11 – Dendrogramme ordonné

L'application ν n'est pas nécessairement injective. Elle doit être seulement strictement croissante pour toute suite totalement ordonnée par inclusion de parties de

$\mathcal{H}(S)$. Ainsi, au niveau de la représentation ν est strictement croissante le long de tout chemin ascendant du diagramme d'arbre. Dans l'exemple précédent où il s'agit d'une hiérarchie binaire, on aurait pu avoir, au lieu de $\nu(4, 8) = 4$, $\nu(4, 8) = 2$.

2.7.1 Chaîne de partitions et arbre de classification associée

Une partition P de S est une famille de parties de S mutuellement disjointes et complémentaires. Ainsi, chaque élément de S appartient à une partie et une seule. Nous notons $P = \{S_i | 1 \leq i \leq k\}$, où S_i désigne la i -ème classe de partition, $1 \leq i \leq k$. Soit $\mathcal{P}(S) \subset S$, $(\mathcal{P}(S), <)$, où $<$ est une relation de finesse entre partition. Pour cette relation $<$, si P et Q sont deux partitions de S , on a $P < Q$ si les classes de Q se déduisent des classes de la partition de P par union. On trouve que $\mathcal{P}(S)$ a une structure de treillis (Barbut et Monjardet, 1970; Lerman, 1970). Une chaîne totalement ordonnée de partitions $(P_1, P_2, \dots, P_j, P_{j+1}, \dots, P_l)$ est telle que la partition P_j est moins fine que P_{j-1} avec $P_j = \{S_i^j | 1 \leq i \leq k_j\}$.

Nous nous intéressons aux chaînes totalement ordonnées de partitions :

$$(P_0, P_1, P_2, \dots, P_j, P_{j+1}, \dots, P_l, P_{l+1}) \quad (2.6)$$

où P_0 est la partition *discrète* comportant n classes singletons et où P_{l+1} est la partition *grossière* à une seule classe S . Considérons maintenant la famille des classes étiquetées de la suite des partitions : $\{S_i^j | 0 \leq j \leq l + 1, 1 \leq i \leq k_j\}$. Sur cette famille de classes étiquetées $(k_0 + k_1 + \dots + k_j + k_{j+1} + \dots + k_l + k_{l+1})$, on définit un graphe orienté, obtenu à partir de ses restrictions sur la famille des classes étiquetées de deux partitions consécutives P_j et P_{j+1} , $0 \leq j \leq l$. Un arc (S_i^j, S_i^{j+1}) signifie l'inclusion au sens large $S_i^j \subset S_i^{j+1}$. Chaque restriction définit un graphe biparti.

Définition 2.38. *Un arbre de classification sur S indicé par la fonction de niveau est la représentation graphique du graphe que nous venons de définir. Les classes S_i^j , $1 \leq i \leq k_j$, sont représentées par des sommets (souvent appelés nœuds) qui occupent le j ème niveau, $0 \leq j \leq l + 1$. Ainsi, P_0 occupe le niveau des feuilles de l'arbre et P_{l+1} , celui de la racine de l'arbre. Chaque classe S_i^{j+1} de P_{j+1} est représentée au niveau $j + 1$ par un nœud qui se trouve à la jonction de branches issues des nœuds du niveau j , correspondants aux classes dont la réunion est S_i^{j+1} ; la disposition latérale de ces derniers nœuds étant telle que ces branches ne se croisent pas, $0 \leq j \leq l$.*

Les représentations en termes de dendrogramme indicé ordinalement et d'arbre de classification indicé par la fonction niveau sont équivalentes et il est aisé de passer de l'une des représentations à l'autre. Ainsi, en coupant le dendrogramme horizontalement entre le niveau j et $j + 1$, on obtient la partition P_j . Les classes de cette dernière sous-tendent les nœuds directement rencontrés en dessous de cette horizontale. Inversement, en partant d'un arbre de classification indicé par la fonction niveau, on obtient le dendrogramme correspondant en remplaçant toute suite d'arcs dont seuls le sommet origine et le sommet extrémité sont distincts, par un seul arc reliant le sommet origine à celui extrémité. Nous donnons ci-dessous l'arbre de classification correspondant au dendrogramme ci-dessus de la Figure 2.12.

Un arbre de classification est binaire si et seulement si chaque réunion de classes permettant de passer de la partition P_j à la suivante P_{j+1} comporte exactement deux arguments et cela pour tout j , $0 \leq j \leq l$. Cependant, lors du passage entre P_j et P_{j+1} , plusieurs paires de classes sans composante commune peuvent fusionner en même temps. Ainsi pour être explicite, si on considère par exemple que la partition P_j comporte cinq classes : $P_j = \{S_1^j, S_2^j, S_3^j, S_4^j, S_5^j\}$, une partition P_{j+1} telle que :

$$P_{j+1} = \{S_1^j \cup S_2^j, S_3^j, S_4^j \cup S_5^j\} \tag{2.7}$$

résulte d'une fusion binaire ; mais pas comme :

$$P'_{j+1} = \{S_1^j \cup S_2^j, S_3^j, S_4^j \cup S_5^j\} \tag{2.8}$$

car il existe une fusion comportant trois arguments.

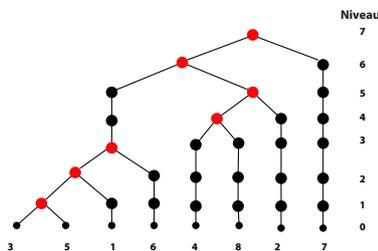


Figure 2.12 – Arbre de classification ordinalement indicé

2.7.2 Construction d'un dendrogramme ordinal compatible avec $\mathcal{H}_b(S)$

A partir d'une hiérarchie binaire de parties $\mathcal{H}_b(S)$, on peut construire un arbre de classification binaire, dont la suite des niveaux est ordinalement indicée.

Soit une chaîne totalement ordonnée de partitions $P_0, P_1, \dots, P_j, P_{j+1}, \dots, P_l, P_{l+1}$ où P_0 est la partition *discrète* comportant n classes singletons et où P_{l+1} est la partition *grossière* à une seule classe S , qui constitue une hiérarchie de parties \mathcal{H}_0 . Une hiérarchie de parties \mathcal{H}_1 est compatible avec \mathcal{H}_0 si $\mathcal{H}_1 \subset \mathcal{H}_0$. On a alors cette définition :

Définition 2.39. *Une hiérarchie de parties $\mathcal{H}(S)$ est compatible avec un arbre de classification sur S si et seulement si chacune des parties de $\mathcal{H}(S)$ se trouve sous-tendant un nœud, une feuille ou la racine de l'arbre de classification.*

Définition 2.40. *Une chaîne complète de parties, éléments de $\mathcal{H}_b(S)$, est une suite $(X_0, X_1, \dots, X_g, X_{g+1}, \dots, X_h, X_{h+1})$ strictement croissante au sens de l'inclusion de parties, éléments de $\mathcal{H}_b(S)$, telle que X_0 est un singleton, $X_{h+1} = S$ et où aucun élément de $\mathcal{H}_b(S)$ ne peut correspondre à une partie strictement comprise entre deux parties X_g et X_{g+1} , $0 \leq g \leq h$.*

Pour tout g , $0 \leq g \leq h$, on a $X_{g+1} = X_g \cup Y$, où $Y \subseteq S$ est un élément de $\mathcal{H}_b(S)$. En effet, X_{g+1} est la réunion d'exactly deux parties, éléments de la hiérarchie. X_{g+1} en est une et $X_{g+1} - X_g = Y$ en est l'autre. X_0 représente la feuille de l'arbre et X_{h+1} sa racine.

Chaque singleton $\{x\}$ ($x \in S$), va donner lieu à une chaîne complète de parties qui peut être construite récursivement. Pour une partie de X_g construite à partir de $\{x\}$, la partie suivante X_{g+1} est définie comme étant celle de $\mathcal{H}_b(S)$ la plus petite contenant strictement X_g , avec $X_{g+1} = X_g \cup Y$.

Désignons par $\mathcal{C} = \{C_x | x \in S\}$. Deux chaînes complètes se rejoignent, i.e. leurs sections finissantes sont identiques.

Soient deux chaînes complètes issues de x et de y avec ($x \neq y$):

$$C_x = (\{x\}, X_1, \dots, X_g, X_{g+1}, \dots, X_{h+1})$$

$$C_y = (\{y\}, Y_1, \dots, Y_j, Y_{j+1}, \dots, X_{k+1}),$$

il existe un indice g_0 et un indice j_0 à partir desquels :

$X_{g_0+1}, \dots, X_{h+1} = Y_{j_0+1}, \dots, Y_{k+1}, X_{h+1} = Y_{k+1} = S$. La section finissante de $g_0 + 1$ de C_x est identique à celle, à partir de $j_0 + 1$ de C_y .

Exemple 2.7.3. *Soit*

$$\begin{aligned} \mathcal{H}_b(S) = \{ & \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}, \{9\}, \{10\}, \{11\}, \{12\}, \{13\}, \{14\}, \{15\}, \\ & \{1, 2\}, \{5, 6\}, \{10, 11\}, \{13, 14\}, \{1, 2, 3\}, \{5, 6, 7\}, \{10, 11, 12\}, \{1, 2, 3, 4\}, \\ & \{5, 6, 7, 8\}, \{5, 6, 7, 8, 9\}, \{10, 11, 12, 13, 14\}, \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, \\ & \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}, \\ & \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\} \} \end{aligned}$$

Les deux chaînes complètes suivantes :

$$\begin{aligned} C_1 = (\{ & 1\}, \{1, 2\}, \{1, 2, 3\}, \{1, 2, 3, 4\}, \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, \\ & \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}, \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}) \\ C_5 = (\{ & 5\}, \{5, 6\}, \{5, 6, 7\}, \{5, 6, 7, 8\}, \{5, 6, 7, 8, 9\}, \{5, 6, 7, 8, 9, 1, 2, 3, 4\}, \\ & \{5, 6, 7, 8, 9, 1, 2, 3, 4, 10, 11, 12, 13, 14\}, \{5, 6, 7, 8, 9, 1, 2, 3, 4, 10, 11, 12, 13, 14, 15\}) \end{aligned}$$

se rejoignent sur la partie $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. On a $g_0 = 4$ et $j_0 = 5$.

Définition 2.41. *Une chaîne complète C' est moins fine d'une chaîne complète C si et seulement si la suite des parties C' , non compris le singleton de départ, est une section finissante de C .*

Ainsi, la chaîne complète :

$$\begin{aligned} (\{ & 4\}, \{1, 2\}, \{1, 2, 3\}, \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, \\ & \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}, \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}) \end{aligned}$$

est moins fine que C_1 ci-dessus.

2.7.2.1 Principe de la construction

C est l'état initial d'une chaîne complète, mais pour des raisons de rapidité calcul, on partira de l'une des chaînes complètes les plus longues (au sens de leur nombre d'éléments), sous la forme :

$C = (X_0 = \{x\}, X_1, \dots, X_g, X_{g+1}, \dots, X_h, X_{h+1} = S)$ avec X_0 et X_{h+1} définiront respectivement une feuille de l'arbre des classifications et sa racine.

$X_1, \dots, X_g, X_{g+1}, \dots, X_h$ définiront des neouds propres de l'arbre de classification.

Ecartons les chaînes complètes comparables à C (moins fine) et dans l'ensemble des autres chaînes complètes considérons celle la plus longue notée :

$$D = (Y_0 = \{y\}, Y_1, \dots, Y_j, Y_{j+1}, \dots, Y_{k+1}, X_{k+1} = S).$$

Imaginons que la jonction des deux chaînes se fasse pour $g = g_0 + 1$ et $j = j_0 + 1$; soit $X_{g_0+1}, \dots, X_h = Y_{j_0+1}, \dots, Y_k$. Il s'agit dans ces conditions d'inscrire la formation des nœuds surplombant Y_1, Y_2, \dots, Y_{j_0} par rapport à ceux, déjà établis et concernant X_1, X_2, \dots, X_{g_0} . Il est entendu que dans cette insertion, un même niveau peut être associé soit à un X_g seul, soit à un Y_j seul, soit enfin à un X_g et à un Y_j . En effet, comme nous l'avons mentionné ci-dessus, l'arbre est binaire mais plus d'une fusion peut apparaître à un niveau donné.

$Y_0 = \{y\}$ va occuper le niveau 0 des feuilles. Nous allons, de façon récursive et de proche en proche étudier le positionnement des nœuds associés aux Y_j par rapport à ceux établis pour les X_g ($1 \leq j \leq j_0, 1 \leq g \leq g_0$) et la transformation de la valeur des niveaux qui en résulte. Relativement aux nœuds associés aux $X_g, 1 \leq g \leq g_0$, considérons la suite croissante des indices : $1, 2, \dots, g, g + 1, \dots, g_0$.

Le nœud associé à Y_1 peut être placé soit dans un intervalle strict de la forme $]g, g+1[$ ($1 \leq g \leq g_0$ où $g = 0$ (resp. g_0)) correspond à l'intervalle avant 1 (resp., après g_0)) soit sur un niveau d'indice g . Dans le cas où Y_1 est placé strictement entre g et $g + 1$, on attribue au nœud associé à Y_1 le niveau $g + 1$ et les niveaux associés à la suite croissante des nœuds relative à $(X_{g+1}, \dots, X_{g_0}, X_{g_0+1}, \dots, X_h, X_{h+1} = S)$ se trouvent augmentés d'une unité. Dans ce cas à chaque niveau il se crée une fusion binaire. Dans le cas où Y_1 est placé au niveau g , on attribue au nœud associé à Y_1 le niveau g et les niveaux associés à la suite croissante des nœuds ci-dessus, restent invariants. Dans ce cas, au niveau g il se produit deux fusions binaires surplombant respectivement X_g et Y_1 .

Dans le premier cas où le niveau du nœud Y_1 est $g + 1$, le choix de Y_2 s'effectue de la même manière que celle de Y_1 ; mais par rapport à la nouvelle suite croissante des indices : $(g + 2, g + 3, \dots, g_0 + 1)$.

Dans le second cas où le niveau du nœud Y_1 est g , le choix de Y_2 s'effectue de la même manière que celle de Y_1 ; mais par rapport à la nouvelle suite : $(g + 1, g + 2, \dots, g_0)$.

On peut remarquer que l'ensemble X_{g_0} défini avant le placement de Y_1 sous-tend, après ce placement, le nœud $g_0 + 1$ dans le premier cas et celui g_0 dans le second. Le placement de Y_{j+1} après celui de Y_j suit exactement le même processus, $1 \leq j \leq j_0 - 1$. La hauteur totale de l'arbre binaire obtenu jusqu'à la racine est comprise entre $h + 1$ et $j_0 + h + 1$. La valeur $h + 1$ concerne le cas où Y_1, Y_2, \dots, Y_{j_0} se placent tous à des niveaux déjà occupés par des nœuds existants associé aux X_g . Celle $j_0 + h + 1$,

concerne le cas où Y_1, Y_2, \dots, Y_{j_0} se placent tous à des niveaux différents de ceux déjà occupés par un nœud.

Définition 2.42. C, C' et C'' étant trois chaînes complètes et mutuellement incomparables au sens de la Définition 2.41, C est dite plus proche de C' que de C'' si le premier nœud de C à se retrouver dans C' est de rang plus petit que celui à se retrouver dans C'' .

2.7.3 Hiérarchie de fourches, hiérarchie binaire de parties et arbre binaire de classification

Par rapport à ce qui existe dans la littérature concernant une relation symétrique sur S (notion de hiérarchie de parties), la notion de hiérarchie binaire est la plus proche de celle visée. Néanmoins, il faut définir une autre notion de hiérarchie binaire de parties qui conduira directement à la notion de hiérarchie implicative. Il s'agira de ce que nous appellerons une hiérarchie de fourches.

Définition 2.43. Une fourche de S est une paire $\{X, Y\}$ de parties disjointes de S , où X et Y sont des parties propres de S : $S : X, Y, X \neq \emptyset, Y \neq \emptyset$ et $X \cap Y = \emptyset$.

Définition 2.44. Le sommet Z d'une fourche $\{X, Y\}$ de S est le sous ensemble $X \cup Y$.

Une fourche est figurée par le sommet Z représentant $X \cup Y$ duquel sont issues de façon descendante deux branches reliant le sommet précédent à deux sommets représentant respectivement X et Y , à la manière de la figure 2.13 ci-dessous :



Figure 2.13 – Fourche non orientée

Définissons maintenant la notion de hiérarchie de fourches $\mathcal{HF}(S)$ et celle associée, de hiérarchie complète de fourches $\mathcal{H}_c\mathcal{F}(S)$ sur S .

Définition 2.45. Une hiérarchie de fourches $\mathcal{HF}(S)$ est un ensemble de fourches telles que si $\{X, Y\}$ est une fourche et si X (resp., Y) n'est pas un singleton, c'est qu'il s'agit du sommet d'une fourche. D'autre part, si $\{X, Y\}$ et $\{Z, T\}$ sont deux fourches pour lesquelles – sans restreindre la généralité – $\text{card}(X \cup Y) \leq \text{card}(Z \cup T)$, on a l'un des cas suivants :

- $(X \cup Y) \cap (Z \cup T) = \emptyset$.
- $X \cup Y \subset Z$ ou $X \cup Y \subset T$.

Propriété 2.3. Si Z est le sommet d'une fourche, la fourche de sommet Z est unique

Démonstration. Soit $\{X, Y\}$ une fourche de sommet Z . s'il existait une autre fourche $\{X', Y'\}$, on aurait : $\text{card}(X' \cup Y') = \text{card}(X \cup Y) = \text{card}(Z)$.

Comme $\text{card}(X' \cup Y') \cap \text{card}(X \cup Y) = \emptyset$, selon la définition 2.45 tiret-2, l'un des deux cas suivants : $X' \cup Y' \subset X$ ou $X' \cup Y' \subset Y$. Or chacun de ces deux cas est impossible □

Définition 2.46. Une hiérarchie complète de fourches $\mathcal{H}_c\mathcal{F}(S)$ est une hiérarchie de fourches $\mathcal{HF}(S)$ (Cf. Définition 2.45) pour laquelle on a les deux conditions additionnelles suivantes :

- Pour tout singleton $\{x\} (x \in S)$, il existe au moins une fourche dont l'une des composantes est ce singleton ;
- Il existe une fourche $\{X, Y\}$ pour laquelle $X \cup Y = S$.

Associons à une hiérarchie binaire $\mathcal{H}_b(S)$ celle qu'on obtient en enlevant les singletons (parties réduites à un élément) et que nous notons $\mathcal{H}_b^{-s}(S)$:

$$\mathcal{H}_b^{-s}(S) = \mathcal{H}_b(S) - \{\{x\} | x \in S\} \quad (2.9)$$

On peut de façon naturelle associer à une telle hiérarchie binaire de la forme $\mathcal{H}_b^{-s}(S)$, une hiérarchie complète de fourches $\mathcal{H}_c\mathcal{F}(S)$. En effet, par définition, chaque partie Z de S ($Z \subset S$), non réduite à un singleton, appartenant à $\mathcal{H}_b(S)$, est exactement la réunion de deux parties disjointes X et Y (sous ensembles propres de S). Ainsi, nous associons à Z , la fourche $\{X, Y\}$ dont Z représente le sommet et X et Y les nœuds pendants.

Inversement, à une hiérarchie complète de fourches de la forme $\mathcal{H}_c\mathcal{F}(S)$, on peut associer une hiérarchie binaire de la forme $\mathcal{H}_b^{-s}(S)$. En effet, on associe à chaque sommet de $\mathcal{H}_c\mathcal{F}(S)$, la partie Z qu'il représente. On a bien que Z peut se mettre sous la forme

$$Z = X \cup Y \tag{2.10}$$

où X (resp. Y) est le sommet d'une fourche s'il ne s'agit pas d'un singleton (Cf. Définition 2.4). En désignant par \mathcal{F} la correspondance ainsi définie entre l'ensemble $\mathcal{H}_b^{-s}(S)$ et celui $\mathcal{H}_c\mathcal{F}(S)$ qui lui correspond par construction, on a :

Théorème 2.2. *La correspondance \mathcal{F} est bijective.*

Nous avons construit un **arbre binaire indicé** associé à une *hiérarchie binaire*, pouvant avoir plusieurs *nœuds* à un *niveau* donné. La correspondance bijective (Cf. Théorème 2.2) entre une hiérarchie binaire de la forme $\mathcal{H}_b^{-s}(S)$ et une hiérarchie complète de fourches de la forme $\mathcal{H}_c\mathcal{F}(S)$, autorise la même construction.

2.7.4 Hiérarchie de fourches orientées et hiérarchie implicative

Dans cette partie, nous nous référons au travail de Israël-César LERMAN ([Lerman, 2008](#))

Définition 2.47. *Une fourche orientée de S est définie par la donnée d'un couple (X, Y) de parties propres et disjointes de S remplissant les conditions suivantes :*

- $X \subset S, Y \subset S$
- $X \neq \emptyset, X \neq S, Y \neq \emptyset, Y \neq S$
- $X \cap Y = \emptyset$

Définition 2.48. *Le sommet Z d'une fourche orientée (X, Y) de S représente l'union $X \cup Y$.*

La fourche orientée définie par (X, Y) est représentée par la figure 2.14 suivante :

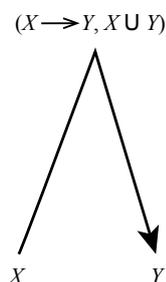


Figure 2.14 – Fourche orientée

L'axiomatique qui régit une *hiérarchie implicative (orientée)* complète est en tout point analogue à celle d'une hiérarchie complète de fourches $\mathcal{H}_c\mathcal{F}(S)$ à cela près qu'au lieu de s'agir d'un ensemble de paires de parties de S , il s'agit d'un ensemble de couples de parties de S . Nous allons dans ces conditions reprendre la Définition 2.45 pour l'adapter au cas orienté.

Définition 2.49. Une hiérarchie implicative $\mathcal{HI}(S)$ sur S est un ensemble de fourches orientées (X, Y) de S telles que si (X, Y) est une fourche orientée et si X (resp., Y) n'est pas un singleton, c'est qu'il s'agit du sommet d'une fourche. D'autre part, si (X, Y) et (Z, T) sont deux fourches pour lesquelles – sans restreindre la généralité – $\text{card}(X \cup Y) \leq \text{card}(Z \cup T)$, on a l'un des cas suivants :

- $(X \cup Y) \cap (Z \cup T) = \emptyset$.
- $X \cup Y \subset Z$ ou $X \cup Y \subset T$.

Propriété 2.4. Si Z est le sommet d'une fourche orientée, la fourche (X, Y) de sommet Z est unique.

Définition 2.50. Une hiérarchie implicative complète de fourches orientées $\mathcal{HI}_c(S)$ est une hiérarchie de fourches orientées pour laquelle on a les deux conditions additionnelles suivantes :

- Pour tout singleton $\{x\}$ ($x \in S$), il existe exactement une fourche orientée dont l'une des composantes est ce singleton ;
- Il existe exactement une fourche orientée (X, Y) pour laquelle $X \cup Y = S$.

Définition 2.51. Une hiérarchie binaire incomplète de parties $\mathcal{H}_{bi}(S)$ est un ensemble de parties de S tel que :

- $(\forall x \in S), \{x\} \in \mathcal{H}_{bi}(S)$
- $\forall (X, Y \in \mathcal{H}_{bi}(S)), X \cap Y = X$ ou $X \cap Y = Y$ ou $X \cap Y = \emptyset$
- Toute partie non réduite à un singleton peut être obtenue comme la réunion d'exac-
tament deux parties disjointes de la hiérarchie.

La donnée d'une hiérarchie binaire incomplète de parties $\mathcal{H}_{bi}(S)$ induit la donnée d'une partition :

$$P = \{S_1, S_2, \dots, S_j, \dots, S_k\} \quad (2.11)$$

de S , telle que la restriction de la hiérarchie $\mathcal{H}_{bi}(S)$ sur chacune des classes S_j , non réduite à un singleton, est une hiérarchie binaire complète $\mathcal{H}_b(S_j)$. Inversement, à partir des différentes hiérarchies complètes $\mathcal{H}_b(S_j)$, $1 \leq j \leq k$, on peut obtenir une hiérarchie binaire incomplète compatible de la forme $\mathcal{H}_{bi}(S)$, en construisant une fonction surjective f de l'ensemble des nœuds des différentes hiérarchies de la forme $\mathcal{H}_b(S_j)$, dans $[1, 2, \dots, m]$ de l'ensemble des entiers. Cette fonction ν est strictement croissante sur chaque chaîne complète de chaque $\mathcal{H}_b(S_j)$. $m \geq$ longueur de la plus longue chaîne complète. Soit une hiérarchie binaire incomplète correspondant à $\mathcal{H}_b(S)$ mais sans singleton, notée $\mathcal{H}_{bi}^{-s}(S)$ et définie par :

$$\mathcal{H}_{bi}^{-s}(S) = \mathcal{H}_{bi}(S) - \{\{x\} | x \in S\} \quad (2.12)$$

Pour une hiérarchie implicative $\mathcal{HI}(S)$, en associant à chaque sommet de l'une de ses fourches orientées (X, Y) , la partie $X \cup Y$ de S , on aura une correspondance bijective avec une hiérarchie binaire incomplète de parties, à laquelle on enlèvera les singletons, de la forme $\mathcal{H}_{bi}^{-s}(S)$.

2.7.5 Ultramétrie

Gras, Kuntz, et Briand (2003) ont considéré cette propriété, relative à l'indice de dissimilarité entre classes adopté dans la construction ascendante d'une hiérarchie implicative. L'indice de dissimilarité entre deux éléments de l'ensemble organisé est égal à l'indice de dissimilarité entre les deux classes composant une fourche et auxquelles appartiennent respectivement les deux éléments. Le caractère monotone

de cet indice le long de la suite des niveaux assure son caractère ultramétrique. Relativement à une hiérarchie implicative ou orientée $\mathcal{HI}(S)$ sur S , nous avons associé à chacune de ses fourches orientées (X, Y) le nœud d'une hiérarchie binaire représentant $X \cup Y$ qui correspond au sommet de la fourche. Une telle hiérarchie est généralement incomplète et a été notée $\mathcal{H}_{bi}(S)$ (Cf. Définition 2.51). Nous avons pu associer à une telle hiérarchie plusieurs fonctions ordinales compatibles de l'ensemble de ses nœuds dans un intervalle commençant $[1, 2, \dots, m]$ de l'ensemble des entiers. Fixons en une, quelconque, que nous notons ν . Cette dernière se présente sous la forme :

$$\nu = \mathcal{H}_{bi}^{-s}(S) \rightarrow [1, 2, \dots, m]. \quad (2.13)$$

Elle attribue à chacun des nœuds, un rang définissant le niveau de sa formation. Soit un nœud N de $\mathcal{H}_{bi}(S)$ surplombant un sous ensemble de la forme $X \cup Y$ où X et Y sous-tendent chacun un nœud ou un singleton de $\mathcal{H}_{bi}(S)$. La valeur de la fonction δ symétrique que nous allons définir sur $S \times S$ est telle que :

$$(\forall (x, y) \in X \times Y), \delta(x, y) = \delta(y, x) = \nu(N) \quad (2.14)$$

Plus précisément, la fonction $\delta : S \times S \rightarrow [1, 2, \dots, m, \infty]$ associe à chaque couple d'éléments (x, y) de $S \times S$, le rang du niveau du premier nœud où la paire d'éléments $\{x, y\}$ se trouve réunie. Dans le cas où l'un des deux éléments x ou y reste seul dans sa classe, ce qui correspond alors à un singleton, c'est la valeur ∞ qui est attribuée par δ . Il est classique et d'ailleurs aisé de voir que - à la valeur ∞ près - δ définit une distance ultramétrique sur S caractérisée par la propriété :

$$(\forall x, y, z \in S), \delta(x, y) \leq \max \{ \delta(x, z), \delta(y, z) \} \quad (2.15)$$

En effet, le niveau de première réunion de x et y ne peut être strictement supérieur à chacun, de x et z d'une part et de y et z d'autre part. À partir de l'équation 2.15 on démontre que tout triangle $\{x, y, z\}$ est isocèle, la base étant le plus petit des côtés. Plus précisément, si on suppose, sans restreindre la généralité, que :

$$\delta(x, y) \leq \delta(x, z) \leq \delta(y, z) \quad (2.16)$$

on établit aisément que : $\delta(x, z) = \delta(y, z)$. Néanmoins, ce qu'on a de plus compte tenu

du caractère binaire de l'arbre, est le caractère strict de la première des inégalités de 2.16, soit : $\delta(x, y) < \delta(x, z) = \delta(y, z)$. on partira de la notion de chaîne complète des parties $\mathcal{H}_{bi}(S)$ qui reprend la Définition 2.40 en enlevant la condition $X_h = S$. $\mathcal{H}_{bi}(S)$ s'obtient en regroupant en tant que ses éléments, l'ensemble des parties de S qui définissent des composantes de chaînes complètes mutuellement incomparables et qui sont directement obtenues à partir de la donnée de la distance ultramétrique δ . L'algorithme est le suivant :

1. Choisir un élément quelconque x de S et établir le vecteur des similarités ultramétriques à x :

$$\{\delta(x, y) | y \in S - \{x\}\} \quad (2.17)$$

2. Trier la famille de ces valeurs et établir la suite strictement croissante des valeurs atteintes que nous notons :

$$0 < \delta_1^x < \delta_2^x < \dots < \delta_k^x < \dots < \delta_l^x \quad (2.18)$$

3. Former la suite croissante des parties qui représentent des cercles ultramétriques :

$$S^x = \{C_k^x = \{y | \delta(x, y) \leq \delta_k^x\} | 1 \leq k \leq l\} \quad (2.19)$$

cette suite de parties est une chaîne complète de $\mathcal{H}_{bi}(S)$ où chaque partie C_k^x représente un neoud de la hiérarchie dont le niveau est δ_k^x .

4. Ayant bâti un ensemble $\{S^x | x \in F \subset S\}$, des chaînes, on choisira, tant que $S - F$ n'est pas vide, pour construire la chaîne de parties S^y dont on abandonnera la section finissante à partir de $C_{k_0}^y$ si l'ensemble défini par $C_{k_0}^y$ a été déjà rencontré précédemment dans l'un des S^x , pour un x de F .

Nous avons interprété une hiérarchie binaire incomplète $\mathcal{H}_{bi}(S)$ comme une famille de fourches (non orientées), qu'une hiérarchie implicative $\mathcal{HI}(S)$ pouvait s'interpréter comme une famille de fourches orientées (Cf. Définition 2.49 et Figure 2.14). Maintenant, nous voulons voir le correspondant adéquat de la notion d'ultramétrie et la propriété correspondante du caractère isocèle des triangles avec une base strictement plus petite que chacun des deux côtés égaux. Nous allons considérer le cas d'une hiérarchie implicative complète $\mathcal{HI}_c(S)$; la généralisation au cas d'une hiérarchie implicative quelconque se comprenant assez bien à partir de la restriction

complète à chacune des classes S_j d'une partition P de S .

Théorème 2.3. *La donnée d'une hiérarchie implicative complète $\mathcal{HL}_c(S)$ définit un ordre total sur S , ce dernier correspond à celui de gauche à droite des feuilles de la hiérarchie.*

Démonstration. Considérons la relation binaire R sur S définie comme suit :

$$(\forall(x, y) \in S \times S), xRy \Leftrightarrow \exists(A, B) \in \mathcal{HL}_c(S), (x, y) \in A \times B \quad (2.20)$$

(A, B) est une fourche orientée (Cf. Définition 2.47). La relation R est une relation d'ordre total et strict :

$$(\forall(x, y) \in S \times S), xRy \Rightarrow \neg yRx \quad (2.21)$$

Selon la Propriété 2.4, le couple (B, A) n'appartient pas à la hiérarchie implicative (ce n'est pas une fourche orientée). Outre, pour tout sous ensemble B' de B , respectivement A' de A , (B', A') ne peut appartenir à la hiérarchie, et avec (A, B) (Cf. Définition 2.49) s'en trouve violé.

$$(\forall(x, y, z) \in S \times S \times S), xRy \text{ et } yRx \Rightarrow xRz \quad (2.22)$$

En effet, il existe deux fourches orientées (A, B) et (C, D) telles que :

$(x, y) \in A \times B$ et $(y, z) \in C \times D$ et comme $B \cap C \neq \emptyset$, avec $y \in B$ et $y \in C$ et selon la Définition 2.49: $A \cup B \subset C$ et $C \cup D \subset B$, on a $(x, z) \in C \times D$ dans le premier cas et $(x, z) \in A \times B$ dans le second cas. □

Définition 2.52. (x, y, z) est un triangle orienté relativement à l'ordre total R si on a : xRy et yRz

Pour une hiérarchie implicative dont l'ordre induit est R , l'ordre des feuille de gauche à droite est x, y et z . Soit \mathcal{T} , l'ensemble de tels triangles, leur énumération est donnée par : $Card(\mathcal{T}) = \frac{(N-2) \times (N-1) \times N}{6}$, où $N = Card(S)$. En effet, il y a une correspondance bijective entre l'ensemble des parties à 3 éléments de S et l'ensemble des triangles orientés. La restriction de la hiérarchie implicative pour l'ensemble $\{x, y, z\}$ a deux cas possible :



Figure 2.15 – Deux formes implicatives

2.8 Conclusion

Nous avons présenté dans ce chapitre quelques notions mathématiques telles que : les treillis, les familles de Moore, les correspondances de Galois, le système implicatif et les hiérarchies et qui seront utiles dans la résolution de problèmes de la fouille des règles d'association, la construction et la représentation condensée des règles d'association extraites à partir d'un contexte formel (map-reduce).

Chapitre 3

Règle d'Association et mesure d'intérêt

“La justice de l’intelligence est la sagesse. Le sage n’est pas celui qui sait beaucoup de choses, mais celui qui voit leur juste mesure.”

Platon
Philosophe

Sommaire

3.1 Introduction	66
3.2 Modélisation et quelques pratiques pionnières	66
3.2.1 Modélisation mathématique	67
3.2.2 Quelques définitions	68
3.3 Extraction des règles d’association	70
3.3.1 Quelques algorithmes d’extraction des règles d’association	70
3.3.2 Algorithme Apriori	71
3.3.3 Autres algorithmes	72
3.3.4 Test du χ^2	76
3.4 Mesure de qualité des règles d’association	78
3.4.1 Critères de qualité d’une mesure des règles d’association .	80
3.4.2 Quelques mesures de qualité	86
3.5 Conclusion	90

3.1 Introduction

L'une des premières applications des règles d'association fut l'étude du panier de la ménagère dans le domaine de la grande distribution. La recherche des règles d'association est une méthode pour extraire de la connaissance dans les données. Elle consiste à mettre en évidence des combinaisons de produits achetés ensemble dans un supermarché. D'un point de vue marketing, ces règles détectent les comportements et les besoins nouveaux du consommateur. Les jeux de données constitués par les enregistrements des tickets de caisses dissimulent à cet effet des informations utiles sur ses attitudes et les nouvelles tendances, qu'elles soient générales, ou particulières. Le processus de fouille de données par la recherche des règles d'association se base sur une classe particulière de motifs appelés itemsets fréquents ou conjonction d'items fréquents (Agrawal et al., 1993b). En s'appuyant sur la particularité de fréquence des itemsets, la technique consiste à mettre en évidence des règles de la forme *prémisse* (antécédent) \implies *conclusion* (conséquence). Cette implication non pas au sens mathématique, mais comme étant la tendance de la conclusion à être « vraie » lorsque la prémisse est « vraie » (Blanchard, 2005). Les règles d'association expriment alors à partir des données contenues dans une base de données relationnelle les tendances implicatives entre les attributs de la prémisse, et ceux qui apparaissent dans la conclusion.

Les règles d'association ont été étudiées et employées dans de nombreux contextes et pour des domaines d'applications variés. On peut citer notamment : l'aide à la planification commerciale, l'aide au diagnostic et en recherche médicale, l'amélioration des processus de télécommunications, l'organisation et l'accès aux sites Internet, l'analyse de données spatiales et de données statistiques (Fauré, 2007).

3.2 Modélisation et quelques pratiques pionnières

Dans un contexte de la fouille de données binaires $\mathbb{K} = (\mathcal{I}, \mathcal{V}, \mathcal{R})$ où \mathcal{I} est un ensemble fini d'individus (ou d'entités ou d'objets), \mathcal{V} un ensemble fini non vide de variables (ou d'attributs, ou items, ou des propriétés) et \mathcal{R} une relation binaire de \mathcal{I} vers \mathcal{V} . La base de données $\mathbb{K} = (\mathcal{I}, \mathcal{V}, \mathcal{R})$ est appelée un contexte formel (Barbut et Monjardet, 1970; Wille, 1982; Ganter et Wille, 1999).

L'appartenance d'un couple $(i, v) \in \mathcal{I} \times \mathcal{V}$ au graphe de la relation \mathcal{R} exprime le

fait que l'individu i possède la propriété ou la variable v . Alors toute variable peut être identifiée à une application de \mathcal{I} dans le doubleton $\{0, 1\}$, où la valeur 1 mesure la présence de la variable chez un individu de \mathcal{I} . Dans toute la suite, n désigne la cardinalité de \mathcal{I} ($n = \text{card}(\mathcal{I}) = |\mathcal{I}|$).

Tout sous-ensemble X de \mathcal{V} s'appelle un *motif* ou *transaction* ou *itemset* de \mathcal{V} . Le *motif négatif associé* à X est notée par \overline{X} .

Pour tout motif X de \mathcal{V} , nous avons (Totohasina, 2008):

1. $\forall v \in \mathcal{V}, 1 - v$, est le complément à 1 de v , i.e $(1 - v)$ identifie l'absence de la variable v chez un individu.
2. $\forall i \in \mathcal{I}, X(i) = 1 \iff \forall v \in X, i\mathcal{R}v$, i.e. $v(i) = 1$; soit $X = \bigwedge_{v \in X} v = 1$, la conjonction des présences d'un nombre fini de variables de \mathcal{V} .
3. $\overline{X} = \bigvee_{1-v \in X} v \in X$, la disjonction des absences des variables de X .
4. $X' = \{i \in \mathcal{I} | \forall v \in X, i\mathcal{R}v\}$, i.e. l'ensemble de tous les individus communs à tous les éléments de X : c'est le dual d'un motif X de \mathcal{V} , ou l'**extension du motif** X .
5. De façon duale, pour une partie $E \in \mathcal{I}$, le motif contenu par E , défini par $E' = \{v \in \mathcal{V} | \forall i \in E, v(i) = 1\}$, est appelé l'**intension de** E : c'est l'ensemble des variables communes aux individus de E .
6. $\overline{X'} = \mathcal{I} - X' = \overline{\overline{X}}$.
7. $\overline{\overline{X}} = X$.
8. $X \subseteq \mathcal{V}$, mais $\overline{X} \not\subseteq \mathcal{V}$.

En terme d'extension, on a $(X \wedge Y)' = X' \cap Y'$ et $(X \vee Y)' = X' \cup Y'$.

3.2.1 Modélisation mathématique

Néanmoins, il existe plusieurs modélisations probabilistes de la fouille de données dans un contexte binaire comme dans (Lerman, 1981; Lerman, Gras, et Rostam, 1981; Lerman, 1984).

Dans le cadre d'une hypothèse d'équiprobabilité des événements élémentaires de \mathcal{I} , considérons un espace probabilisé discret fini uniforme $(\mathcal{I}, \mathcal{P}(\mathcal{I}), P)$ tel que pour tout événement A de $\mathcal{P}(\mathcal{I})$, $P(A) = \frac{\text{card}(A)}{\text{card}(\mathcal{I})}$.

On notera \mathcal{I} l'ensemble des n individus : $\mathcal{I} = \{i_1, i_2, i_3, \dots, i_n\}$, sur lesquels on a

mesuré m variables aléatoires de Bernoulli, et \mathcal{V} celui de ces dernières qui seront également appelées variables binaires : $\mathcal{V} = \{v_1, v_2, v_3, \dots, v_m\}$.

Pour tout $A \in \mathcal{P}(\mathcal{V}) \setminus \{\emptyset; \mathcal{V}\}$, pour tout $a_i \in A$, a_i est une application de \mathcal{I} vers $\{0; 1\}$ et $P(a_i = 1) = \frac{\text{card}(a_i^{-1}(1))}{n}$, avec $n = \text{card}(\mathcal{I})$. Toute partie non vide de \mathcal{V} sera appelée un motif de \mathcal{V} . Chaque individu i_n consiste à un ensemble de variables de \mathcal{V} . A chaque individu, on associe un identificateur unique appelé Transaction IDentifier (TID). Une base de transactions \mathbb{K} est alors un ensemble de couples formés d'un identificateur de transactions TID et de la transaction proprement dite.

Un motif désigne indifféremment un ensemble éventuellement réduit à une variable. Alors pour deux motifs X et Y dans $\mathcal{P}(\mathcal{V})$: $X' = X^{-1}(1)$, $Y' = Y^{-1}(1)$, $n_X = \text{card}(X)$ et $n_Y = \text{card}(Y)$ sont les nombres de transactions qui réalisent respectivement les motifs X et Y , $n_{XY} = \text{card}(X \cap Y)$ est le nombre de celles qui réalisent à la fois X et Y . \bar{X} désignera la négation de logique de X et $\bar{X}' = \mathcal{I} - X'$. Les réels $P(X')$ et $P(Y')$ seront respectivement appelés le *support* du motif X et celui de Y , et notés $\text{supp}(X)$ et $\text{supp}(Y)$.

Prenons un exemple d'un contexte binaire $\mathbb{K} = (\mathcal{I}, \mathcal{V}, \mathcal{R})$ (Cf. Tableau 4.2). Considérons l'ensemble d'individus (*transactions ou entités*) $\mathcal{I} = \{i_1, i_2, i_3, i_4, i_5, i_6\}$ et l'ensemble de variables $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$ (*articles ou attributs*). Pour $X = \{v_2, v_3\}$, on a $X' = \{i_2, i_3, i_5, i_6\}$ et $\bar{X}' = \{i_1, i_4\}$

$\mathcal{I} \setminus \mathcal{V}$	v_1	v_2	v_3	v_4	v_5
i_1	1	0	1	1	0
i_2	0	1	1	0	1
i_3	1	1	1	0	1
i_4	0	1	0	0	1
i_5	1	1	1	0	1
i_6	0	1	1	0	1

Tableau 3.1 – Exemple de tableau d'un contexte binaire.

3.2.2 Quelques définitions

Définition 3.1. Notons $n_X = \text{card}(\mathcal{I}(X))$ (resp. n_Y), le nombre d'individus \mathcal{I} où X (resp. Y) est à Vrai.

Définition 3.2. Notons $n_{XY} = \text{card}(\mathcal{I}(X \cap Y))$ le nombre d'individus \mathcal{I} où X et

TID	Transaction
i_1	$\{v_1, v_3, v_4\}$
i_2	$\{v_2, v_3, v_5\}$
i_3	$\{v_1, v_2, v_3, v_5\}$
i_4	$\{v_2, v_5\}$
i_5	$\{v_1, v_2, v_3, v_5\}$
i_6	$\{v_2, v_3, v_5\}$

Tableau 3.2 – Base de données transactionnelles.

Y sont tous les deux à Vrai. Cette valeur est appelée indice « brut » d'association entre X et Y .

Définition 3.3. Le support ou taux de couverture d'une règle d'association $X \rightarrow Y$ est définie par l'équation :

$$supp(X \rightarrow Y) = \frac{n_{XY}}{n} = P(X' \cap Y') \quad (3.1)$$

avec $n = \text{card}(\mathcal{I})$. Cette mesure représente le pourcentage d'individus (transactions) vérifiant la règle. Elle définit la portée ou la fiabilité de la règle.

Définition 3.4. La confiance d'une règle d'association $X \rightarrow Y$ est défini par :

$$conf(X \rightarrow Y) = \frac{n_{XY}}{n_X} = \frac{P(X' \cap Y')}{P(X')} = P(Y'|X') \quad (3.2)$$

Cette mesure représente le pourcentage d'individus vérifiant la conclusion de la règle parmi celles qui vérifient la prémisse. $P(Y'|X')$ représente la probabilité conditionnelle qu'une transaction contienne la variable Y sachant qu'elle contient la variable X . On considère que cette valeur est une approximation satisfaisante de $P(Y'|X')$. Elle mesure le degré d'inclusion de X dans Y et permet d'évaluer la solidité de la règle. Elle définit, entre autres, la précision de la règle. Néanmoins, nous verrons que cette mesure *confiance* n'est pas à l'abri des défauts (Totohasina et Feno, 2008). D'où la recherche d'une mesure plus pertinente et efficace à partir d'une approche plus formelle.

Définition 3.5. Une règle d'association $X \rightarrow Y$ est dite **forte** ou **valide** au sens de support-confiance si son support et sa confiance sont supérieurs à des seuils minimaux de support (minSupport) et de confiance (minConfiance) définis par l'utilisateur,

c'est-à-dire si

$$\text{supp}(X \rightarrow Y) \geq \text{minsupp} \text{ et } \text{conf}(X \rightarrow Y) \geq \text{minconf}. \quad (3.3)$$

Remarque 2. *On requiert, en général que le conséquent d'une règle d'association ne soit pas vide et, que la prémisse et le conséquent soient disjoints, pour ne pas considérer des règles triviales qui n'apportent à l'utilisateur aucune information.*

3.3 Extraction des règles d'association

Dans la fouille d'une règle d'association d'une grande base de données, le problème est de trouver toutes les *règles d'association valides* de cette base de données en connaissant les seuils *minsup* et *minconf*. Ce problème est subdivisé en deux sous-problèmes (Hipp, Gntzer, et Nakhaeizadeh, 2000):

1. trouver les motifs fréquents, i.e. les motifs dont le *Support* est supérieur ou égal au *minsupp*, seuil fixé par l'utilisateur. L'espace de recherche d'énumération de tous les motifs est 2^k , où k est le nombre de variables (items). On peut représenter tous les motifs sous forme d'un treillis.
2. générer toutes les règles d'association valides dérivant des motifs fréquents vérifiant le critère de confiance.

Différents algorithmes vont être présentés pour la recherche de l'ensemble des motifs fréquents.

3.3.1 Quelques algorithmes d'extraction des règles d'association

Les algorithmes usuels effectuèrent de multiples boucles à partir des données pour extraire les ensembles des motifs fréquents, puis le filtrage sur la *confiance* nous permet de retenir les **règles** satisfaisantes (**valides**).

Les algorithmes **AIS** (Agrawal et al., 1993b) et **SETM** (Houtsma et Swami, 1995) sont les travaux connus pour l'extraction des motifs fréquents. L'algorithme **OCD** (Mannila, Toivonen, et Verkamo, 1995) fonctionne de la même façon que *Apriori*. Ces

deux algorithmes ont été optimisés pour améliorer leurs efficacités, comme **AprioriHybrid** (Agrawal et Srikant, 1994), **DHP** (Park, Chen, et Yu, 1995), **Sampling** (Toivonen, 1996), **Partition** (Savasere, Omiecinski, et Navathe, 1995), **DIC** (Brin, Motwani, Ullman, et Tsur, 1997).

3.3.2 Algorithme Apriori

L'algorithme **Apriori** (Agrawal et Srikant, 1994) est le premier algorithme proposé. C'est un algorithme de référence, et a été subi par de nombreuses améliorations. Il procède en plusieurs passes. Dans la première passe les 1-motifs (ensembles de variables de taille 1) fréquents sont cherchés. Lors des passes suivantes, l'algorithme utilise une procédure **Apriori-Gen** pour générer un ensemble fréquent candidat de taille k à partir de deux ensembles fréquents de taille $k - 1$. L'algorithme s'arrête quand il n'est plus possible de générer de nouveaux candidats de taille supérieure.

Algorithme 2 Algorithme de recherche des motifs fréquents **Apriori**

Entrée : Base de données; *minsupp*: seuil minimum de support

Sortie : L ensemble de couples $(X, supp(X))$ où X est un motif et $supp(X)$ son support

```

1 :  $L_1 = \{1\text{-motifs fréquents}\}$ 
2 :  $k \leftarrow 2$ 
3 : tant que  $L_{k-1} \neq \emptyset$  faire
4 :    $C_k = \text{Apriori\_Gen}(L_{k-1});$  // Elagage de  $C_k$ 
5 :   pour tout instance  $i \in \mathcal{I}$  faire
6 :      $C_i \not\subseteq (C_{k,i})$ 
7 :     pour tout candidat  $c \in C_i$  faire
8 :        $c.\text{count}++;$ 
9 :     fin pour
10 :  fin pour
11 :   $L_k = \{c \in C_k / c.\text{count} \geq \text{minsupp}\};$  // Les candidats sont filtrés
12 :   $k++$ 
13 : fin tant que
14 :  $L = \bigcup_i L_i$ 
15 : Retourner  $L$ 

```

Algorithme **Apriori-Gen** (génération des motifs candidats) : cet algorithme construit à partir des $(k - 1)$ -motifs fréquents L_{k-1} , l'ensemble des k motifs candidats C_k . Il se déroule en deux étapes :

- Il fusionne deux $(k - 1)$ -motifs P et Q qui partagent leur $(k - 2)$ premiers variables.
- Il supprime de C_k tout motif X pour lequel au moins un sous-ensemble de longueur $k - 1$ de X n'appartient pas à L_{k-1} .

Algorithme 3 Algorithme de Génération des motifs candidats **Apriori-Gen**

Entrée : L_k ensemble de motifs X fréquents; $minconf$

Sortie : L ensemble de couples $(X, supp(X))$ où X est un motif et $supp(X)$ son support

```

1 :  $C_k = \{ \}$ 
2 : pour chaque  $X$  de  $L_{k-1}$  faire
3 :   pour chaque  $X$  de  $L_{k-1}$ , avec  $X < Y$ ,  $X$  et  $Y$  partagent leur  $k - 2$  premiers items faire
4 :     pour chaque  $Z \subset X \cup Y$  tel que :  $|Z| = k - 1$  faire
5 :       si  $Z \notin L_{k-1}$  alors
6 :         continuer avec  $Y$  suivant ;
7 :       fin si
8 :     fin pour
9 :     Ajouter  $(X \cup Y)$  à  $C_k$ ;
10 :   fin pour
11 : fin pour
12 : Retourner  $C_k$ 

```

3.3.3 Autres algorithmes

En présence de données denses ou fortement corrélées, l'efficacité de l'algorithme **Apriori** diminue. Il existe de nombreux algorithmes de recherche des règles d'association améliorés pour accélérer la construction des ensembles fréquents. Voici quelques descriptions de certains de ces algorithmes :

- **AprioriTID** (Cf. Annexe A - Algorithme 5) : est une variante de l'algorithme Apriori proposé par Agrawal et Srikant (1994). Il permet de diminuer progressivement la taille de la base de données considérée, dans le but de la stocker en mémoire et de ne plus réaliser d'opération d'entrée/sortie, après le premier parcours de celle-ci. En effet, on met toute la base en mémoire et à chaque niveau du treillis, on représente les transactions par les k -motifs candidats qu'elle contient ; une seule passe suffit, mais il faut que toute la base tienne en mémoire.

- **DIC**: (Dynamic Itemset Counting) a été proposé par [Brin, Motwani, et Silverstein \(1997\)](#) en 1997. Cet algorithme procède par niveaux, mais au niveau k dès qu'un motif a atteint le seuil de fréquence, on introduit les motifs candidats du niveau $k + 1$ qu'il contribue à générer, ce qui diminue le nombre de passes nécessaires sur la base.
- **Apriori Partition** (Cf. Annexe A - Algorithme 9;10;11) : Cet algorithme a été proposé par [Savasere et al. \(1995\)](#) comme extension toujours d'*Apriori* où la base de données est divisée en N partitions qui tiennent chacune en mémoire. Les partitions peuvent ainsi être facilement traitées en parallèle. Chaque partition est traitée indépendamment et on réalise la découverte des ensembles fréquents pour chaque partition.
- **Sampling** (Cf. Annexe A - Algorithme 6;7;8) : ([Toivonen, 1996](#)) Il se base sur l'extraction d'un échantillon de la base qui tient en mémoire, à partir duquel on construit l'ensemble des motifs fréquents dans l'échantillon ainsi que sa bordure négative constituée des motifs non fréquents minimaux dont toutes les parties sont fréquentes, ce qui limite le risque de non exhaustivité.
- **Algorithme bitmap**: ([Gardarin, 1999](#)) se base sur la notion d'index bitmap. Les calculs pour les opérations d'unions et d'intersections sont plus rapides dans les vecteurs binaires que dans les listes. L'index bitmap est une matrice binaire de N lignes et P colonnes, où les lignes sont les transactions de la base de données et les colonnes sont les différentes valeurs possibles de la variable indexée. Si les variables sont continues alors la construction de l'index est plus difficile.

Des algorithmes d'exploration du treillis en profondeur tel que :

- **FP-Growth** (Cf. Annexe A - Algorithme 13;14) : ([J. Han, Pei, et Yin, 2000](#)) utilise une représentation très condensée des données pour évaluer les fréquences par comptage dans la base.
- **Eclat** (Cf. Annexe A - Algorithme 12) : ([Zaki, Parthasarathy, Ogihara, et Li, 1997](#)) recherche en profondeur dans le treillis par intersection rapide des TID-listes, la procédure étant interrompue dès que l'on est sûr que le motif candidat ne peut plus être fréquent.
- **Close** (Cf. Annexe A - Algorithme 18;19) : ([Taouil, Pasquier, Bastide, et](#)

Lakhal, 1999) extrait les motifs fermés fréquents, qui constituent une partie génératrice des motifs fréquents et de leur support, ce qui réduit les temps d'extraction et produit des règles non redondantes.

- **MaxEclat** (Zaki et al., 1997) ou **Max-Miner** (Cf. Annexe A - Algorithme 15;16;17) (Roberto et Bayardo, 1998): sont des algorithmes de recherche de motifs fréquents maximaux où les sur-ensembles de motifs sont non-fréquents. Mais ils se prêtent mal au calcul du support des ensembles de fréquents qu'ils contiennent, calcul pourtant nécessaire pour le calcul de la confiance des règles générées.

Il existe de nombreux algorithmes parallèles et distribués basés sur la recherche de motifs fréquents et notamment sur l'algorithme Apriori. Zaki (1999) propose de classer les algorithmes par stratégie d'équilibrage de charge (load-balancing), architecture et parallélisme. Le parallélisme apporte deux perspectives intéressantes : parallélisme de données et parallélisme de tâches (Chattratchat et al., 1997). Les deux paradigmes diffèrent par le fait que l'ensemble de candidats soit distribué ou non à travers les processeurs. Dans le paradigme de parallélisme de données, chaque nœud compte le même ensemble de candidats alors que dans le paradigme de parallélisme de tâches, l'ensemble des candidats est divisé et distribué à travers les processeurs, et chaque nœud compte un ensemble différent de candidats.

Parallélisme de données :

Les algorithmes qui adoptent le paradigme de parallélisme de données incluent : **Count Distribution** (Agrawal et Shafer, 1996), **PDM** (Parallel Data Mining) (Park et al., 1995), **DMA** (Distributed Mining Algorithm) (Cheung, Ng, Fu, et Fu, 1996), et **CCPD** (Common Candidate Partitioned Database) (Zaki, Parthasarathy, Ogihara, et Li, 1996). Ces algorithmes parallèles diffèrent par l'utilisation ou non de techniques d'élimination de candidats (pruning) ou de technique de comptage de candidats efficaces.

Parallélisme de tâches :

Les algorithmes adoptant le paradigme de parallélisme de tâche incluent : **DD** (Data Distribution) (Agrawal et Shafer, 1996), **IDD** (Intelligent Data Distribution) (J. Han, Tseng, et Keleher, 1997), **HPA** (Hash-based Parallel mining of Association rules) (Shintani et Kitsuregawa, 1996) et **PAR** (Parallel Association Rules) (Zaki et al., 1997), qui inclue les algorithmes (**Par-Eclat**, **Par-MaxEclat**, **Par-Clique** et

Par-MaxClique) (Cf. Annexe A - Algorithme 20;21). Ils divisent tous aussi bien les candidats que la base de données parmi les processeurs. Ils diffèrent dans la façon dont les candidats et la base de données sont divisés.

Une grande partie de l'activité de recherche dans le domaine de l'exploration de données a été centrée autour des règles d'association. Il y a beaucoup d'excellentes publications qui résument cette recherche, comme dans (Chen, Han, et Yu, 1996; Fayyad, Piatetsky-Shapiro, et Smyth, 1996a; Fayyad et Storloz, 1997). La tâche principale d'extraction des règles d'association est de dériver un ensemble de règles d'association fortes sous la forme de $X \rightarrow Y$, où X et Y sont des ensembles disjoints d'éléments (Agrawal, Imielinski, et Swami, 1993a; Brin, Motwani, et Silverstaein, 1997; E. Han, Karypis, et Kumar, 1997; Miller et al., 1997; Park, Chen, et Yu, 1997; Srikant et Agrawal, 1997). Afin de mettre en œuvre cette tâche, un large éventail de problèmes ont été étudiés couvrant des sujets aussi divers que les algorithmes efficaces pour l'extraction des règles d'association (Agrawal et Srikant, 1994; Brin, Motwani, et Silverstaein, 1997; Park et al., 1997), les mesures des motifs (Brin, Motwani, et Silverstaein, 1997; Aggarawal, Charu, et Yu, 1998), l'exploration de données parallèle pour les règles d'association (E. Han et al., 1997), etc.

Le plus souvent, les transactions à partir desquelles sont extraites les règles d'association ne sont qu'un échantillon d'une population plus vaste. Au terme de la procédure d'extraction et d'évaluation des règles, on dispose d'une multitude de règles décrites par différentes mesures, au minimum le *support* et la *confiance*, ainsi qu'une mesure de l'intérêt de la règle. Différentes questions se posent classiquement : ces mesures, en particulier le *support* et la *confiance*, dépassent-elles significativement pour toutes les règles le seuil requis ? Telle mesure dépasse-t-elle significativement une valeur fixée ? La *confiance* $P(Y'|X')$ d'une règle $X \rightarrow Y$ est-elle significativement supérieure à sa fréquence a priori ? Dans ce dernier cas, il s'agit de tester l'hypothèse d'indépendance (H_0) du conséquent Y et de l'antécédent X contre une hypothèse de dépendance positive (H_1), afin de ne retenir que les règles statistiquement significatives. On doit donc pratiquer une multitude de tests, ce qui pose le problème du contrôle du risque multiple. Par exemple, si l'on effectue le test d'indépendance de X et Y pour 10000 règles successivement, en fixant à 0.05 le niveau du risque de 1ère espèce $\alpha = P(\text{décider } H_1/H_0)$, alors même qu'aucune règle ne serait pertinente, on sélectionne quand même 500 règles en moyenne. Dans la suite, nous

rappelons simplement une approche typiques de validation des règles d'association établies dans la littérature actuelle : le test du χ^2 .

3.3.4 Test du χ^2

La méthode de test du chi-carré a été présentée par [Brin, Motwani, et Silverstaein \(1997\)](#). Cette méthode mesure la signification des associations par le test du χ^2 pour la corrélation utilisée dans les statistiques classiques. Cela conduit à une mesure qui est fermée vers le haut dans le treillis de l'ensemble d'éléments, ce qui permet de réduire le problème d'exploration de données à la recherche d'une bordure entre les ensembles d'objets corrélés et non corrélés dans le treillis. Cette approche est utile car elle capture non seulement la corrélation, mais détecte également l'implication négative. Nous allons expliquer cela par rapport à un exemple.

Exemple 3.3.1. *Supposons que nous ayons les données du panier de consommation d'une épicerie, constituée de n paniers. Concentrons-nous sur l'achat de thé et de café. Dans le tableau suivant, les lignes t et \bar{t} correspondent à des paniers qui contiennent ou non du thé, et de même les colonnes c et \bar{c} correspondent à café. Les chiffres représentent le pourcentage de paniers.*

	c	\bar{c}	Σ
t	20	5	25
\bar{t}	70	5	75
Σ	90	10	100

Tableau 3.3 – Achat de thé et de café

Nous pouvons appliquer le modèle de support-confiance à la règle d'association potentielle $t \rightarrow c$. Le *support* pour cette règle est de 20%, ce qui est assez élevé. La *confiance* est la probabilité conditionnelle qu'un client achète du café, étant donné qu'il achète du thé, i.e. $P[t \wedge c] / P[t] = 20/25 = 0,8$, soit 80%, ce qui est également élevée. À ce stade, nous pouvons conclure que la règle $t \rightarrow c$ est une *règle valide*.

Considérons le fait que la probabilité a priori qu'un client achète du café est de 90%. En d'autres termes, un client qui est connu pour acheter du thé est moins susceptible d'acheter du café (de 10%) qu'un client dont nous n'avons aucune information.

Cela signifie que la valeur de corrélation sera prise en paramètre pour déterminer

si une règle est valide ou non. A cet effet, Brin et al. ont proposé une mesure pour découvrir les règles d'association en utilisant le test du χ^2 comme suit.

Soit $P(X')$ la probabilité que l'événement X se produise, et $P(\overline{X'}) = 1 - P(X')$ la probabilité que l'événement X ne se produise pas. De même, $P(X' \cap Y')$ est la probabilité que l'événement X et l'événement Y se produisent ensemble, et $P(\overline{X'} \cap Y)$ est la probabilité que l'événement Y se produise, mais pas l'événement X , et ainsi de suite.

Définition 3.6. *Les événements X et Y sont indépendants si $P(X' \cap Y') = P(X')P(Y')$. De même, si $P(X' \cap Y' \cap Z') = P(X')P(Y')P(Z')$, alors X , Y et Z sont tous les trois indépendants.*

Définition 3.7. *Si deux événements ne sont pas indépendants, ils sont dépendants ou corrélés.*

S'il y a une série de n essais, le nombre d'occurrences de l'événement X est noté $O_n(X)$, ou simplement $O(X)$ lorsque n est compris. $P(X')$ peut être estimé par $O_n(X)/n$. De cette manière, on peut estimer si $P(X' \cap Y') \neq P(X')P(Y')$.

Soit R l'ensemble de toutes les valeurs de panier possibles et r est une valeur de panier unique avec $R = \{i_1, \overline{i_1}\} \times \dots \times \{i_k, \overline{i_k}\}$ et $r = r_1 \dots r_k$. Chaque valeur de r désigne une cellule. Cette terminologie provient de la visualisation de R comme une table de dimension k , connue sous le nom de table de contingence. Soit $O(r)$ le nombre de paniers tombant dans la cellule r . Pour tester si une cellule donnée est dépendante, nous devons déterminer si le nombre réel dans la cellule r diffère suffisamment de l'attente.

Dans le test du χ^2 , les attentes sont calculées en supposant l'indépendance. Ainsi, $E[i_j] = O_n(i_j)$ pour un seul article, $E[\overline{i_j}] = n - O_n(i_j)$, et $E[r] = n \times E[r_1]/n \times \dots \times E[r_k]/n$. En effet, la statistique du χ^2 est définie comme suit :

$$\chi^2 = \sum_{r \in R} \frac{(O(r) - E[r])^2}{E[r]} \quad (3.4)$$

En effet, il s'agit d'un écart normalisé par rapport à l'attente. La statistique du χ^2 , telle que définie, spécifiera si tous les k -éléments sont indépendants de k . Afin de déterminer si certains sous-ensembles d'éléments sont *corrélés*, comme par exemple si on a, X , Y et Z , nous limitons simplement la gamme de r à $\{X', \overline{X'}\} \times \{Y', \overline{Y'}\} \times$

$\{Z', \overline{Z'}\}$. Si toutes les variables étaient réellement indépendantes, la valeur du χ^2 serait 0 (en tenant compte des fluctuations si $n \rightarrow \infty$). Si elle est supérieure à une valeur seuil (3,84 au niveau de signification de 95%), nous rejetons l'hypothèse d'indépendance.

Exemple 3.3.2. *Nous avons le tableau de contingence ci-dessous (Cf. Tableau 3.4) pour les motifs X et Y*

	Y	\overline{Y}	Σ
X	3	3	6
\overline{X}	3	1	4
Σ	6	4	10

Tableau 3.4 – Tableau de contingence de X et Y

Maintenant $E[X] = O(X) = 6$, alors que $E[Y] = O(Y) = 6$. Notons que $E[X]$ est la somme de la ligne 1, alors que $E[Y]$ est la somme de la colonne 1. La valeur du χ^2 est :

$$\begin{aligned}
 & (3 - 6 \times 6/10)^2 / (6 \times 6/10) + (3 - 6 \times (10 - 6)/10)^2 / (6 \times (10 - 6)/10) \\
 & + (3 - (10 - 6) \times 6/10)^2 / ((10 - 6) \times 6/10) \\
 & + (1 - (10 - 6) \times (10 - 6)/10)^2 / ((10 - 6) \times (10 - 6)/10) \\
 & = 0.1 + 0.15 + 0.15 + 0.225 = 0.625
 \end{aligned}$$

Puisque 0,625 est inférieur à 3,84, nous ne rejetons pas l'hypothèse d'indépendance à l'intervalle de confiance de 95%.

3.4 Mesure de qualité des règles d'association

Les mesures de qualité des règles d'association, ou mesures d'intérêt ou mesures probabilistes de qualité, sont des indicateurs numériques destinés à guider l'utilisateur vers les connaissances potentiellement intéressantes dans les grands volumes de règles produits par les algorithmes de fouille de données. Ces mesures offrent d'évaluer la qualité des règles selon différents points de vue, et permettent d'ordonner les règles des meilleures aux plus mauvaises. Elles peuvent également jouer le rôle de filtre, en rejetant les règles en dessous d'un seuil de qualité minimale.

Définition 3.8. Soient les motifs $X \in \mathcal{P}(\mathcal{V})$ et $Y \in \mathcal{P}(\mathcal{V})$. Une mesure de qualité probabiliste est une fonction réelle μ de $\mathcal{P}(\mathcal{V}) \times \mathcal{P}(\mathcal{V})$ telle que pour toute règle d'association $X \rightarrow Y$, $\mu(X \rightarrow Y)$ est calculée à partir des quatre quantités définies par : $n = \text{card}(\mathcal{I})$, $P(X')$, $P(Y')$ et $P(X' \cap Y')$, où P désigne la probabilité discrète uniforme sur l'espace probabilisable discret $(\mathcal{I}, \mathcal{P}(\mathcal{I}))$.

Définition 3.9. Une mesure de qualité des règles μ sera dite symétrique si pour toute règle d'association $X \rightarrow Y$, on a $\mu(X \rightarrow Y) = \mu(Y \rightarrow X)$.

μ sera dite parfaitement symétrique, si pour toute règle d'association $X \rightarrow Y$, on a $\mu(X \rightarrow Y) = \mu(\bar{X} \rightarrow \bar{Y})$ (Azé et al., 2003).

Définition 3.10. Une mesure de qualité des règles μ sera dite orientée, s'il existe au moins une règle d'association $X \rightarrow Y$ telle que l'on a $\mu(X \rightarrow Y) \neq \mu(Y \rightarrow X)$

Définition 3.11. Une mesure de qualité des règles μ sera dite implicative, s'il existe pour toute règle d'association $X \rightarrow Y$, on a $\mu(Y \rightarrow X) = \mu(X \rightarrow Y)$ (Totohasina, 2003)

Terminologies : Un motif X est appelé la *prémisse* de la règle $X \rightarrow Y$, et Y la *conclusion*. La limitation à ces quatre paramètres, pour définir une mesure de qualité de règles, se justifie par le fait qu'ils suffisent pour retrouver les effectifs correspondants aux cinq cases restantes dans le tableau de contingence obtenu par le croisement de X et Y (Cf. Tableaux 3.5,3.6), car $X = (X \cap Y) + (X \cap \bar{Y})$ et $Y = (Y \cap X) + (Y \cap \bar{X})$.

	Y	\bar{Y}	
X	(1) $X \cap Y$	(2) $X \cap \bar{Y}$	(7)
\bar{X}	(3) $\bar{X} \cap Y$	(4) $\bar{X} \cap \bar{Y}$	(8)
	(5)	(6)	(9) \mathcal{V}

⇒

	Y'	\bar{Y}'	
X'	(1) $X' \cap Y'$	(2) $X' \cap \bar{Y}'$	(7)
\bar{X}'	(3) $\bar{X}' \cap Y'$	(4) $\bar{X}' \cap \bar{Y}'$	(8)
	(5)	(6)	(9) \mathcal{I}

Tableau 3.5 – Tableaux de contingence associés.

	Y'	\bar{Y}'	
X'	(1) $P(X' \cap Y')$	(2)	(7) $P(X')$
\bar{X}'	(3)	(4)	(8)
	(5) $P(Y')$	(6)	(9) $\frac{Card(\mathcal{I})}{n} = 1$

Tableau 3.6 – Tableau de probabilités conjointes associée.

Compléter les cinq cases restantes du Tableau 3.6 connaissant au moins les quatre paramètres $n = Card(\mathcal{I})$, $P(X')$, $P(Y')$ et $P(X' \cap Y')$ selon : (5) = (1) + (3); (6) = (2) + (4); (7) = (1) + (2); (8) = (3) + (4); (9) = (5) + (6) = (7) + (8).

Une règle d'association $X \rightarrow Y$ est dite **exacte** si $X' \subset Y'$ et s'il existe une des mesures de qualité μ parmi les mesures considérées telle que $\mu(X \rightarrow Y) = 1$, sinon elle est dite règle d'association **approximative** qui est aussi connue sous le nom d'**implication partielle** ou **implication statistique**. Une règle d'association de type $X \rightarrow \bar{Y}$ sera dite une **règle négative à droite**, et $\bar{X} \rightarrow Y$ une **règle négative à gauche**.

3.4.1 Critères de qualité d'une mesure des règles d'association

Tout d'abord, nous allons voir quelques critères de qualité pour les mesures de règles d'association. Mais, il est impossible pour une mesure de satisfaire tous les critères de qualité de l'utilisateur. Pour apprécier une mesure μ , nous allons établir une liste non exhaustive tirée dans quelques littératures (Piatetsky-Shapiro, 1991; Lallich et Teytaud, 2004; Tan, Kumar, et Srivastava, 2002; Lenca, Meyer, Picouet, Vaillant, et Lallich, 2003; Blanchard, Kuntz, Guillet, et Gras, 2004; Guillaume, 2010).

1. Compréhensibilité de la mesure pour l'utilisateur
2. Nature des règles ciblées par la mesure
3. Sensibilité à l'apparition des exemples et des contre-exemples
4. Sens de variation de la mesure
5. Nature de la variation linéaire/non linéaire

6. Impact de la rareté du conséquent
7. Sensibilité à la taille des données
8. Caractère discriminant de la mesure
9. Utilisation d'un seuil d'élagage
10. Classement induit par une mesure
11. Comportement par rapport au contexte des règles étudiées
12. Déviation à l'équilibre
13. Contradiction des connaissances a priori de l'utilisateur
14. Sensibilité au bruit

3.4.1.1 Compréhensibilité ou intelligibilité

Pour [Lallich et Teytaud \(2004\)](#), la mesure doit être intelligible pour pouvoir communiquer et expliquer les résultats obtenus. Dans [Lenca et al. \(2003\)](#) ont été bien définis les 3 niveaux d'intelligibilité : *la mesure μ est difficile à interpréter, μ se ramène à des quantités usuelles et μ peut s'expliquer par une phrase*. Comme, le support d'une règle $X \rightarrow Y$ représente « la fréquence d'apparition de cette règle », tandis que la confiance de la règle $X \rightarrow Y$ représente la « force de cette règle ». La plupart des mesures de qualité ne possèdent pas cette propriété, ce qui rend difficile l'interprétation de leur valeur pour l'utilisateur.

3.4.1.2 Nature des règles ciblées par la mesure

1. La mesure μ doit pouvoir faire la distinction entre $X \rightarrow Y$ et $X \rightarrow \bar{Y}$. Ce qui veut dire que les exemples d'une règle $X \rightarrow Y$ forment les contre-exemples de l'autre $X \rightarrow \bar{Y}$ ([Freitas, 1999](#); [Lallich et Teytaud, 2004](#)). D'une autre manière, [Tan, Kumar, et Srivastava \(2004\)](#) ont montré que la mesure μ doit vérifier les relations suivantes : $\mu(X \rightarrow Y) = -\mu(X \rightarrow \bar{Y})$ et $\mu(X \rightarrow \bar{Y}) = -\mu(X \rightarrow Y)$.
2. Il est aussi important de chercher le lien orienté ou non entre X et Y : si l'orientation entre X et Y n'est pas significative, il convient d'utiliser des mesures μ évaluant de la même façon $X \rightarrow Y$ et $\bar{Y} \rightarrow \bar{X}$ ([Kodratoff, 1999](#)).
3. Par contre, si l'orientation est importante, alors l'utilisation de telles mesures est à prohiber au profit de mesures évaluant différemment les règles $X \rightarrow Y$ et $Y \rightarrow X$ ([Tan et al., 2002](#); [Lallich et Teytaud, 2004](#)).

3.4.1.3 Sensibilité à l'apparition des exemples et des contre-exemples

L'évaluation de l'intérêt d'une règle peut se mesurer en fonction du nombre (élevé) d'exemples de la règle ou en fonction du nombre (faible) de ses contre-exemples (Freitas, 1999). Dans le premier cas, le comportement de la mesure doit rendre compte de l'apparition des exemples par une augmentation de sa valeur. Et dans le second cas, l'apparition des contre-exemples doit se traduire par une diminution de la valeur de la mesure. Rappelons que l'observation n_{XY} correspond au nombre d'exemples de la règle $X \rightarrow Y$ et que l'observation $n_{X\bar{Y}}$ correspond au nombre de ses contre-exemples.

3.4.1.4 Sens de variation de la mesure

L'utilisation de mesures de qualité prenant des valeurs positives pour les règles intéressantes permet de se rapprocher des a priori de l'utilisateur sur la notion de qualité. De plus, pour deux règles R1 et R2, si la qualité de la règle R1 est supérieure à celle de R2 alors, la mesure de qualité μ doit assurer que $\mu(R1) > \mu(R2)$. Une bonne mesure μ de la qualité de la règle $X \rightarrow Y$ doit être (Piatetsky-Shapiro, 1991):

- $\mu(X \rightarrow Y) < 0$ en cas de répulsion : $P(Y'|X') < P(Y')$
- $\mu(X \rightarrow Y) = 0$ en cas d'indépendance de X et Y : $P(Y'|X') = P(Y')$
- $\mu(X \rightarrow Y) > 0$ en cas d'attraction de X et Y : $P(Y'|X') > P(Y')$

La définition de ces deux zones permet de capturer un certain aspect de la qualité des règles d'association : le comportement par rapport à l'indépendance.

3.4.1.5 Nature de la variation linéaire/non linéaire

La mesure peut varier linéairement en fonction de $n_{X\bar{Y}}$ ou bien avoir un comportement permettant de rendre compte de l'apparition progressive des contre-exemples. La mesure ayant alors une tendance à décroître lentement lorsque peu de contre-exemples apparaissent puis de plus en plus rapidement jusqu'à atteindre une valeur minimale (idéalement nulle) i.e. $P(X' \cap Y')$ tend vers 0. Une mesure μ variant linéairement en fonction du nombre d'exemples (resp. contre-exemples) de la règle $X \rightarrow Y$ est a priori plus sensible au bruit qu'une mesure ayant une variation non linéaire. En effet, si les données sont bruitées, pour une règle $X \rightarrow Y$, quelques exemples vont se transformer en contre-exemples et inversement. Les mesures de

qualité ayant un comportement non linéaire en fonction des exemples permettent donc d'accepter quelques contre-exemples sans pour autant diminuer la valeur de la mesure.

3.4.1.6 Impact de la rareté du conséquent

Une mesure μ doit être une fonction croissante de $1 - P(Y')$, i.e., la rareté de conséquent. En effet, plus le conséquent Y est rare, plus le fait qu'il contienne la prémisse X pour une modélisation donnée est intéressant. L'influence des observations n_X et n_Y dans l'évaluation de la mesure de qualité μ permet de contrôler le comportement de la mesure par rapport aux tailles des prémisses et des conclusions des règles étudiées. Soient deux règles R1 et R2, si $conf(R1) = conf(R2)$ et $supp(R1) = supp(R2)$, alors la seule observation qui varie est la taille du conséquent (n_Y).

3.4.1.7 Sensibilité à la taille des données

Une mesure est dite descriptive, si elle ne change pas en cas de dilatation des données, dans le cas contraire, elle est dite mesure statistique. Donc, pour une mesure statistique μ , la taille de données n doit intervenir dans son évaluation (Lallich et Teytaud, 2004). Pour une mesure statistique, en fixant les quantités marginales $P(X')$ et $P(Y')$, il est intéressant de savoir comment évaluer la règle $X \rightarrow Y$ si on augmente la taille de données n . Si une mesure varie de façon croissante avec n et admet une valeur maximale, alors elle risque de perdre son pouvoir discriminant quand n devient suffisamment grand.

3.4.1.8 Utilisation d'un seuil d'élagage

Les mesures de qualité retenues pour extraire les règles d'association doivent pouvoir être utilisées avec un seuil d'élagage de manière à éliminer toutes les règles qui n'intéressent pas l'utilisateur (Lenca, Meyer, Vaillant, Picouet, et Lallich, 2004). Les mesures ayant un sens concret pour l'utilisateur, ainsi que les mesures normalisées et ayant un caractère statistique se prêtent bien à la détermination d'un seuil d'élagage. Ce seuil peut être fixé par l'utilisateur soit avant la phase d'extraction des règles d'association, soit lors d'une phase de post-élagage des règles. Lorsque le seuil est déterminé a priori par l'utilisateur, ce seuil ne prend pas en considération la nature

des données et peut donc conduire à des résultats ne représentant pas toujours les données. L'utilisation de seuils d'élagage calculés directement à partir des données peut permettre d'éviter ce problème. De plus, ceci évite de solliciter l'utilisateur en lui demandant de déterminer a priori des seuils associés à des mesures dont la compréhension peut lui échapper. De tels seuils peuvent être obtenus à partir des valeurs moyennes observées sur les données. Une méthode classique en fouille de données (Lerman, 1984; Daudé, 1992), consiste à centrer et réduire les valeurs observées par rapport à la moyenne et à l'écart-type de la mesure.

3.4.1.9 Classement induit par une mesure

Considérons deux mesures de qualité μ et μ' . Si les deux mesures sont comparables au sens des critères précédemment énoncés, comment aider l'utilisateur à choisir une des deux mesures ? Un des critères de comparaison pouvant être utilisé est l'ordre induit par ces deux mesures sur les règles obtenues. Soit \mathcal{R} l'ensemble des règles obtenues en utilisant un algorithme indépendant de μ et de μ' . Les deux mesures sont équivalentes du point de vue de l'ordre induit sur les règles, si et seulement si : $\forall R1, R2 \in \mathcal{R} \mu(R1) > \mu(R2) \Leftrightarrow \mu'(R1) > \mu'(R2)$.

C'est-à-dire que pour tout couple de règles $(R1, R2)$ trouvées par l'algorithme, les mesures μ et μ' classent toujours $R1$ et $R2$ de la même manière l'une par rapport à l'autre.

3.4.1.10 Comportement par rapport au contexte des règles étudiées

Nous appelons contexte un ensemble de règles d'association obtenu par un algorithme tel qu'*Apriori*. Les règles de ce contexte vérifient un ensemble de contraintes imposées sur les mesures de qualité utilisées par l'algorithme d'extraction (*support* et *confiance minimaux* pour *Apriori* par exemple). Les règles de ce contexte sont considérées comme des règles valides. Soit une règle d'association $R : X \rightarrow Y$ située dans un contexte \mathcal{K}_1 ou \mathcal{K}_2 . Une mesure μ est dite sensible au contexte si elle évalue différemment la règle R en fonction du contexte duquel R est extraite. Notons $\mu_{\mathcal{K}}(R)$ une règle d'association, issue d'un ensemble de règles valides \mathcal{K} , et dont la qualité est évaluée avec la mesure μ . Nous considérons qu'une règle R se situe dans un contexte \mathcal{K} si $R \in \mathcal{K}$. Si μ est sensible au contexte et si $\mathcal{K}_1 \neq \mathcal{K}_2$ alors $\mu_{\mathcal{K}_1} \neq \mu_{\mathcal{K}_2}$. Si R est une règle intéressante pour l'utilisateur et si \mathcal{K}_1 contient moins de règles

intéressantes que \mathcal{K}_2 alors $\mu_{\mathcal{K}_1} \geq \mu_{\mathcal{K}_2}$. En effet, si la règle R se situe dans un contexte \mathcal{K}_1 contenant peu de règles intéressantes alors le contraste entre les règles intéressantes et les autres est élevé. Inversement, si la règle R se situe dans un contexte \mathcal{K}_2 contenant beaucoup de règles intéressantes, alors la mesure μ lui associera une valeur plus faible que celle associée à R dans \mathcal{K}_1 .

3.4.1.11 Déviation à l'équilibre

Une mesure de qualité doit tenir compte de l'équilibre, i.e., lorsque les nombres d'exemples et de contre-exemples de la règle sont égaux, une mesure de qualité doit avoir une valeur constante, ou tout au moins asymptotiquement constante en fonction de la taille de l'échantillon (Blanchard, Guillet, Briand, et Gras, 2005).

3.4.1.12 Contradiction des connaissances a priori de l'utilisateur

Ce critère permet de comparer les règles d'association obtenues à un ensemble de connaissances fournies par l'utilisateur et considérées comme valides pour le domaine étudié. Si on a $conf(X \rightarrow Y) \gg conf(Y \rightarrow X)$ et si l'utilisateur a présenté la règle $Y \rightarrow X$ comme étant une *connaissance a priori* du domaine, alors la règle $X \rightarrow Y$ est considérée comme intéressante car elle indique bien qu'il existe une relation entre X et Y mais que le sens de la relation est inversé par rapport *a priori* du domaine. La règle $X \rightarrow Y$ *contredit des connaissances a priori* de l'utilisateur (Suzuki, 1997; Suzuki et Kodratoff, 1998).

3.4.1.13 Sensibilité au bruit

Nous savons bien, qu'il est rare de trouver des données réelles parfaites et l'étude de ces données ne doit pas produire des résultats trop éloignés de ceux obtenus sur les mêmes données non bruitées. Le bruit peut prendre différentes formes : valeur absente dans les données ou remplacée par une valeur par défaut ; erreur de réglage d'un appareil de mesure utilisé pour obtenir les données ; fautes de frappe lors de la création de la base, etc. Cela nous suffit de dire l'importance du bruit et de son étude dans un système d'extraction de connaissances. La prise en considération du bruit dans une mesure de qualité est difficile car le bruit est par nature mal défini. Idéalement, un système d'extraction de règles d'association ne doit pas fournir à

l'expert des règles erronées même lorsque les données sont bruitées.

3.4.2 Quelques mesures de qualité

Il existe une soixantaine de mesures répertoriés dans les littératures, plusieurs études ont été déjà faites sur les classements de ces mesures comme dans (Nguifo, Grissa, et Guillume, 2012; Rakotomalala, Totohasina, et Diatta, 2018), mais dans le cadre de cette thèse, nous présentons seulement quelques mesures et ses sémantiques.

Support (Agrawal et al., 1993b): Le *Support* d'une règle $X \rightarrow Y$ indique la proportion d'objets (entités, individus) vérifiant à la fois la prémisse et le conséquent de la règle.

$$\text{supp}(X \rightarrow Y) = P(X' \cap Y') \quad (3.5)$$

Elle est symétrique et est souvent utilisée pour élaguer les règles d'association non intéressantes. Il suffit de fixer un seuil d'élagage et supprimer les règles d'association de valeurs inférieures à ce seuil. Mais cela ne suffit à évaluer l'intérêt des règles extraites.

Confiance (Agrawal et al., 1993b): La *Confiance* indique la proportion d'objets (entités, individus) vérifiant le conséquent parmi ceux vérifiant la prémisse.

$$\text{conf}(X \rightarrow Y) = P(Y'|X') = \frac{P(X' \cap Y')}{P(X')} \quad (3.6)$$

C'est une mesure non symétrique et insensible à la taille des données. En effet, quelque soit la taille de \mathcal{I} , l'ensemble d'individus (objets) étudiés, la confiance d'une règle $X \rightarrow Y$ reste toujours constante. Ainsi, elle évalue de manière identique dans des situations différentes, c'est pour cela qu'on a l'occasion à se référer à la confiance centrée.

Confiance centrée (Lallich et Teytaud, 2004): La *confiance centrée* ($\text{conf}_{centree}$) permet de prendre en considération la taille de conclusion de la règle $X \rightarrow Y$.

$$\text{conf}_{centree}(X \rightarrow Y) = \frac{P(X' \cap Y')}{P(X')} - P(Y') \quad (3.7)$$

Elle permet alors de relativiser la confiance d'une règle par rapport à la taille de sa conclusion. Elle est sensible à la taille des données. Mais la connaissance de sa valeur ne suffit pas de connaître la nature de la règle.

Confiance causale (Kodratoff et al., 2001): La mesure *Confiance causale* ($conf_{caus}$) est définie par :

$$conf_{caus}(X \rightarrow Y) = 1 - \frac{1}{2} \left(\frac{1}{P(X')} + \frac{1}{P(\overline{Y'})} \right) P(X' \cap \overline{Y'}) \quad (3.8)$$

C'est l'addition de la confiance due aux instances directes de la règle $X \rightarrow Y$ et la confiance apportée par sa contraposée $\overline{Y} \rightarrow \overline{X}$. Le coefficient 1/2 introduit dans sa définition provient du fait que chaque probabilité (a posteriori) peut être égale à 1.

Intérêt (ou lift) (Brin, Motwani, et Silverstaein, 1997): L'*intérêt* (int) mesure la déviation du *support* de la règle par rapport au cas d'indépendance de la forme :

$$int(X \rightarrow Y) = \frac{P(X' \cap Y')}{P(X') \times P(Y')} \quad (3.9)$$

L'intérêt est une mesure symétrique varie dans l'intervalle $[0, +\infty[$. Si X et Y sont indépendants alors $int(X \rightarrow Y) = 1$. Plus X et Y sont incompatibles, plus $P(X' \cap Y')$ tend vers 0 et donc l'intérêt est proche de 0. Plus X et Y sont dépendants, plus l'intérêt est supérieur à 1.

Conviction (Brin, Motwani, Ullman, et Tsur, 1997): L'intérêt est utilisé pour mesurer la dépendance entre X et Y , qui est symétrique et qui mesure essentiellement la cooccurrence de X et Y . Pour mesurer l'implication, Brin, Motwani, Ullman, et Tsur (1997) introduit la conviction définie par :

$$Conviction(X \rightarrow Y) = \frac{P(X') \times P(\overline{Y'})}{P(X' \cap \overline{Y'})} \quad (3.10)$$

La mesure *Conviction* n'est pas symétrique et varie dans l'intervalle $[0, +\infty[$, elle vaut 1 lorsque X et Y sont indépendantes.

Rappel (Lavrac, Flach, et Zupan, 1999): Le *rappel* (*rappel*) est définie par :

$$\text{rappel}(X \rightarrow Y) = P(X'|Y') = \frac{P(Y' \cap X')}{P(Y')} \quad (3.11)$$

Cette mesure permet d'évaluer la proportion d'objets vérifiant la prémisse de la règle parmi ceux vérifiant le conséquent. Cette mesure est indépendante de la taille pré-misse qui est le même problème que confiance. On peut aussi alors utiliser le rappel centré.

Pearl (Pearl, 1988): La mesure de *Pearl* (*pearl*) est définie par :

$$\text{pearl}(X \rightarrow Y) = P(X') \times |P(Y'|X') - P(Y')| \quad (3.12)$$

Cette mesure permet d'évaluer l'intérêt de la règle $X \rightarrow Y$ par rapport à l'hypothèse d'indépendance entre la prémisse et le conséquent de la règle. L'utilisation de cette mesure permet de vérifier que la règle n'est pas purement le fruit de hasard et qu'elle apporte vraiment une connaissance nouvelle sur les données. En effet, si la mesure de Pearl est proche de 0, cela nous indique que considérer les variables X et Y liées par une relation est aussi intéressant que les considérer indépendantes.

Correlation (Lerman, 1981): La mesure de *Correlation* (*correlation*) est définie par :

$$\text{correlation}(X \rightarrow Y) = \frac{P(X' \cap Y') - P(X')P(Y')}{\sqrt{P(X')P(\bar{X}')P(Y')P(\bar{Y}')}} \quad (3.13)$$

La corrélation d'une règle $X \rightarrow Y$ est évaluée par l'écart à l'indépendance de l'indice associatif brut normalisé par le produit des marges intervenant dans l'étude de l'association liant X et Y . Cette mesure est symétrique, elle est sensible à la taille des données et lorsque les marges n_X , n_Y et n_{XY} sont fixés alors la mesure tend vers $\alpha \times n \times n_{XY}$ lorsque la valeur de n augmente, avec ($\alpha \ll n$). Cette mesure tend donc à trouver des relations entre toutes les variables étudiées dès que les données sont suffisamment volumineuses. Cette mesure est appelée aussi *ϕ - coefficient*.

Piatetsky-Shapiro (Piatetsky-Shapiro et Frawley, 1991): La mesure Piatetsky-Shapiro (*Piatetsky*) est définie par :

$$Piatetsky(X \rightarrow Y) = nP(X')(P(Y'|X') - P(Y')) \quad (3.14)$$

La mesure Piatetsky-Shapiro évalue l'intérêt d'une règle par rapport à son écart à l'indépendance. Elle est une mesure symétrique, sensible à la taille de données, car l'indice de Piatetsky-Shapiro peut aussi s'écrire sous la forme : $n_{XY} - \frac{n_X n_Y}{n}$, mais celle-ci est moins sensible à la taille par rapport à l'indice de Pearl. La mesure de Piatetsky-Shapiro n'est pas une mesure implicative. Cette mesure peut prendre la forme : $Piatetsky(X \rightarrow Y) = n \times Pearl(X \rightarrow Y)$.

Nouveauté (Lavrac et al., 1999): La mesure Nouveauté (*nouveauté*) est définie par :

$$nouveaute(X \rightarrow Y) = P(X' \cap Y') - P(X')P(Y'). \quad (3.15)$$

La mesure *Nouveauté* est une mesure de l'écart à l'indépendance entre la prémisse et le conséquent de la règle. Elle est symétrique, non implicative. Elle dépend de la taille des données.

Loevinger (Loevinger, 1947): La mesure Loevinger (*Loevinger*) est définie par :

$$Loevinger(X \rightarrow Y) = \frac{P(Y'|X') - P(Y')}{P(\bar{Y}')}. \quad (3.16)$$

La mesure de Loevinger est une mesure non symétrique, implicative et sensible à la taille de données. Cette mesure normalise la mesure Confiance centrée par le nombre d'entités ne vérifiant pas le conséquent de la règle.

Satisfaction (Lavrac et al., 1999): La mesure Satisfaction (*Satisfaction*) est définie par :

$$Satisfaction(X \rightarrow Y) = \frac{P(\bar{Y}') - P(\bar{Y}'|X')}{P(\bar{Y}')}. \quad (3.17)$$

Cette mesure varie dans l'intervalle $] -\infty, 1]$ et vaut 0 en cas d'indépendance de X et de Y . La mesure *Satisfaction* n'est pas utile pour classer les règles exactes car

sa valeur est 1 puisque les règles exactes ont une confiance 1. La variation de cette mesure dépend de $P(Y')$. Plus $P(Y')$ est faible, plus la *Satisfaction* est élevée.

3.5 Conclusion

Ce chapitre nous a permis de voir quelques notions sur les règles d'association, l'algorithme de base *Apriori* et quelques algorithmes améliorés d'extraction des règles d'association, des algorithmes d'exploration par treillis en profondeur, des algorithmes parallèles et distribués (parallélisme de données et parallélisme de tâches). Nous avons aussi eu un aperçu sur les critères d'une mesure de qualité des règles d'association ainsi que quelques exemples de mesures de qualité.

Deuxième partie

Contributions : Extraction des
règles d'association M_{GK} -valides
et Classification Hiérarchique
Implicative et Cohésitive
selon M_{GK} (CHIC- M_{GK})

Chapitre 4

Extraction des règles d'association M_{GK} -valides

*“ La connaissance s’acquiert par
l’expérience, tout le reste n’est
que de l’information. ”*

Albert Einstein

Sommaire

4.1 Introduction	94
4.2 l’Analyse Statistique Implicative	95
4.2.1 Quasi-règle implicative selon l’approche classique de R. Gras	95
4.2.2 Intensité d’implication	98
4.3 Mesure de qualité des règles d’association M_{GK}	100
4.3.1 Construction de la mesure M_{GK}	101
4.3.2 Propriétés de la mesure de qualité des règles d’association M_{GK}	102
4.3.3 Commentaire et plaidoyer sur M_{GK}	109
4.4 Extraction des règles d’association M_{GK}-valides avec contri- bution du <i>Support</i>	110
4.4.1 Implication entre variables binaires	111
4.4.2 Prise en compte des valeurs critiques de M_{GK}	113
4.4.3 Apport par considération du <i>Support</i> de la règle d’asso- ciation valide en fonction de M_{GK}	122
4.4.4 Normalisation de la mesure de qualité $supp_{M_{GK}}$	124
4.4.5 Vers un nouvel indice de cohésion implicative selon M_{GK}	133
4.5 Conclusion	142

4.1 Introduction

L'extraction de règles d'association traditionnelle s'est principalement concentrée sur l'identification des relations fortement associées parmi les ensembles de variables qui ont une corrélation fréquente et élevée. Les règles d'association nous permettent de détecter les éléments qui se produisent généralement ensemble dans une application. Les règles d'association par rapport à l'approche Support-Confiance sont des règles positives. Ils indiquent que la présence de certains motifs impliquera la présence d'autres motifs dans les mêmes transactions. Dans les applications, une règle d'association $X \rightarrow Y$ est utilisée pour prédire que « si X se produit, alors Y se produit aussi » i.e. $X' \subset Y'$ (implication exacte). Dans la vie quotidienne, il y a une exception : « si X se produit, alors Y se produit généralement » i.e. $X' \not\subset Y'$ (quasi-implication), et c'est sur cette dernière sémantique est l'origine et la base de l'Analyse Statistique Implicative (ASI).

Outre, il est évidemment impensable de traiter des données réelles avec plusieurs dizaines, voire une centaine des variables sans l'aide des outils informatiques. De nombreux logiciels sont actuellement disponibles pour extraire des connaissances à partir des données (ECD), notamment, l'extraction des règles d'association. On peut citer, entre autres, le package *arules* du logiciel R, dans lequel on peut se servir de plusieurs mesures comme le support, la confiance, le lift, etc. Il y a aussi le logiciel CHIC et R-CHIC, développé par l'équipe de Régis GRAS pour l'extraction des règles d'association basées sur l'intensité d'implication. Étant donné les limites du couple support et confiance, et le fait que CHIC et R-CHIC n'étaient pas conçus pour mesurer des règles négatives, et surtout, les propriétés mathématiques et sémantiques (compatibilité à la logique classique), il est alors légitime de penser à la conception d'un outil d'extraction des connaissances à partir des données basé sur la mesure M_{GK} . Par rapport à cet objectif, nous allons consacrer ce chapitre à détailler les propriétés de la mesure de qualité des règles d'association M_{GK} , ainsi que la description des nouvelles algorithmes permettant de faire une extraction des règles d'association sur ladite mesure M_{GK} tout en comparant les règles extraites avec l'outil CHIC de R. Gras basé sur la théorie de l'ASI.

4.2 l'Analyse Statistique Implicative

La taxonomie organise l'information de façon hiérarchique, de la simple restitution de faits jusqu'à la manipulation complexe des concepts, qui est souvent mise en œuvre par les facultés cognitives dites supérieures. L'usage de taxonomies se retrouve entre autres dans la pédagogie par objectifs, ou dans les propositions du mouvement de l'éducation nouvelle. L'une des taxonomies célèbres est la taxonomie de Bloom : c'est un modèle pédagogique proposant une classification des niveaux d'acquisition des connaissances. En 1978, Régis Gras a construit une taxonomie d'objectifs cognitifs en mathématiques afin de disposer un outil pour apprécier les acquisitions d'élèves de groupes expérimentaux et témoins. Dans sa thèse, [Gras \(1979\)](#), et inspiré par l'indice de similarité de I-C Lerman, un indice à fondement probabiliste fut construit, mais cette fois-ci rendant compte de la dissymétrie des faits observés. Cet indice est appelé *Indice d'implication*. Afin de répondre à l'étude de « si X alors Y », [Gras \(1979\)](#), [Lerman et al. \(1981\)](#) ont défini la notion de *quasi-implication* « si on a X , on a généralement Y », mesurée par une *intensité d'implication* et la notion de graphe d'implication.

4.2.1 Quasi-règle implicative selon l'approche classique de R. Gras

Une population \mathcal{I} d'individus est croisée avec des variables que l'on explore de la façon suivante : « dans quelle mesure peut-on considérer qu'instancier la variable X implique instancier la variable Y » ?. Autrement dit : « les individus ont-ils tendance à être Y si l'on sait qu'ils sont X » ?.

Dans la situation usuelle, où les individus présentant les comportements désignés par le motif X possèdent aussi les comportements désignés par le motif Y . Par rapport à l'ensemble des populations étudiées, on peut dire que pour un individu choisi au hasard, « si cet individu (élève) possède X , alors il possède aussi Y ». On a en effet une règle exacte de type « si X , alors Y ».

Notons X' (resp. Y'), l'ensemble des individus contenant les comportements désignés par le motif X (resp. Y). Dans la pratique, il est parfois de rencontrer des situations où X' n'est pas totalement inclus dans Y' , c'est-à-dire qu'il existe une fraction d'in-

dividus qui n'admettent pas les comportements désignés par le motif Y' . L'approche classique de R. Gras et son équipe consiste alors à analyser le nombre des individus (d'élèves) qui échappent à la règle : *la petitesse de contre-exemples*. En effet, si X' n'est pas inclus totalement dans Y' , il existe parmi les individus sur lesquels on a observé les comportements modélisés par le motif X , ceux qui n'ont pas les comportements désignés par au moins une des variables constituant le motif Y (partie hachurée sur la figure 4.1). Il faut alors déterminer jusqu'où ce contre-exemple reste une exception non significative. Dans cette situation, la validité de la quasi-règle $X \rightarrow Y$, va être évaluée en fonction de la rareté des contres-exemples à la règle par rapport à la taille de la population, des cardinaux des ensembles X' et Y' pour mesurer mathématiquement la validité d'une règle approximative (ou quasi-règle). Considérons un ensemble fini de variables $\mathcal{V} = \{x, y, \dots\}$. A un ensemble fini \mathcal{I} de n individus, on associe les fonctions :

- $i \rightarrow x(i)$, où $x(i) = 1$ (ou $x(i) = \text{vrai}$ si i possède le caractère x ;
- et $x(i) = 0$ (ou $x(i) = \text{faux}$ sinon

La règle $x \rightarrow y$ est vraie si pour tout i de l'échantillon, $y(i)$ n'est nul que dans le cas où $x(i)$ l'est aussi, i.e. si l'ensemble Z des i pour lesquels $x(i) = 1$ est contenu dans l'ensemble T des i pour lesquels $y(i) = 1$. Relativement aux cardinaux de \mathcal{I} (soit n), mas aussi de Z (soit n_x) et T (soit n_y), c'est le poids des contre-exemples (soit $n_{x \wedge \bar{y}}$ qu'il faudra prendre en compte pour accepter statistiquement de conserver ou non la quasi-règle $x \rightarrow y$.

Pour mathématiser cette quasi-règle, nous considérons deux parties quelconques T et Z de \mathcal{I} , choisies aléatoirement et indépendamment (absence de lien a priori entre ces deux parties) et de mêmes cardinaux respectifs que X et Y . Soit \bar{Y} et \bar{Z} les complémentaires respectifs de Y et de Z dans \mathcal{I} de même cardinal $n_{\bar{y}} = n - n_y$.

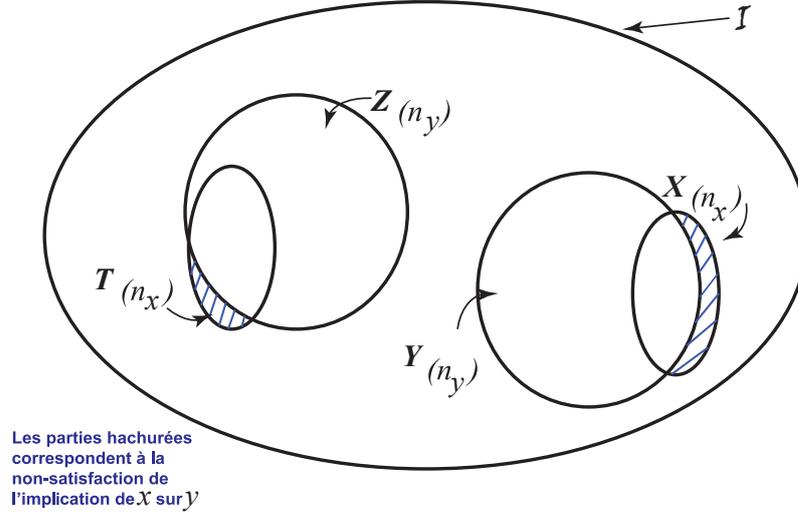


Figure 4.1 – Représentation par les diagrammes d'Euler

Définition 4.1. La quasi-règle $x \rightarrow y$ est admissible au niveau de confiance $1 - \alpha$ si et seulement si $P[(\text{Card}(X \cap \bar{Y}) \leq \text{Card}(Z \cap \bar{T}))] \leq \alpha$.

Ce qui signifie que le quasi-implication $x \rightarrow y$ sera admissible à l'issue d'une expérience si le nombre d'individus de \mathcal{I} la contredisant est *invraisemblablement petit* sous l'hypothèse d'absence de lien.

Définition 4.2. On appelle intensité d'implication de la quasi-règle $X \rightarrow Y$, le nombre :

$$\varphi(x, y) = \begin{cases} P[(\text{Card}(X \cap \bar{Y}) \leq \text{Card}(Z \cap \bar{T}))] & \text{si } n_y \neq n; \\ 0 & \text{si } n_y = n. \end{cases} \quad (4.1)$$

Définition 4.3. On appelle indice d'implication de la quasi-règle $x \rightarrow y$, la variable aléatoire, $Q(x, \bar{y})$, déduite de la variable aléatoire $\text{Card}(X \cap \bar{Y})$ par centrage et réduction.

$$Q(x, \bar{y}) = \frac{\text{card}(X \cap \bar{Y}) - \frac{n_x n_{\bar{y}}}{n}}{\sqrt{\frac{n_x n_{\bar{y}}}{n}}} \quad (4.2)$$

La valeur empirique de l'indice

$$q(x, \bar{y}) = \frac{n_{x\bar{y}} - \frac{n_x n_{\bar{y}}}{n}}{\sqrt{\frac{n_x n_{\bar{y}}}{n}}} \quad (4.3)$$

Trois modèles sont retenus pour définir la loi de probabilité de la variable aléatoire « nombre de contre-exemples » afin de déterminer la valeur de l'intensité d'implication (modèle hypergéométrique, binomial, et poissonnien).

- hypergéométrique de paramètres (n, n_x, n_y) sous l'hypothèse H_1 ;
- binomial de paramètres $(n, P(X') \times P(Y'))$ sous l'hypothèse H_2 ;
- poisson de paramètres $(n, n \times P(X') \times P(Y'))$ sous l'hypothèse H_3 .

Avec $H = \{H_1, H_2, H_3\}$ l'ensemble des trois formes fondamentales de l'hypothèse d'absence de lien (Lerman et al., 1981). On obtient, les trois indices suivants :

$$q_1(X, Y) = \sqrt{n} \times \frac{P(X' \wedge Y') - P(X') \times P(Y')}{\sqrt{P(X') \times P(Y') \times P(\bar{X}') \times P(\bar{Y}')}} \quad (4.4)$$

$$q_2(X, Y) = \sqrt{n} \times \frac{P(X' \wedge Y') - P(X') \times P(Y')}{\sqrt{P(X') \times P(Y') \times [1 - P(X') \times P(Y')]} } \quad (4.5)$$

$$q_3(X, Y) = \sqrt{n} \times \frac{P(X' \wedge Y') - P(X') \times P(Y')}{\sqrt{P(X') \times P(Y')}} \quad (4.6)$$

avec $n_X = |X'|$, $n_Y = |Y'|$ et $n = \text{card}(\mathcal{I})$.

4.2.2 Intensité d'implication

L'intensité d'implication (Gras, 1979) évalue la petitesse de contre-exemples à la règle $X \rightarrow Y$ par rapport à la quantité attendue sous l'hypothèse d'indépendance, et est définie par :

$$\varphi(X, \bar{Y}) = 1 - \Phi(q_3(X, \bar{Y})) \quad (4.7)$$

où Φ est la fonction de répartition de la loi normale centrée et réduite.

Définition 4.4. Dans le cas où $n_Y \neq n$, l'intensité d'implication de X sur Y est donnée par :

$$\varphi(X, \bar{Y}) = 1 - P[Q(X, \bar{Y}) \leq q_3(X, \bar{Y})] = \frac{1}{\sqrt{2\pi}} \int_{q_3(X, \bar{Y})}^{\infty} e^{-\frac{t^2}{2}} dt \quad (4.8)$$

Évaluation de l'intensité d'implication Le calcul de l'intensité d'implication dépend de la distribution de la variable aléatoire $\text{Card}(Z \cap \bar{T})$ qui, à son tour, dépend des hypothèses des tirages (Gras, Kuntz, et Briand, 2001; Gras et Régnier, 2009). On

peut choisir une modélisation à l'issue de laquelle la variable aléatoire $Card(Z \cap \bar{T})$ va suivre la loi de Poisson de paramètre $\lambda = \frac{n_X n_{\bar{Y}}}{n}$ ($\lambda = n \times \frac{n_X}{n} \times \frac{n_{\bar{Y}}}{n}$), avec $n = |\mathcal{I}|$, $n_X = |X'|$ et $n_{\bar{Y}} = |\bar{Y}'|$. Dans ce cas, on a :

$$\begin{aligned} \varphi(X, \bar{Y}) &= 1 - P[Card(Z \cap \bar{T}) \leq n_{X\bar{Y}}], \\ &= 1 - \sum_{k=0}^{n_{X\bar{Y}}} \frac{\lambda^k}{k!} e^{-\lambda}. \end{aligned}$$

Pour un $\lambda > 5$, après un centrage et une réduction, la variable aléatoire $Card(Z' \cap \bar{T}')$ va suivre une loi Gaussienne centrée et réduite $\left(\frac{Card(Z \cap \bar{T}) - \lambda}{\sqrt{\lambda}} \rightsquigarrow \mathcal{N}(0, 1) \right)$.

Exemple 4.2.1. Prenons le Tableau 4.1 ci-dessous :

	Y	\bar{Y}	Σ
X	5	1	6
\bar{X}	10	84	94
Σ	15	85	100

Tableau 4.1 – Tableau d'exemple pour calculer l'intensité d'implication de la quasi-règle $X \rightarrow Y$

Nous avons $Card(X \cap \bar{Y}) = n_{X\bar{Y}} = 1$. Alors, en appliquant l'équation (4.3), on a $q(X, \bar{Y}) = -1.816$ et en appliquant l'équation (4.8), l'intensité d'implication est : $\varphi(X, \bar{Y}) = \frac{1}{\sqrt{2\pi}} \int_{-1.816}^{\infty} e^{-\frac{t^2}{2}} dt = 0.9653$ Dans ce cas, on peut dire que $X \rightarrow Y$ est admissible au niveau de confiance 96.53%.

À partir du concept d'intensité d'implication découle un bon nombre d'outils d'analyse des données, en l'occurrence les graphes implicatifs, l'arbre hiérarchique, la notion de cohésion, etc. Toutes ces notions se trouvent à la base de la conception du logiciel CHIC (Classification Hiérarchique Implicative et Cohésitive). Dans le cadre de notre thèse, nous choisissons la mesure M_{GK} pour la conception d'un outil d'analyse de données dénommé CHIC- M_{GK} .

4.3 Mesure de qualité des règles d'association M_{GK}

Le modèle du facteur de certitude (CF)¹ est une méthode de gestion de l'incertitude dans les systèmes basés sur des règles. Shortliffe et Buchanan (1975) ont développé le modèle CF au milieu des années 1970 pour MYCIN, un système expert pour le diagnostic et le traitement de la méningite et des infections du sang. Depuis lors, le modèle des CF est devenu l'approche standard de la gestion de l'incertitude dans les systèmes fondés sur des règles et c'est l'origine de la mesure de qualité M_{GK} .

Depuis, la mesure d'intérêt M_{GK} des règles d'association prend diverses appellations de façon indépendante, selon les chercheurs et l'année de sa découverte : Inspiré par l'indice de Loevinger, M_{GK} (mesure de Guillaume-Kenchaff) a été indépendamment proposée et dénommée en 2000 par Guillaume (2000), *CPIR* (Conditional Probability Incrementation Ratio) en 2004 par Wu, Zhang, et Zhang (2004), *ION* (Implication Oriented Normalized) en 2003 par Totohasina (2003), vérifiant la propriété implicative orientée de Brin, Motwani, et Silverstaein (1997), en 1997. En raison de l'expression d'un rapport de condition minimale et d'efficacité pour extraire les règles non redondantes, et en appliquant le *Support* et la *Confiance* des implications $conf_G$ de Ferré (2002) montre que cette mesure est à la fois plus précise et compréhensible.

Soient $\mathbb{K} = (\mathcal{I}, \mathcal{V}, \mathcal{R})$ un contexte de la fouille de données binaires, X et Y deux motifs de \mathbb{K} . Pour un motif X , son dual X' est défini par $\bar{X} = \mathcal{V} - X$. On dit que :

- X et Y sont incompatibles, si $X' \cap Y' = \emptyset$ soit $P(X' \cap Y') = 0$;
- X favorise Y (équivalent à Y favorise X), si $P(Y'|X') > P(Y')$: X apporterait une information positive sur la réalisation de Y ;
- X et Y sont statistiquement indépendants, si $P(Y'|X') = P(Y')$: X n'apporterait aucune information sur la réalisation de Y et vice versa ;
- X défavorise Y (équivalent à Y défavorise X), si $P(Y'|X') < P(Y')$: X apporterait une information négative sur la réalisation de Y et vice versa ;
- X implique logiquement Y , si $X' \subseteq Y'$, i.e. $P(Y'|X') = 1$.

1. en anglais CF : Certainty Factor

4.3.1 Construction de la mesure M_{GK}

1. Si X défavorise Y , alors on a : $P(Y'|X') < P(Y')$, soit $P(Y'|X') - P(Y') < 0$. Dans le cas de l'indépendance de X et Y , on a : $P(Y'|X') - P(Y') = 0$. Par ailleurs, $0 \leq P(Y'|X')$ implique $-P(Y') \leq P(Y'|X') - P(Y')$. L'égalité est atteinte au cas de l'incompatibilité entre X et Y . En considérant l'incompatibilité comme situation limite de dépendance négative, on peut écrire : $-P(Y') \leq P(Y'|X') - P(Y') \leq 0$, si X défavorise Y , soit

$$-1 \leq \frac{P(Y'|X') - P(Y')}{P(Y')} \leq 0, \text{ si } X \text{ défavorise } Y \quad (4.9)$$

2. Si X favorise Y , alors on a : $P(Y'|X') > P(Y')$, soit $P(Y'|X') - P(Y') > 0$. Dans le cas de l'indépendance de X et Y , on a : $P(Y'|X') - P(Y') = 0$. Par ailleurs, $P(Y'|X') \leq 1$ implique $P(Y'|X') - P(Y') \leq 1 - P(Y')$. L'égalité est atteinte au cas de l'implication logique entre X et Y . En considérant l'implication logique comme situation limite de dépendance positive, on peut écrire : $-P(Y') \leq P(Y'|X') - P(Y') \leq 0$, si X favorise Y , soit

$$0 \leq \frac{P(Y'|X') - P(Y')}{1 - P(Y')} \leq 1, \text{ si } X \text{ favorise } Y \quad (4.10)$$

Grâce à ces deux propriétés sur les probabilités, il apparaît logique de poser la définition suivante.

Définition 4.5. Soit X et Y deux motifs d'un contexte de fouille de données. On définit M_{GK} par :

$$M_{GK}(X \rightarrow Y) = \begin{cases} \frac{P(Y'|X') - P(Y')}{1 - P(Y')}, \text{ si } X \text{ favorise } Y; \\ \frac{P(Y'|X') - P(Y')}{P(Y')}, \text{ si } X \text{ défavorise } Y \end{cases} \quad (4.11)$$

avec $M_{GK}^f(X \rightarrow Y) = \frac{P(Y'|X') - P(Y')}{1 - P(Y')}$ que nous appelons la composante favorisante de la règle $X \rightarrow Y$, et $M_{GK}^d(X \rightarrow Y) = \frac{P(Y'|X') - P(Y')}{P(Y')}$ sa composante défavorisante. Alors, la relation (4.11) peut prendre la forme :

$$M_{GK}(X \rightarrow Y) = \begin{cases} M_{GK}^f(X \rightarrow Y), \text{ si } X \text{ favorise } Y; \\ M_{GK}^d(X \rightarrow Y), \text{ si } X \text{ défavorise } Y \end{cases} \quad (4.12)$$

Ainsi, surtout et essentiellement la composante M_{GK}^f qui va guider la sémantique

de M_{GK} ; de plus, les deux composantes pourront s'écrire en fonction de la confiance de la règle, car en fait, à travers ses deux composantes, M_{GK} exprime un taux d'incrémentement de probabilités, en effet :

$$M_{GK}^f(X \rightarrow Y) = \frac{\text{conf}(X \rightarrow Y) - \text{supp}(Y)}{1 - \text{supp}(Y)} \text{ avec } \text{supp}(Y) \neq 1 \text{ et}$$

$$M_{GK}^d(X \rightarrow Y) = \frac{\text{conf}(X \rightarrow Y) - \text{supp}(Y)}{\text{supp}(Y)} \text{ avec } \text{supp}(Y) \neq 0$$

La relation (4.11) devient ainsi :

$$M_{GK}(X \rightarrow Y) = \begin{cases} \frac{\text{conf}(X \rightarrow Y) - \text{supp}(Y)}{1 - \text{supp}(Y)}, & \text{si } X \text{ favorise } Y \text{ et } \text{supp}(Y) \neq 1; \\ \frac{\text{conf}(X \rightarrow Y) - \text{supp}(Y)}{\text{supp}(Y)}, & \text{si } X \text{ défavorise } Y \text{ et } \text{supp}(Y) \neq 0 \end{cases} \quad (4.13)$$

Pour deux motifs X et Y non indépendants, l'un des deux cas suivants peut se produire :

- Cas 1 : l'attraction mutuelle, i.e. la dépendance est positive entre le deux motifs X et Y et l'on considère $X \rightarrow Y$;
- Cas 2 : la répulsion, il existe toujours une dépendance positive entre le deux motifs X et \bar{Y} et l'on considère $X \rightarrow \bar{Y}$ d'une part, puis entre \bar{X} et Y et l'on considère $\bar{X} \rightarrow Y$ d'autre part, ce sont les deux cas importants de règles négatives.

Dans ces deux cas, on aura toujours à considérer une dépendance orientée positive extraite à partir de M_{GK}^f .

4.3.2 Propriétés de la mesure de qualité des règles d'association M_{GK}

Soient $\mathbb{K} = (\mathcal{I}, \mathcal{V}, \mathcal{R})$ un contexte de la fouille de données binaires, X et Y deux motifs de \mathbb{K} , tels que $P(X') \neq 0$, $P(Y') \neq 0$, $P(X') \neq 1$, $P(Y') \neq 1$.

Lemme 4.1. (Feno, 2007)

- (i) X favorise Y , si et seulement si Y favorise X .
- (ii) X défavorise Y , si et seulement si Y défavorise X .

Démonstration.

$$\begin{aligned}
 (i) \quad X \text{ favorise } Y &\Leftrightarrow P(Y'|X') > P(Y') \\
 &\Leftrightarrow P(X' \cap Y') > P(X')P(Y') \\
 &\Leftrightarrow P(X'|Y') > P(X') \\
 &\Leftrightarrow Y \text{ favorise } X
 \end{aligned}$$

(ii) se démontre de la même manière □

Lemme 4.2 (Dépendance positive et dépendance négative).

(1)- Les trois conditions suivantes sont équivalents : (i) X défavorise Y ,
(ii) X favorise \bar{Y} et (iii) \bar{X} favorise Y .

(2)- Les quatres conditions suivantes sont équivalents : (i) X favorise Y ,
(ii) X défavorise \bar{Y} , (iii) \bar{X} favorise \bar{Y} et \bar{X} défavorise Y .

Démonstration.

(1)- Montrons que (i) \Leftrightarrow (ii)

$$\begin{aligned}
 X \text{ défavorise } Y &\Leftrightarrow P(Y'|X') < P(Y') \\
 &\Leftrightarrow -P(Y'|X') > -P(Y') \\
 &\Leftrightarrow 1 - P(Y'|X') > 1 - P(Y') \\
 &\Leftrightarrow P(\bar{Y}'|X') > P(\bar{Y}') \\
 &\Leftrightarrow X \text{ défavorise } \bar{Y}
 \end{aligned}$$

(1)- Montrons que (i) \Leftrightarrow (iii)

$$\begin{aligned}
 X \text{ défavorise } Y &\Leftrightarrow Y \text{ défavorise } X \\
 &\Leftrightarrow P(X'|Y') < P(X') \\
 &\Leftrightarrow -P(X'|Y') > -P(X') \\
 &\Leftrightarrow 1 - P(X'|Y') > 1 - P(X') \\
 &\Leftrightarrow P(\bar{X}'|Y') > -P(\bar{X}') \\
 &\Leftrightarrow Y \text{ favorise } \bar{X} \\
 &\Leftrightarrow \bar{X} \text{ favorise } Y
 \end{aligned}$$

(2)- Montrons que (i) \Leftrightarrow (ii)

Il suffit d'appliquer (1) à X défavorise \bar{Y} .

En effet, X défavorise $\bar{Y} \Leftrightarrow X$ favorise $\bar{\bar{Y}}$, alors X favorise Y .

Les autres équivalences se démontrent de même manière. □

Proposition 4.1. (Situation de référence). Pour tous motifs X et Y , on a :

- X et Y sont incompatibles, si et seulement si $M_{GK}(X \rightarrow Y) = -1$;
- X défavorise Y , si et seulement si $-1 < M_{GK}(X \rightarrow Y) < 0$;
- X et Y sont indépendants, si et seulement si $M_{GK}(X \rightarrow Y) = 0$;
- X favorise Y , si et seulement si $0 < M_{GK}(X \rightarrow Y) < 1$;
- X implique logiquement Y , si et seulement si $M_{GK}(X \rightarrow Y) = 1$.

Proposition 4.2. (Situation de référence à l'équilibre)(Totohasina et Ralambondrainy, 2005),(Diatta, Ralambondrainy, et Totohasina, 2007).

A l'équilibre : $M_{GK}(X \rightarrow Y) \approx \frac{1}{2}$

M_{GK} est une mesure de déviation d'équilibre pour n suffisamment grand.

Démonstration. Une règle pourrait intéressante si la prémisse favorise le conséquent. Soit une règle d'association $X \rightarrow Y$ telle que X favorise Y . A l'équilibre $|X' \cap Y'| = |X' \cap \bar{Y}'|$ et notons $n = |\mathcal{E}|$. Pour n suffisamment grand, nous avons l'approximation :

$$\begin{aligned} M_{GK}(X \rightarrow Y) &= \frac{P(Y'|X') - P(Y')}{1 - P(Y')} \\ &= \frac{\frac{1}{2} - \frac{|Y'|}{n}}{1 - \frac{|Y'|}{n}} \\ &= \frac{1}{2} + o\left(\frac{1}{n}\right) \\ &\approx \frac{1}{2} \end{aligned}$$

□

Proposition 4.3. La mesure M_{GK} est une mesure statistique car elle est sensible à la taille de données.

Proposition 4.4. Soient deux motifs X et Y .

(i) Si X favorise Y , on a la relation :

$$M_{GK}^f(Y \rightarrow X) = \frac{1 - P(Y')}{1 - P(X')} \frac{P(X')}{P(Y')} M_{GK}^f(X \rightarrow Y) \quad (4.14)$$

(ii) Si X défavorise Y , on a la relation :

$$M_{GK}^d(Y \rightarrow X) = M_{GK}^d(X \rightarrow Y) \quad (4.15)$$

Démonstration. Vérifions la symétrie de la mesure dans les deux cas : *favorisant* et *défavorisant*

(i) Si X favorise Y , alors

$$\begin{aligned}
 M_{GK}^f(Y \rightarrow X) &= \frac{P(X'|Y')-P(X')}{1-P(X')} \\
 &= \frac{P(X' \cap Y')-P(X')P(Y')}{P(Y')(1-P(X'))} \\
 &= \frac{P(X')}{P(Y')} \frac{1-P(Y')}{1-P(X')} \frac{P(X' \cap Y')-P(X')P(Y')}{P(X')(1-P(Y'))} \\
 &= \frac{P(X')}{P(Y')} \frac{1-P(Y')}{1-P(X')} M_{GK}^f(X \rightarrow Y)
 \end{aligned}$$

(ii) Si X défavorise Y , alors

$$\begin{aligned}
 M_{GK}^d(Y \rightarrow X) &= \frac{P(X'|Y')-P(X')}{P(X')} \\
 &= \frac{P(X' \cap Y')-P(X')P(Y')}{P(Y')(P(X'))} \\
 &= M_{GK}^d(X \rightarrow Y)
 \end{aligned}$$

□

Corollaire 4.1. *La mesure M_{GK}^f est non symétrique, autrement dit la mesure M_{GK} est favorablement non symétrique.*

Proposition 4.5. *Soient les motifs X et Y .*

(i) *Si X favorise Y , on a la relation d'équivalence des deux règles contraposées suivante :*

$$M_{GK}^f(\bar{Y} \rightarrow \bar{X}) = M_{GK}^f(X \rightarrow Y) \quad (4.16)$$

(ii) *Si X défavorise Y , on a la relation :*

$$M_{GK}^d(Y \rightarrow X) = \frac{P(X')P(Y')}{(1-P(X'))(1-P(Y'))} M_{GK}^d(X \rightarrow Y) \quad (4.17)$$

Démonstration. Pour les deux motifs X et Y ,

(i) Si X favorise Y , on a

$$\begin{aligned}
 M_{GK}^f(\bar{Y} \rightarrow \bar{X}) &= \frac{P(\bar{X}'|\bar{Y}')-P(\bar{X}')}{1-P(\bar{X}')} \\
 &= \frac{1-P(X'|\bar{Y}')-1+P(X')}{P(X')} \\
 &= \frac{-P(X' \cap \bar{Y}')+P(X')P(\bar{Y}')}{P(X')(1-P(Y'))} \\
 &= \frac{-P(X')+P(X' \cap Y')+P(X')-P(X')P(Y')}{P(X')(1-P(Y'))} \\
 &= \frac{P(X' \cap Y')-P(X')P(Y')}{P(X')(1-P(Y'))} \\
 &= M_{GK}^f(X \rightarrow Y)
 \end{aligned}$$

(ii) Si X défavorise Y , on a

$$\begin{aligned}
 M_{GK}^d(\bar{Y} \rightarrow \bar{X}) &= \frac{P(\bar{X}'|\bar{Y}')-P(\bar{X}')}{P(\bar{X}')} \\
 &= \frac{1-P(X'|\bar{Y}')-1+P(X')}{1-P(X')} \\
 &= \frac{-P(X'\cap\bar{Y}')+P(X')P(\bar{Y}')}{1-P(X')(1-P(Y'))} \\
 &= \frac{-P(X')+P(X'\cap Y')+P(X')-P(X')P(Y')}{1-P(X')(1-P(Y'))} \\
 &= \frac{P(X'\cap Y')-P(X')P(Y')}{1-P(X')(1-P(Y'))} \\
 &= \frac{P(X')P(Y')}{1-(P(X')(1-P(Y')))} \frac{P(X'\cap Y')-P(X')P(Y')}{P(X')P(Y')} \\
 &= \frac{P(X')P(Y')}{1-(P(X')(1-P(Y')))} M_{GK}^d(X \rightarrow Y)
 \end{aligned}$$

□

Corollaire 4.2. *La mesure M_{GK} est favorablement implicative.*

Proposition 4.6. *Soient deux motifs X et Y , on a l'égalité et l'équivalence suivantes :*

$$M_{GK}^f(X \rightarrow \bar{Y}) = -M_{GK}^d(X \rightarrow Y) \quad (4.18)$$

Et on a, :

$$(-1 < -M_{GK}^d(X \rightarrow Y) < -\alpha \Leftrightarrow \alpha < M_{GK}^f(X \rightarrow \bar{Y}) < 1) \quad (4.19)$$

Démonstration.

$$\begin{aligned}
 M_{GK}^f(X \rightarrow \bar{Y}) &= \frac{P(\bar{Y}'|X')-P(\bar{Y}')}{1-P(\bar{Y}')} \\
 &= \frac{1-P(Y'|X')-1+P(Y')}{1-1+P(Y')} \\
 &= \frac{-P(Y'|X')+P(Y')}{P(Y')} \\
 &= -M_{GK}^d(X \rightarrow Y)
 \end{aligned}$$

□

De la proposition 4.6 résulte que plus le degré de quasi-incompatibilité entre deux motifs est élevé, plus la qualité de la règle négative à droite correspondante est favorablement meilleure. Alors l'équivalence de l'équation (4.19) permet d'élaguer directement toute candidate règle négative à droite dont la valeur M_{GK} de la règle positive associée est négative et située dans l'intervalle $[-\alpha, 0[$, pour un seuil α fixé dans $]0, 1[$.

Proposition 4.7. *Pour deux motifs X et Y , on a les égalités suivantes :*

(1) *Si X défavorise Y (X favorise \bar{Y} et aussi \bar{X} favorise Y), alors*

$$M_{GK}^f(\bar{X} \rightarrow Y) = \frac{P(X')}{1-P(X')} \frac{P(Y')}{1-P(Y')} M_{GK}^f(X \rightarrow \bar{Y}) \quad (4.20)$$

(2) Si X favorise Y (X défavorise \bar{Y} et aussi \bar{X} défavorise Y), alors

$$M_{GK}^d(\bar{X} \rightarrow Y) = \frac{P(X')}{1 - P(X')} \frac{1 - P(Y')}{P(Y')} M_{GK}^d(X \rightarrow \bar{Y}) \quad (4.21)$$

Démonstration.

Pour (1)

$$\begin{aligned} M_{GK}^f(\bar{X} \rightarrow Y) &= \frac{P(\bar{Y}'|X') - P(\bar{Y}')}{1 - P(\bar{Y}')} \\ &= \frac{P(Y') - P(X' \cap Y') - P(Y')(1 - P(X'))}{(1 - P(Y'))(1 - P(X'))} \\ &= \frac{P(Y') - P(X' \cap Y') - P(Y')P(X')}{(1 - P(Y'))(1 - P(X'))} \\ &= - \frac{P(X')(P(Y'|X') - P(Y'))}{(1 - P(X'))(1 - P(Y'))} \\ &= \frac{P(X')}{1 - P(X')} \frac{P(Y')}{1 - P(Y')} M_{GK}^f(X \rightarrow \bar{Y}). \end{aligned}$$

Pour (2)

$$\begin{aligned} M_{GK}^d(\bar{X} \rightarrow Y) &= \frac{P(Y'|\bar{X}') - P(Y')}{P(Y')} \\ &= \frac{P(Y') - P(X' \cap Y') - P(Y')(1 - P(X'))}{(1 - P(X'))P(Y')} \\ &= \frac{P(Y') - P(X' \cap Y') - P(Y') - P(Y')P(X')}{(1 - P(X'))P(Y')} \\ &= - \frac{P(X')(1 - P(Y'))(P(Y'|X') - P(Y'))}{(1 - P(X'))(P(Y'))(1 - P(Y'))} \\ &= \frac{P(X')}{1 - P(X')} \frac{1 - P(Y')}{P(Y')} M_{GK}^d(X \rightarrow \bar{Y}). \end{aligned}$$

□

Selon les deux propositions 4.6 et 4.7, on peut tirer les relations entre règle négative à gauche et la règle positive correspondante.

Corollaire 4.3. Pour deux motifs X et Y , on a les égalités suivantes :

(1) Si X défavorise Y (X favorise \bar{Y} et aussi \bar{X} favorise Y), alors

$$M_{GK}^f(\bar{X} \rightarrow Y) = \frac{P(X')}{1 - P(X')} \frac{P(Y')}{1 - P(Y')} M_{GK}^d(X \rightarrow Y) \quad (4.22)$$

(2) Si X favorise Y (X défavorise \bar{Y} et aussi \bar{X} défavorise Y), alors

$$M_{GK}^d(\bar{X} \rightarrow Y) = \frac{P(X')}{1 - P(X')} \frac{1 - P(Y')}{P(Y')} M_{GK}^f(X \rightarrow Y) \quad (4.23)$$

Selon le corollaire (4.3), on obtient une relation entre seuils permettant de sélectionner des règles négatives à gauche intéressantes ou d'élaguer celles non intéressantes.

Corollaire 4.4. Pour deux motifs X et Y , on a les égalités suivantes. Si X favorise Y (X défavorise \bar{Y} et aussi \bar{X} défavorise Y), alors pour un seuil $\alpha \in]0, 1[$ fixé :

$$-1 < M_{GK}^d(X \rightarrow Y) < -\alpha \Leftrightarrow \frac{P(X')}{1-P(X')} \frac{P(Y')}{1-P(Y')} \alpha < M_{GK}^f(\overline{X} \rightarrow Y) < \frac{P(X')}{1-P(X')} \frac{P(Y')}{1-P(Y')}$$

Proposition 4.8 (multiplicativité). *Aux motifs emboîtés sont associées des règles d'association exactes ou implications totales.*

De plus :

(i) Si $X \subseteq Y \subseteq Z \subseteq \mathcal{V}$, alors

$$M_{GK}(X \rightarrow Z) = M_{GK}(X \rightarrow Y)M_{GK}(Y \rightarrow Z)$$

(ii) Si $X_1 \subseteq X_2 \subseteq \dots \subseteq X_i \subseteq X_{i+1} \subseteq \dots \subseteq X_p \subseteq \mathcal{V}$, alors

$$M_{GK}(X_1 \rightarrow X_p) = \prod_{i=1}^{p-1} M_{GK}(X_i \rightarrow X_{i+1})$$

M_{GK} est multiplicative sur une chaîne de treillis motifs, ce qui paraît intéressant sur le plan d'élagage grâce à la transitivité axiale.

Démonstration.

(i) En passant aux extensions $X \subseteq Y \subseteq Z \Rightarrow Z' \subseteq Y' \subseteq X'$. En effet,

$P(Y'|X') = \frac{P(X' \cap Y')}{P(X')} = \frac{P(Y')}{P(X')} \geq P(Y')$. Par conséquent, X favorise Y . De même, par analogie, X favorise Z et Y favorise Z . Ainsi les trois motifs X, Y et Z se favorisent

deux à deux. On a alors :

$$\begin{aligned} M_{GK}(X \rightarrow Y) &= \frac{\frac{P(X' \cap Y')}{P(X')} - P(Y')}{1 - P(Y')} \\ &= \frac{\frac{P(Y')}{P(X')} - P(Y')}{1 - P(Y')} \\ &= \frac{P(Y')(1 - P(X'))}{P(X')(1 - P(Y'))}. \end{aligned}$$

Nous avons en effet,

$$\begin{aligned} M_{GK}(X \rightarrow Y)M_{GK}(Y \rightarrow Z) &= \frac{P(Y')(1 - P(X'))}{P(X')(1 - P(Y'))} \frac{P(Z')(1 - P(Y'))}{P(Y')(1 - P(Z'))} \\ &= \frac{P(Z')(1 - P(X'))}{P(X')(1 - P(Z'))} \\ &= M_{GK}(X \rightarrow Z). \end{aligned}$$

(ii) s'obtient facilement à l'aide d'un raisonnement par récurrence. \square

Corollaire 4.5. *Soient $X_1, X_2, \dots, X_i, X_{i+1}, \dots, X_p$ des motifs tels que $X_1 \subseteq X_2 \subseteq \dots \subseteq X_i \subseteq X_{i+1} \subseteq \dots \subseteq X_p$.*

(i) *Si $X_1 \rightarrow X_p$ est (M_{GK}, α) -valide alors $\forall i, j \in \{1, \dots, p\}$, avec $i < j$, $X_i \rightarrow X_j$ est (M_{GK}, α) -valide.*

(ii) *Si il existe $i, j \in \{1, \dots, p\}$ tels que $X_i \rightarrow X_j$ est non (M_{GK}, α) -valide alors $\forall l, k \in \{1, \dots, p\}$ tels que $l \leq i$ et $j \leq k$, $X_l \rightarrow X_k$ est aussi non (M_{GK}, α) -valide.*

Proposition 4.9.

(1) Aux marginales fixées, *Confiance* et M_{GK}^f sont deux fonctions croissantes du nombre d'exemples de règle, cependant M_{GK}^f croît plus lentement que *Confiance*.

(2) $\forall (X \rightarrow Y)$ telle que X favorise Y , alors

(i) Si leurs extensions sont telles que $X' \subseteq Y'$, alors

$conf(X \rightarrow Y) = M_{GK}^f(X \rightarrow Y) = 1$: $X \rightarrow Y$ est dite une règle exacte.

(ii) Si $X' \subseteq Y'$, alors $0 < M_{GK}^f(X \rightarrow Y) < conf(X \rightarrow Y)$,

ou encore $\frac{1-M_{GK}^f}{1-conf}(X \rightarrow Y) > 1$: $X \rightarrow Y$ est dite une règle approximative.

Démonstration. Pour deux motifs X et Y , posons $t = |X' \cap Y'|$, alors : (1)

$$conf(X \rightarrow Y) = \frac{t}{n_X} = f_1(t) \text{ et } M_{GK}^f(X \rightarrow Y) = \frac{\frac{t}{n_X} - P(Y')}{1 - P(Y')} = f_2(t).$$

Comparons les dérivées de ces deux fonctions :

$$0 < f_2'(t) = \frac{1}{n_X(1-P(Y'))} < f_1'(t) = \frac{1}{n_X}.$$

Ceci dit que *Confiance* croît plus vite vers la valeur maximale 1 que M_{GK} dans la zone d'attraction.

Le résultat (2)(i) est évident. Enfin, pour (2)(ii), on vérifie sous cette hypothèse :

$$\frac{1-M_{GK}^f}{1-conf}(X \rightarrow Y) = \frac{1}{P(Y')} > 1$$

□

4.3.3 Commentaire et plaidoyer sur M_{GK}

Plusieurs travaux ont été déjà faits [Totohasina \(2003\)](#), [Totohasina, Ralambondrainy, et Diatta \(2004\)](#), [Totohasina et Ralambondrainy \(2005\)](#), [Feno \(2007\)](#), [Diatta et al. \(2007\)](#), [Totohasina et Feno \(2008\)](#) détaillant les principales propriétés de la mesure de qualité des règles M_{GK} : elle prend ses valeurs sur l'intervalle $[-1, 1]$ tout en reflétant les situations de référence telles que l'incompatibilité, la dépendance négative, l'indépendance, la dépendance positive et l'implication logique entre la prémisse et le conséquent d'une règle. De plus, elle est une mesure de déviation à l'équilibre pour n suffisamment grand [Diatta et al. \(2007\)](#). C'est une mesure statistique, car elle est sensible à la taille de données. Elle permet de calculer les règles négatives valides à partir des règles positives correspondantes non valides. Elle a également une propriété multiplicative, ce qui nous permet l'élagage des branches de transitivité sur un diagramme de Hasse partiel ou dans un graphe sous-treillis a priori dense. Elle sélectionne nettement moins de règles que la mesure *confiance* pour un

même seuil de validité α fixé par l'utilisateur ($M_{GK}(X \rightarrow Y) \leq \text{conf}(X \rightarrow Y)$). Enfin, le plus important selon nous, c'est sa propriété *non symétrique*, car elle est une mesure implicative dans la zone d'attraction au sens de l'implication logique, à l'instar de ce que l'on a en logique formelle, où deux implications contraposées ont la même valeur logique, ($M_{GK}^f(\bar{Y} \rightarrow \bar{X}) = M_{GK}^f(X \rightarrow Y)$), ainsi nous pouvons dire que M_{GK} est favorablement non symétrique, et M_{GK}^f (composante favorisante) est appelée aussi le **déclencheur de prise de décision**. Pour la suite de nos études, il est heureusement incontournable de prendre toujours la composante favorisante M_{GK}^f qui est implicative.

4.4 Extraction des règles d'association M_{GK} -valides avec contribution du *Support*

L'étude des règles d'association entre variables booléennes liée à l'analyse des tableaux croisés 2 x 2 est ancienne. L'équipe de Hajek., Havel., et Cyril (1966) a initié l'une des premières méthodes de recherche des règles d'association qui s'appelaient GUHA². Les mesures pionnières *Support* et *Confiance* proposées par Agrawal et al. (1993b) ne sont pas à l'abri des critiques et ont provoqué la prolifération rapide d'autres mesures plus pertinentes. En effet, la validation des règles d'association par rapport à ces deux mesures est critique, car entre autres défauts, elle élague des règles d'association valides et pertinentes, notamment dans le cas où l'utilisateur a mal choisi les seuils minimaux définis par lui-même. Face à cela, l'objectif de cette étude est alors d'extraire les règles d'association à *Supports*-valides à base de la mesure d'intérêt M_{GK} en vue de ne pas ignorer les règles d'association *pertinentes* et *significatives*, malgré leurs *supports* de faibles valeurs.

Pour mettre en évidence les relations pertinentes pour la prise de décision, il est important de mesurer la qualité des règles extraites dans le contexte donné. Le critère le plus souvent utilisé pendant les années 90 est la combinaison de *Support* et *Confiance*. Par rapport aux trois défauts de la mesure *Confiance* (Fleury, 1996): elle varie linéairement, c'est un indice fréquentiel qui est insensible à la dilatation des effectifs, et elle ne permet pas de rejeter les règles dues au hasard ou les règles

2. GUHA : General Unary Hypotheses Automation

évidentes. Ainsi, des mesures issues de statistiques nous permettent, de calculer le degré et l'existence de dépendance ou l'absence de dépendance entre les motifs, comme par exemple le χ^2 ; mais celle-ci ne nous montre pas la direction de la relation. [Brin, Motwani, et Silverstein \(1997\)](#) proposent de rechercher des motifs corrélés en exploitant conjointement le test de χ^2 et une mesure d'intérêt. Il teste l'hypothèse d'indépendance dans les données, mais sans faire valoir la probabilité de l'erreur commise lors du test. Pour pallier à ces problèmes, [Gras \(1979\)](#) a défini une mesure objective appelée *Intensité d'Implication* basée sur une modélisation probabiliste des règles d'association et cette dernière permet de mesurer la significativité statistique des règles découvertes, interprétées comme le degré d'étonnement statistique du décideur face à la petitesse du nombre de contre-exemples. Celle-ci ne nous extrait que les règles positives. Depuis le début des années 2000, une mesure appelée M_{GK} résout tous ces problèmes sus-mentionnés et nous permet aussi d'extraire et de mesurer les règles négatives en plus de celles positives.

4.4.1 Implication entre variables binaires

Considérons un contexte formel $\mathbb{K} = (\mathcal{I}, \mathcal{V}, \mathcal{R})$, où \mathcal{R} est une relation binaire sur $\mathcal{I} \times \mathcal{V}$ tel que l'appartenance d'un couple $(i, v) \in (\mathcal{I}, \mathcal{V})$ au graphe de la relation \mathcal{R} exprime le fait que l'individu i possède la variable v . Tout sous-ensemble X de \mathcal{V} s'appelle motif de \mathcal{V} . Un motif désigne indifféremment un ensemble éventuellement réduit à une variable. Considérons un espace probabilisé discret fini uniforme $(\mathcal{I}, \mathcal{P}(\mathcal{I}), P)$: pour tout $A \subset \mathcal{P}(\mathcal{I})$, $P(A) = \text{card}(A)/\text{card}(\mathcal{I})$, où \mathcal{I} est l'ensemble des n individus, sur lesquels on a mesuré m variables aléatoires de Bernoulli, et \mathcal{V} celui de ces dernières qui sont des variables binaires : $\mathcal{V} = \{v_1, v_2, v_3, \dots, v_m\}$. Pour tout $X \in \mathcal{P}(\mathcal{V}) \setminus \{\emptyset; \mathcal{V}\}$, pour tout $v_k \in \mathcal{V}$, v_k est une application de \mathcal{I} vers $\{0, 1\}$, où la valeur 1 mesure la présence de la variable chez un individu de \mathcal{I} et $P(v_k) = P(v_k = 1) = (\text{card}(v_k^{-1}(1)))/n$, avec $n = \text{card}(\mathcal{I})$. Alors pour deux motifs X et Y dans $\mathcal{P}(\mathcal{V})$: $X' = X^{-1}(1)$, $Y' = Y^{-1}(1)$, $n_X = \text{card}(X')$ et $n_Y = \text{card}(Y')$ sont les nombres de transactions qui réalisent respectivement les motifs X et Y , $n_{XY} = \text{card}(X' \cap Y')$ est celui de celles qui réalisent à la fois X et Y . \bar{X} désignera la négation de logique de X et $\bar{X}' = \mathcal{I} - X'$. Le réel $P(X')$ sera appelé le *support* du motif X , et noté $\text{supp}(X)$.

Pour deux motifs X et Y du contexte, en termes d'extension, on a $(X \wedge Y)' = X' \cap Y'$

et $(X \vee Y)' = X' \cup Y'$.

Prenons un exemple d'un contexte binaire. Considérons l'ensemble d'individus (ou d'objets) $\mathcal{I} = \{i_1, i_2, \dots, i_{30}\}$ et l'ensemble de variables (ou attributs) $\mathcal{V} = \{v_1, v_2, \dots, v_{10}\}$.

$I \setminus V$	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
i_1	0	1	0	1	0	1	1	0	1	1
i_2	0	1	0	1	0	1	1	0	1	1
i_3	0	1	0	1	0	1	1	0	0	1
i_4	0	1	0	1	0	1	1	0	0	1
i_5	0	1	1	1	1	1	1	1	0	1
i_6	0	1	1	0	1	1	1	1	1	1
i_7	1	1	1	0	1	1	1	1	1	1
i_8	1	1	1	0	1	1	1	1	1	1
i_9	1	1	1	0	1	1	1	1	1	1
i_{10}	1	1	1	0	1	0	1	1	1	1
i_{11}	0	1	0	0	1	0	1	0	1	1
i_{12}	0	1	0	1	1	0	1	0	1	0
i_{13}	0	1	0	1	1	0	1	0	1	0
i_{14}	0	1	0	1	1	0	1	0	1	0
i_{15}	0	1	0	1	1	1	1	0	0	0
i_{16}	0	0	0	1	1	1	1	0	0	0
i_{17}	0	0	0	1	1	1	0	0	0	1
i_{18}	1	0	0	1	1	1	0	0	0	1
i_{19}	1	0	0	1	1	1	0	1	0	1
i_{20}	1	0	0	1	1	0	1	1	0	1
i_{21}	1	0	0	1	1	0	1	1	0	1
i_{22}	1	0	0	0	1	0	1	1	0	1
i_{23}	1	0	0	1	1	0	0	1	1	1
i_{24}	1	1	0	0	1	0	0	1	1	1
i_{25}	1	0	1	1	1	0	1	1	1	1
i_{26}	1	0	1	0	1	0	1	1	1	0
i_{27}	1	1	0	0	1	0	1	1	1	0
i_{28}	1	1	1	1	1	0	1	1	1	0
i_{29}	1	0	1	1	1	0	1	1	1	0
i_{30}	1	1	0	1	0	0	1	1	1	0

Tableau 4.2 – Tableau d'un contexte binaire.

Dans le cas binaire, les variables ne prennent que deux valeurs : 0 et 1. La situation générique est la suivante. On croise un ensemble d'individus $\mathcal{I} = \{i_1, i_2, \dots, i_{30}\}$ et un ensemble de variables $\mathcal{V} = \{v_1, v_2, \dots, v_{10}\}$ (cf. Tableau 4.2). Dans les situations réelles de l'implication stricte de la variable v_i sur la variable v_j , situation où tout individu de \mathcal{I} qui vérifierait v_i vérifierait également v_j , on veut donner un sens statistique à une implication non stricte : $v_i \iff v_j$. Dans ce cas, certains individus peuvent vérifier v_i et non pas v_j . En termes ensemblistes, X' et Y' représentant les sous populations respectives d'individus où les variables v_i et v_j prennent la valeur 1. Il existe une analogie étroite avec la mesure de l'inclusion non stricte de X

dans Y . Au départ, la construction des implications repose sur des considérations de nature logique et algébrique, comme dans l'analyse booléenne. Dans notre cas, nous adoptons M_{GK} comme mesure de la qualité des règles d'association. Ainsi, la relation deux à deux entre variables pourrait se présenter dans un tableau de contingence comme le montre ci-dessous (Cf. Tableau 4.3) :

	v_j	1	0
v_i			
1		$n_{v_i v_j}$	$n_{v_i \bar{v}_j}$
0		$n_{\bar{v}_i v_j}$	$n_{\bar{v}_i \bar{v}_j}$

Tableau 4.3 – Tableau de contingence.

	V2	non_V2	khi2	4,4744	support(V1)	0,5666
V1	8	9	support(V2)	0,6333	similarite	-0,843
non_V1	11	2	confiance(V1,V2)	0,4705	correlation	-0,386

	V3	non_V3	khi2	3,3257	support(V1)	0,5666
V1	8	9	support(V3)	0,3333	similarite	0,9801
non_V1	2	11	confiance(V1,V3)	0,4705	correlation	0,3329

	V10	non_V10	khi2	0,625	support(V8)	0,6
V8	13	5	support(V10)	0,6666	similarite	0,2886
non_V8	7	5	confiance(V8,V10)	0,7222	correlation	0,1443

	V10	non_V10	khi2	1,7942	support(V9)	0,6333
V9	11	8	support(V10)	0,6666	similarite	-0,468
non_V9	9	2	confiance(V9,V10)	0,5789	correlation	-0,244

Tableau 4.4 – Extrait de tableaux de contingence et quelques valeurs données par le χ^2 , la *confiance*, la *similarité* et la *correlation* des couples de variables. B.11-TB.1

4.4.2 Prise en compte des valeurs critiques de M_{GK}

Le test d'indépendance de χ^2 est utile pour valider le lien implicatif entre deux variables. Dans ce cas, on peut en extraire les *valeurs critiques* de M_{GK} B.11-TB.2. Il est alors important dans le processus d'extraction des règles d'association de pouvoir établir que la dépendance observée est statistiquement significative. Tout ceci constitue des raisons pour lesquelles le test d'indépendance de χ^2 doit utilement entrer dans la validation d'une règle d'association. Il est utilisé pour valider si oui ou non, à un certain seuil fixé, le lien entre les variables est ainsi significatif. En effet, il est logiquement préférable que pour n'importe quel seuil de confiance, le résultat fourni par une quelconque mesure vis-à-vis d'un éventuel lien entre deux variables soit cohérent avec le test d'indépendance de χ^2 . Pour mettre en évidence la nécessité

d'un test d'indépendance par rapport aux mesures *confiance* et M_{GK} , nous allons considérer les deux situations ci-après (Cf. Tableau 4.5), et en appliquant la formule (3.4) de la sous-section (3.3.4),

	Y	\bar{Y}	Σ
X	1200	100	1300
\bar{X}	300	600	900
Σ	1500	700	2200

	T	\bar{T}	Σ
Z	1775	300	2075
\bar{Z}	100	25	125
Σ	1875	325	2200

Tableau 4.5 – Exemples comparatifs

nous avons obtenu les valeurs ci-dessous.

	Pour la règle ($X \rightarrow Y$)	Pour la règle ($Z \rightarrow T$)
Conf($Z \rightarrow T$)	0, 92	0, 94
$M_{GK}(X \rightarrow Y)$	0, 75	0, 06
$\chi^2_{\text{Observé}}$	8,52	2, 87
Risque d'erreur (α)	5%	5%
$\chi^2_{\text{Théorique}}$	3, 84	3, 84

D'après ces deux exemples, les règles $X \rightarrow Y$ et $Z \rightarrow T$ ont toutes les deux une valeur très élevée de la mesure confiance (0, 92 et 0, 94). Si on se contentait d'utiliser seulement cette mesure, on pourrait croire que les deux règles étaient intéressantes ; pire encore, on pourrait même être amené à croire que la règle $Z \rightarrow T$ est plus significative par rapport à la règle $X \rightarrow Y$ (Conf($Z \rightarrow T$) > Conf($X \rightarrow Y$)). Pourtant, le test d'indépendance de χ^2 montre clairement qu'au risque d'erreur de moins de 5%, on doit accepter l'hypothèse selon laquelle les deux motifs Z et T étaient indépendants, autrement dit, il n'existe aucune liaison entre Z et T ($\chi^2_{\text{Observé}} \ll \chi^2_{\text{Théorique}}$). Ce simple contre-exemple montre qu'on ne doit pas se contenter seulement de la mesure *confiance* pour valider une règle ; après tout, la dite mesure *confiance* n'est rien d'autre que la probabilité conditionnelle sachant la prémisse de la variable du conséquent. Dans notre cas, chaque $\chi^2_{\text{Observé}}$ du contexte binaire \mathbb{K} (Tableau 4.2) est donnée dans B.11-TB.2.

Valeurs critiques de M_{GK}

En 2004, l'équipe de Totohasina a publié dans (Totohasina et al., 2004) un lien direct entre la mesure M_{GK} , alors appelée **Ion** à l'époque (**I**mplication statistique **O**rientée **N**ormée) et l'expression de χ^2 . Rappelons ici la proposition qui justifie cette relation (Totohasina, 2008).

Propriété 4.1 (Relation entre χ^2 et M_{GK}). *Pour tous motifs X et Y , on a :*

$$M_{GK}(X \rightarrow Y) = \begin{cases} \sqrt{\frac{1}{n} \frac{n-n_{v_i}}{n_{v_i}} \frac{n_{v_j}}{n-n_{v_j}}} \chi^2, & \text{si } X \text{ favorise } Y \text{ ou } X \text{ et } Y \text{ indépendants,} \\ -\sqrt{\frac{1}{n} \frac{n-n_{v_i}}{n_{v_i}} \frac{n_{v_j}}{n-n_{v_j}}} \chi^2, & \text{si } X \text{ défavorise } Y \text{ ou } X \text{ et } Y \text{ indépendants.} \end{cases} \quad (4.24)$$

Où n, n_{v_i}, n_{v_j} désignent respectivement la taille des données, l'effectif des individus contenant le motif X et l'effectif des individus contenant le motif Y (cf. Tableau 4.6).

Démonstration.

Considérons le tableau de contingence pour les deux motifs X et Y . Ici, $n_{v_i v_j}$ désigne l'effectif des individus contenant à la fois le motif X et Y , pour les notations. Sous l'hypothèse d'indépendance entre X et Y (hypothèse nulle) on devrait avoir le tableau de contingence ci-dessous (cf. Tableau 4.7).

	Y	\bar{Y}	Σ
X	$n_{v_i v_j}$	$n_{v_i \bar{v}_j}$	n_{v_i}
\bar{X}	$n_{\bar{v}_i v_j}$	$n_{\bar{v}_i \bar{v}_j}$	$n_{\bar{v}_i}$
Σ	n_{v_j}	$n_{\bar{v}_j}$	n

	Y	\bar{Y}	Σ
X	$\frac{n_{v_i} n_{v_j}}{n}$	$\frac{n_{v_i} n_{\bar{v}_j}}{n}$	n_{v_i}
\bar{X}	$\frac{n_{\bar{v}_i} n_{v_j}}{n}$	$\frac{n_{\bar{v}_i} n_{\bar{v}_j}}{n}$	$n_{\bar{v}_i}$
Σ	n_{v_j}	$n_{\bar{v}_j}$	n

Tableau 4.6 – Effectifs observés Tableau 4.7 – Effectifs théoriques

Dans ce cas, la valeur théorique de χ^2 s'écrit :

$$\chi_{\text{Observé}}^2 = \frac{\left(n_{v_i v_j} - \frac{n_{v_i} n_{v_j}}{n}\right)^2}{\frac{n_{v_i} n_{v_j}}{n}} + \frac{\left(n_{v_i \bar{v}_j} - \frac{n_{v_i} n_{\bar{v}_j}}{n}\right)^2}{\frac{n_{v_i} n_{\bar{v}_j}}{n}} + \frac{\left(n_{\bar{v}_i v_j} - \frac{n_{\bar{v}_i} n_{v_j}}{n}\right)^2}{\frac{n_{\bar{v}_i} n_{v_j}}{n}} + \frac{\left(n_{\bar{v}_i \bar{v}_j} - \frac{n_{\bar{v}_i} n_{\bar{v}_j}}{n}\right)^2}{\frac{n_{\bar{v}_i} n_{\bar{v}_j}}{n}}. \quad (4.25)$$

En multipliant l'expression (4.25) par $\frac{1}{n} \frac{n-n_{v_i}}{n_{v_i}} \frac{n_{v_j}}{n-n_{v_j}}$, et en prenant la racine carrée, on tombe sur l'expression de $M_{GK}(X \rightarrow Y)$ dans le cas de X favorisant Y .

D'autre part, si on multiplie l'expression (4.25) par $\frac{1}{n} \frac{n - n_{v_i}}{n_{v_i}} \frac{n - n_{v_j}}{n_{v_j}}$, et en prenant la racine carrée, on retrouve l'expression de $M_{GK}(X \rightarrow Y)$ dans le cas de X défavorisant Y . \square

Examinons maintenant le comportement de M_{GK} par rapport aux valeurs de χ^2 . Au risque d'erreur α (erreur de première espèce³), on peut rejeter l'hypothèse nulle lorsque :

$$\chi_{\text{Observé}}^2 > \chi_{\text{Théorique}}^2. \quad (4.26)$$

Or, rejeter l'hypothèse nulle signifie rejeter l'indépendance, d'une autre manière, accepter la dépendance. À partir de l'inégalité (4.26), on peut obtenir :

$$\sqrt{\frac{1}{n} \frac{n - n_{v_i}}{n_{v_i}} \frac{n_{v_j}}{n - n_{v_j}} \chi_{\text{Observé}}^2} > \sqrt{\frac{1}{n} \frac{n - n_{v_i}}{n_{v_i}} \frac{n_{v_j}}{n - n_{v_j}} \chi_{\text{Théorique}}^2}.$$

En effet, d'après (4.24), on a (dans le cas de X favorisant Y) :

$$M_{GK}(X \rightarrow Y) > \sqrt{\frac{1}{n} \frac{n - n_{v_i}}{n_{v_i}} \frac{n_{v_j}}{n - n_{v_j}} \chi_{\text{Théorique}}^2}. \quad (4.27)$$

Ici, $\chi_{\text{Théorique}}^2$ désigne la valeur théorique de χ^2 à 1 degré de liberté et à un risque d'erreur fixé par l'utilisateur. Remarquons qu'on peut se contenter de l'expression de M_{GK} dans le cas d'attraction mutuelle entre prémisses et conséquent. En effet, dans le cas de répulsion (X défavorisant Y par exemple), grâce à la propriété 4.2, on n'a qu'à s'intéresser à la règle $X \rightarrow \bar{Y}$) et utiliser l'expression de M_{GK} dans le cas d'attraction mutuelle. L'inégalité (4.27) nous amène aux définitions ci-dessous.

Définition 4.6 (Valeur critique). Soient X et Y deux motifs, n_{v_i} et n_{v_j} deux entiers tels que : $n_{v_i} = \text{Card}(X')$, $n_{v_j} = \text{Card}(Y')$. On appelle valeur critique de M_{GK} au seuil α , la quantité notée par M_{GK}^α et définie par :

$$M_{GK}^\alpha = \sqrt{\frac{1}{n} \frac{n - n_{v_i}}{n_{v_i}} \frac{n_{v_j}}{n - n_{v_j}} \chi_{\text{Théorique}}^2(\alpha)}. \quad (4.28)$$

Définition 4.7 (Règle valide). Soit X et Y deux motifs fréquents⁴ d'un contexte

3. Rejeter l'hypothèse nulle alors qu'elle est vraie.

4. Par rapport à un *minsup* préalablement établi.

binnaire d'extraction \mathbb{K} . On dit qu'une règle $X \rightarrow Y$ est valide au sens de la mesure M_{GK} , au risque d'erreur $100\alpha\%$ si et seulement si $M_{GK}(X \rightarrow Y) > M_{GK}^\alpha$.

Remarque sur l'utilisation de la valeur critique

Il faut souligner que si on veut avoir un résultat fiable au niveau de confiance plus de $100\alpha\%$ (risque d'erreur moins de $100(1-\alpha)\%$) lors de la prise de décision concernant la validité d'une règle $X \rightarrow Y$, on ne doit pas comparer la valeur de M_{GK} à la quantité $100\alpha\%$ comme c'est le cas dans la plupart des algorithmes d'extraction des règles d'association, en l'occurrence l'ancien algorithme d'extraction des bases des règles M_{GK} valides (Feno, 2007) mais on doit plutôt la comparer à la quantité $M_{GK}^{(1-\alpha)}$. En effet, l'utilisation de $M_{GK}^{(1-\alpha)}$ permet de fournir à la mesure M_{GK} une valeur précise qu'il faut dépasser pour assurer l'existence de dépendance significative entre X et Y selon le test statistique de χ^2 .

Pour illustrer cette affirmation, prenons l'exemple de la règle $X \rightarrow Y$ d'un contexte dressé dans le tableau 4.5. Supposons que l'utilisateur souhaite avoir une règle valide avec une précision⁵ plus grande que $\alpha = 0,95$. Si l'utilisateur compare $M_{GK}(X \rightarrow Y)$ à la valeur $0,95$, il risque de perdre la règle $X \rightarrow Y$. Pourtant avec la même précision (risque d'erreur de première espèce moins de $(1-\alpha) = 0,05$), le test d'indépendance de χ^2 valide la liaison entre X et Y . Ainsi, l'utilisateur ne doit pas comparer la valeur de M_{GK} à la probabilité $\alpha = 0,95$ mais plutôt à la quantité $M_{GK}^{(1-\alpha)}$ (valeur critique de M_{GK} au risque d'erreur $(1-\alpha)$). Pour notre exemple, $M_{GK}(X \rightarrow Y)$ est largement plus grand que $M_{GK}^{0,05}$ ($M_{GK}(X \rightarrow Y) = 0,75 \gg M_{GK}^{0,05} = 0,05088$). Alors, pour une précision de plus de $0,95$, la règle $X \rightarrow Y$ est valide au sens de la mesure M_{GK} . Nous allons utiliser cette expression de M_{GK}^α dans les nouveaux algorithmes d'extraction des règles M_{GK} valides.

Proposition 4.10. *Soit $\mathbb{K} = (\mathcal{I}, \mathcal{V}, \mathcal{R})$ un contexte d'extraction. Pour tous motifs X, Y de (\mathcal{I}) et pour tout $\alpha \in]0, 1[$, on a :*

- (1) *Si X favorise Y , alors $M_{GK}^\alpha(X \rightarrow Y) = M_{GK}^\alpha(\bar{Y} \rightarrow \bar{X})$,*
- (2) *Si X défavorise Y , alors $M_{GK}^\alpha(X \rightarrow \bar{Y}) = M_{GK}^\alpha(Y \rightarrow \bar{X})$.*

Démonstration.

5. Probabilité de ne pas se tromper

À un seuil d'erreur α , si X favorise Y , alors :

$$\begin{aligned} M_{GK}^\alpha(X \rightarrow Y) &= \sqrt{\frac{1}{n} \frac{n - n_X}{n_X} \frac{n_Y}{n - n_Y} \chi_\alpha^2} \\ &= \sqrt{\frac{1}{n} \frac{n_{\bar{X}}}{n - n_{\bar{X}}} \frac{n - n_{\bar{Y}}}{n_{\bar{Y}}} \chi_\alpha^2} \\ &= M_{GK}^\alpha(\bar{Y} \rightarrow \bar{X}). \end{aligned}$$

Par contre, si X défavorise Y , alors X favorise \bar{Y} et on peut appliquer la première propriété :

$$\begin{aligned} M_{GK}^\alpha(X \rightarrow \bar{Y}) &= M_{GK}^\alpha(\bar{\bar{Y}} \rightarrow \bar{X}) \\ &= M_{GK}^\alpha(Y \rightarrow \bar{X}). \end{aligned}$$

□

Corollaire 4.6. *Pour tout couple de motifs X, Y , si X favorise Y , alors les deux règles $X \rightarrow \bar{Y}$ et $\bar{Y} \rightarrow \bar{X}$ ont la même mesure selon M_{GK} et même valeur critique. Par conséquent, si les motifs \bar{X} et \bar{Y} sont fréquents, nous avons les équivalences, au sens de M_{GK} :*

$$\begin{aligned} X \rightarrow Y \text{ valide} &\Leftrightarrow \bar{Y} \rightarrow \bar{X} \text{ valide} , \\ X \rightarrow \bar{Y} \text{ valide} &\Leftrightarrow Y \rightarrow \bar{X} \text{ valide} . \end{aligned}$$

Selon le corollaire 4.6, on peut déduire la validité et la mesure de $\bar{Y} \rightarrow \bar{X}$ (respectivement de $Y \rightarrow \bar{X}$) à partir de celles de $X \rightarrow Y$ (respectivement de $X \rightarrow \bar{Y}$). Il est ainsi inutile de mettre $\bar{Y} \rightarrow \bar{X}$ ainsi que $Y \rightarrow \bar{X}$ lorsqu'on a déjà $X \rightarrow Y$ et $X \rightarrow \bar{Y}$. Contrairement aux anciennes bases des règles M_{GK} -valides, ces deux types de règles ($\bar{Y} \rightarrow \bar{X}$ et $Y \rightarrow \bar{X}$) ne feront pas partie des nouvelles bases des règles M_{GK} -valides. Cette restriction nous permet de réduire considérablement la taille des bases des règles tout en conservant les traces de toutes les informations valides.

Pour valider le lien implicatif entre deux variables, on utilise le test d'indépendance χ^2 pour un seuil α fixé par l'utilisateur. Par analogie, comme déjà démontré en 2004 par [Totohasina et al. \(2004\)](#), il existe un lien direct entre la mesure M_{GK} , appelée *ION* à l'époque (Implication statistique Orientée Normalisée) et χ^2 , appuyé et justi-

fié par [Totohasina et Feno \(2008\)](#) selon la propriété 4.1. Concernant la relation entre M_{GK} et χ^2 , dans notre cas, eu égard à ses propriétés mathématiques intéressantes et cohérentes à une interprétation en implication statistique, basée sur M_{GK}^f susmentionnées. Il est aisé de montrer que M_{GK}^f et la classique statistique de χ^2 à un degré de liberté sont reliés de telle sorte que la valeur critique de M_{GK}^f au seuil de risque α est

$$M_{GK(\alpha)}^f = \sqrt{\frac{1}{n} \frac{n - n_X}{n_X} \frac{n_Y}{n - n_Y} \chi_{\text{Théorique}(\alpha)}^2} \quad (4.29)$$

En tenant compte des valeurs de χ^2 théoriques, l'extraction d'une RA $X \rightarrow Y$ M_{GK} -valides étant conditionnées par l'inégalité : $M_{GK}^f(X \rightarrow Y) > M_{GK(\alpha)}^f$. Alors, nous montrons ci-dessous l'algorithme détaillé ([Rakotomalala et Totohasina, 2018](#)) (Cf. Algorithme 4) et l'algorithme correspondant présenté lors d'une communication orale à Lyon en 2017 (Cf. Figure 4.2) de l'extraction des RA M_{GK} -valides.

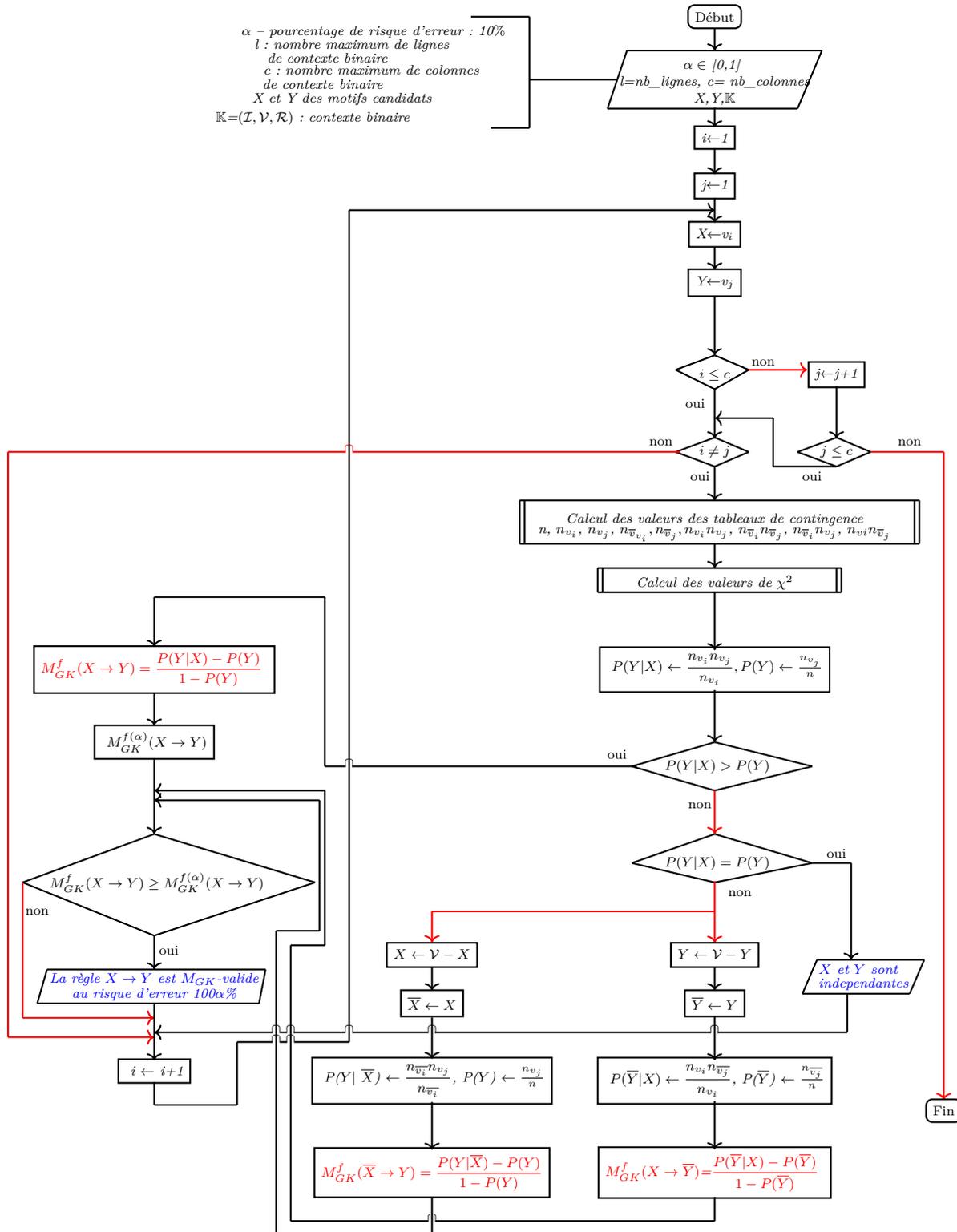


Figure 4.2 – Algorithme d'extraction des règles M_{GK} -valides

Algorithme 4 Algorithme d'extraction des règles d'association M_{GK} -valides

Entrée : $l=nb_lignes$, $c=nb_colonnes$, $\alpha \in [0, 1]$, X, Y

```

1 :  $i \leftarrow 1, j \leftarrow 1$ ;
2 : tant que  $i \leq c$  faire
3 :    $X \leftarrow v_i, Y \leftarrow v_j$ ;
4 :   tant que  $i \neq j$  faire
5 :     Calcul des valeurs des tableaux de contingence
6 :      $supp(v_i) \leftarrow \frac{n_{v_i}}{n}, supp(v_j) \leftarrow \frac{n_{v_j}}{n}$ ;
7 :     Calcul des valeurs de  $\chi^2$ 
8 :      $P(Y|X) \leftarrow \frac{n_{v_i n_{v_j}}}{n_{v_i}}, P(Y) \leftarrow \frac{n_{v_j}}{n}$ ;
9 :     si  $P(Y|X) > P(Y)$  alors
10 :       Aller à : 22
11 :     sinon
12 :       si  $P(Y|X) = P(Y)$  alors
13 :          $X$  et  $Y$  sont indépendantes
14 :         Aller à : 27
15 :       sinon
16 :          $X \leftarrow \mathcal{V} - X, \bar{X} \leftarrow X, P(Y|\bar{X}) \leftarrow \frac{n_{\bar{v}_i n_{v_j}}}{n_{\bar{v}_i}}, P(Y) = \frac{n_{v_j}}{n}$ ;
17 :          $M_{GK}^f(\bar{X} \rightarrow Y) \leftarrow \frac{P(Y|\bar{X}) - P(Y)}{1 - P(Y)}$ ;
18 :          $Y \leftarrow \mathcal{V} - Y, \bar{Y} \leftarrow Y, P(\bar{Y}|X) \leftarrow \frac{n_{v_i n_{\bar{v}_j}}}{n_{v_i}}, P(\bar{Y}) \leftarrow \frac{n_{\bar{v}_j}}{n}$ ;
19 :          $M_{GK}^f(X \rightarrow \bar{Y}) \leftarrow \frac{P(\bar{Y}|X) - P(\bar{Y})}{1 - P(\bar{Y})}$ ;
20 :         Aller à : 23
21 :       fin si
22 :        $M_{GK}^f(X \rightarrow Y) \leftarrow \frac{P(Y|X) - P(Y)}{1 - P(Y)}$ 
23 :        $M_{GK}^{f(\alpha)}(X \rightarrow Y) \leftarrow \sqrt{\frac{1}{n} \frac{n - n_{v_i}}{n_{v_i}} \frac{n_{v_j}}{n - n_{v_j}} \chi_{\text{Théorique}}^2(\alpha)}$ ;
24 :       si  $M_{GK}^f(X \rightarrow Y) \geq M_{GK}^{f(\alpha)}(X \rightarrow Y)$  alors
25 :         La règle  $X \rightarrow Y$  est  $M_{GK}$ -valides au risque d'erreur 100 $\alpha\%$ 
26 :          $supp_{M_{GK}}(X \rightarrow Y) \leftarrow supp(v_i)[(1 - supp(v_j))M_{GK}^f(X \rightarrow Y) + supp(v_j)]$ ;
27 :          $i \leftarrow i + 1$ ;
28 :         Aller à : 3
29 :       sinon
30 :         Aller à : 27
31 :       fin si
32 :     fin si
33 :   fin tant que
34 :    $j \leftarrow j + 1$ ;
35 :   si  $j \leq c$  alors
36 :     Aller à : 4
37 :   sinon
38 :     FIN
39 :   fin si
40 : fin tant que

```

Sortie : Les règles positives et négatives M_{GK} -valides et $supp_{M_{GK}}(X \rightarrow Y)$ respectives.

(Dans notre cas $\alpha=10\%$, soit $\chi^2 = 2,7$). En tenant compte des valeurs de χ^2 théoriques, et χ^2 observées, calculées à partir du logiciel développé par notre équipe sous Java (Cf. Listing B.1-Lignes[331-471], B.2-Lignes[11-33;114-122]), l'extraction d'une règle d'association $v_i \rightarrow v_j$ M_{GK} -valides étant conditionnées par l'inégalité : $M_{GK}^f(v_i \rightarrow v_j) > M_{GK(\alpha)}^f$, détaillée dans l'algorithme 4 et montrée par l'algorithme (Cf. Figure 4.2) (Rakotomalala, Totohasina, et Diatta, 2017a). En guise d'illustration, le tableau 4.8 ci-dessous tiré d'un contexte binaire formel (Cf. Tableau 4.2) nous montre que vingt deux règles d'association sont valides parmi les quatre vingt dix règles d'association extraites possibles, soit un taux de réduction de 24% environ.

M_{GK}	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1			0,206		0,647			0,853		
V2							0,684		0,426	
V3	0,538				1		1	1	0,727	
V4										
V5	0,169		0,1					0,2		
V6										0,571
V7		0,236	0,1							
V8	0,744		0,333		0,667					
V9		0,426	0,211							
V10						0,25				

Tableau 4.8 – Tableau représentant les valeurs de M_{GK} -valides de chaque couple de variables.

4.4.3 Apport par considération du *Support* de la règle d'association valide en fonction de M_{GK}

La relation entre le *Support* d'une règle et sa valeur M_{GK}

D'après la Définition 4.5-Equation 4.12, dans le cas où v_i favorise v_j , la composante favorisante

$$M_{GK}^f(v_i \rightarrow v_j) = \frac{conf(v_i \rightarrow v_j) - supp(v_j)}{1 - supp(v_j)}, \text{ avec } supp(v_j) \neq 1$$

$$\text{or } conf(v_i \rightarrow v_j) = \frac{supp(v_i \rightarrow v_j)}{supp(v_i)}, \text{ avec } supp(v_i) \neq 0$$

$$M_{GK}^f(v_i \rightarrow v_j) = \frac{\frac{supp(v_i \rightarrow v_j)}{supp(v_i)} - supp(v_j)}{1 - supp(v_j)}$$

Nous posons par définition :

Définition 4.8. (Rakotomalala et al., 2017a) *Soit une règle $v_i \rightarrow v_j$. On appelle support d'une règle d'association à base de M_{GK} , telle que v_i favorise v_j , et noté $supp_{M_{GK}}(v_i \rightarrow v_j)$, la quantité définie par :*

$$supp_{M_{GK}}(v_i \rightarrow v_j) = supp(v_i) [(1 - supp(v_j))M_{GK}(v_i \rightarrow v_j) + supp(v_j)] \quad (4.30)$$

soit

$$P(v_i \cap v_j) = P(v_i v_j) = P(v_i) [(1 - P(v_j))M_{GK}(v_i \rightarrow v_j) + P(v_j)] \quad (4.31)$$

Partant d'une règle $(v_i \rightarrow v_j)$, si la règle est constituée des motifs v_i et v_j très fréquents⁶, alors ces motifs sont partagés et les probabilités $P(v_i)$, $P(v_j)$ et $P(v_i v_j)$ sont fortes ; inversement, l'intérêt des connaissances impliquées par les motifs v_i et v_j , du point de vue de la découverte de la connaissance est faible. Si la règle est constituée de motifs v_i et v_j rares, alors ces motifs sont partagés par un petit nombre, mais apparaissent ensemble : ils sont alors vraisemblablement liés dans le contexte étudié. D'après l'équation (4.30), nous pouvons établir les valeurs de $supp_{M_{GK}}$ dans le tableau ci-dessous avec un seuil au risque d'erreur $\alpha = 10\%$.

$supp_{M_{GK}}$	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1			0,267		0,534			0,534		
V2							0,6		0,5	
V3	0,266				0,333		0,333	0,333	0,3	
V4										
V5	0,533		0,333					0,566		
V6										0,4
V7		0,599	0,333							
V8	0,533		0,333		0,567					
V9		0,5	0,3							
V10						0,4				

Tableau 4.9 – Tableau représentant les valeurs de $supp_{M_{GK}}$ de couples de variables.

Généralement, la mesure *Support* apparaît comme moteur de processus d'extraction, et qui écarte les règles ayant une valeur faible de *Support*, alors que certaines peuvent avoir une très forte *Confiance* et présenter un réel intérêt (les pépites de connais-

6. Motifs fréquents : c'est un ensemble d'éléments présents dans un nombre « suffisamment grand » de lignes d'une base de données. Le cadre mathématique qui suit est classique, et formalise celui de défini par R. Agrawal, T. Imielinski et A. Swami en 1993 et raffiné par Agrawal et R. Srikant l'année suivante

sances) (Azé et al., 2003). Nous constatons que les valeurs de $supp_{M_{GK}}$ figurées dans (Cf. Tableau 4.8) sont très faibles et nous risquons de perdre des règles intéressantes en appliquant l'équation (3.3) de la Définition 3.5, en ignorant les informations ou connaissances apportées par lesdites règles élaguées inconsciemment. Pour pallier à ce problème, nous nous référons à la normalisation de la mesure $supp_{M_{GK}}$ pour apporter un nouvel éclairage à celle-ci.

4.4.4 Normalisation de la mesure de qualité $supp_{M_{GK}}$

Dans Totohasina (2003), on a montré l'existence d'une mesure normée et centrée appelée ION à l'époque, qui n'est autre que M_{GK} , non symétrique et qui joue un rôle central par rapport aux indices probabilistes usuels, pour apprécier la qualité des règles d'association avec dépendance orientée interprétable en termes d'implication statistique; d'une autre manière, il s'agit de présenter une vue unificatrice des différentes mesures de qualité. L'objectif de la normalisation, c'est de ramener les valeurs de la mesure de la qualité $supp_{M_{GK}}$ sur l'intervalle $[-1, 1]$ tout en reflétant les **situations de référence** (Cf. Figure 4.3) telles que l'incompatibilité, la dépendance négative, l'indépendance, la dépendance positive et l'implication logique entre la prémisse et le conséquent d'une règle d'association.

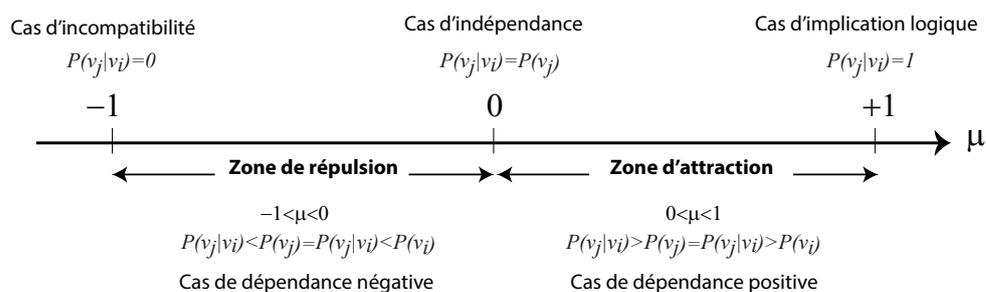


Figure 4.3 – Situations de référence

4.4.4.1 Intervalles homéomorphes

Définition 4.9. On appelle *intervalle* une partie connexe de \mathbb{R} non vide et non réduite à un point de l'une des formes suivantes : \mathbb{R} , $] - \infty, a[$, $] - \infty, a]$, $]a, b[$, $]a, b]$, $[a, b[$, $[a, b]$, $]a, +\infty[$, $]a, +\infty]$, avec $a < b$.

Définition 4.10. Deux intervalles I et J sont de même type s'ils sont homéomorphes, i.e. s'il existe deux applications $f : I \rightarrow J$ et $g : J \rightarrow I$, réciproques l'une de l'autre.

La relation d'homéomorphie est une relation d'équivalence sur les intervalles. Une condition nécessaire et suffisante pour que I et J soient homéomorphes, c'est qu'il existe $f : I \rightarrow J$ continue et bijective (il n'est pas évident que $g = f^{-1}$ soit continue, il faut pour cela montrer que f est strictement monotone), donc aussi g . Dans ce cas, I et J sont deux intervalles de **même type**.

Proposition 4.11. Il y a trois classes d'équivalence d'intervalles (types 1, 2 ou 3) pour la relation d'homéomorphie :

Type 1 Tous les intervalles compacts $[a, b]$,

Type 2 Tous les intervalles semi-ouverts de la forme $] - \infty, a]$, $]a, b]$, $[a, b[$, $[a, +\infty[$,

Type 3 Tous les intervalles de la forme \mathbb{R} , $] - \infty, a[$, $]a, b[$, $]a, +\infty[$.

Démonstration. On prouve les deux propositions suivantes :

- (i) que les intervalles d'une même classe sont homéomorphes ;
- (ii) que deux intervalles de classes différentes ne le sont pas.

Pour (i) : on présente des fonctions qui font le travail :

- $f(x) = a + (b - a)x$ permet de montrer que les intervalles $[0, 1]$ et $[a, b]$ sont homéomorphes, y compris pour les intervalles ouverts, ou semi-ouverts du même côté.
- Pour passer de $[0, 1[$ à $]0, 1]$, on utilise $f(x) = 1 - x$. Cela règle le cas de tous les intervalles bornés.
- On passe d'un intervalle avec une borne infinie à une autre de même type par une translation suivie de l'application $x \mapsto -x$.
- On passe de \mathbb{R} à $]0, 1[$ grâce à $f(x) = \frac{x}{1+|x|}$ (par exemple)
- De \mathbb{R} à $]0, +\infty[$ par l'exponentielle.

Pour (ii) : on utilise des arguments topologiques intrinsèques par rapport aux deux théorèmes : *l'image d'un compact (resp. connexe) par une application continue est compact (resp. connexe)*.

- L'argument de compacité permet de distinguer les intervalles compacts $[a, b]$ des autres.

- Pour distinguer les intervalles de types $[a, b[$ et $]c, d[$, on raisonne ainsi : si on a $f : I = [a, b[\rightarrow J =]c, d[$ continue et bijective, on regarde $I - a$, il est connexe, alors son image l'est aussi. Or, si $f(a) = e$ (avec $c < e < d$), on a $f(I - a) =]c, e[\cup]e, d[$ qui est non connexe et c'est absurde.

□

4.4.4.2 Normalisation de la mesure de qualité $supp_{M_{GK}}$

Définition 4.11. (Totohasina, 2003; Diatta et al., 2007). Soit $v_i \rightarrow v_j$ une règle d'association. Une mesure de qualité μ est dite normalisée, si elle vérifie les cinq conditions suivantes :

- (i) $\mu(v_i \rightarrow v_j) = -1$ si, et seulement si v_i et v_j sont incompatibles ;
- (ii) $-1 < \mu(v_i \rightarrow v_j) < 0$ si, et seulement si v_i défavorise v_j ou v_i et v_j sont négativement dépendants ;
- (iii) $\mu(v_i \rightarrow v_j) = 0$ si, et seulement si v_i et v_j sont indépendants ;
- (iv) $0 < \mu(v_i \rightarrow v_j) < 1$ si, et seulement si v_i favorise v_j ou v_i et v_j sont positivement dépendants ;
- (v) $\mu(v_i \rightarrow v_j) = 1$ si, et seulement si v_i implique logiquement v_j .

Processus de la normalisation

La mesure de qualité $supp_{M_{GK}}$ prend ses valeurs dans $[0, 1]$. Nous allons définir maintenant la mesure normalisée notée $supp_{(n)M_{GK}}$ définie dans $[-1, 1]$ associée à $supp_{M_{GK}}$ selon le processus de la normalisation (Totohasina, 2003; Totohasina et al., 2004; Totohasina, 2008).

Soit x_f (resp. y_f) le coefficient de multiplication (resp. de centrage) de $supp_{M_{GK}}$ dans le cas où v_i favorise v_j . De même, x_d (resp. y_d) dans le cas où v_i défavorise v_j . Désignons par $supp_{(n)M_{GK}}$ la mesure normalisée associée à $supp_{M_{GK}}$. On a alors,

$$supp_{(n)M_{GK}}(v_i \rightarrow v_j) = \begin{cases} x_f x + y_f, & \text{si } v_i \text{ favorise } v_j; \\ x_d x + y_d, & \text{si } v_i \text{ défavorise } v_j. \end{cases} \quad (4.32)$$

Les coefficients x_f , y_f , x_d et y_d se déterminent par passage aux limites dans les situations de référence (incompatibilité, indépendance et implication logique) du fait

de la continuité de l'évolution dans les deux zones : attraction (dépendance positive) et répulsion (dépendance négative). Posons $supp_{M_{GKimp}}(v_i \rightarrow v_j)$ la valeur de $supp_{M_{GK}}(v_i \rightarrow v_j)$ à l'implication, $supp_{M_{GKind}}(v_i \rightarrow v_j)$ celle de $supp_{M_{GK}}(v_i \rightarrow v_j)$ à l'indépendance et $supp_{M_{GKinc}}(v_i \rightarrow v_j)$ la valeur de $supp_{M_{GK}}(v_i \rightarrow v_j)$ à l'incompatibilité.

Au cas où v_i favorise v_j , on a :

$$\begin{cases} x_f supp_{M_{GKimp}}(v_i \rightarrow v_j) + y_f = 1, & \text{implication logique;} \\ x_f supp_{M_{GKind}}(v_i \rightarrow v_j) + y_f = 0, & \text{indépendance à droite.} \end{cases} \quad (4.33)$$

Au cas où, v_i défavorise v_j , on a :

$$\begin{cases} x_d supp_{M_{GKind}}(v_i \rightarrow v_j) + y_d = 0, & \text{indépendance à gauche;} \\ x_d supp_{M_{GKinc}}(v_i \rightarrow v_j) + y_d = -1, & \text{incompatibilité.} \end{cases} \quad (4.34)$$

Ainsi, nous pouvons écrire le système d'équations linéaires suivant :

$$\begin{cases} x_f supp_{M_{GKimp}}(v_i \rightarrow v_j) + y_f = 1 \\ x_f supp_{M_{GKind}}(v_i \rightarrow v_j) + y_f = 0 \\ x_d supp_{M_{GKind}}(v_i \rightarrow v_j) + y_d = 0 \\ x_d supp_{M_{GKinc}}(v_i \rightarrow v_j) + y_d = -1 \end{cases} \quad (4.35)$$

Et sous forme matricielle :

$$\begin{pmatrix} supp_{M_{GKimp}}(v_i \rightarrow v_j) & 1 & 0 & 0 \\ supp_{M_{GKind}}(v_i \rightarrow v_j) & 1 & 0 & 0 \\ 0 & 0 & supp_{M_{GKind}}(v_i \rightarrow v_j) & 1 \\ 0 & 0 & supp_{M_{GKinc}}(v_i \rightarrow v_j) & 1 \end{pmatrix} \begin{pmatrix} x_f \\ y_f \\ x_d \\ y_d \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix} \quad (4.36)$$

Posons M la matrice associée à ce système :

$$M = \begin{pmatrix} supp_{M_{GKimp}}(v_i \rightarrow v_j) & 1 & 0 & 0 \\ supp_{M_{GKind}}(v_i \rightarrow v_j) & 1 & 0 & 0 \\ 0 & 0 & supp_{M_{GKind}}(v_i \rightarrow v_j) & 1 \\ 0 & 0 & supp_{M_{GKinc}}(v_i \rightarrow v_j) & 1 \end{pmatrix} \quad (4.37)$$

Pour que la relation (4.36) ait une solution unique, il faut et il suffit que le déter-

minant de la matrice M définie par (4.37) soit fini et non nul.

Dans Totohasina et al. (2004),Feno (2007) un théorème assure que : une mesure probabiliste de qualité $supp_{M_{GK}}$ est normalisable si, et seulement si, pour toute règle d'association $v_i \rightarrow v_j$, les conditions suivantes sont vérifiées :

- (i) Les quantités $supp_{M_{GKimp}}(v_i \rightarrow v_j)$, $supp_{M_{GKind}}(v_i \rightarrow v_j)$ et $supp_{M_{GKinc}}(v_i \rightarrow v_j)$ sont finies ;
- (ii) Les inégalités suivantes sont vérifiées :
 $supp_{M_{GKimp}}(v_i \rightarrow v_j) \neq supp_{M_{GKind}}(v_i \rightarrow v_j)$
et $supp_{M_{GKind}}(v_i \rightarrow v_j) \neq supp_{M_{GKinc}}(v_i \rightarrow v_j)$.

Comme le déterminant de la matrice associée à l'équation (4.35) est

$$(supp_{M_{GKimp}}(v_i \rightarrow v_j) - supp_{M_{GKind}}(v_i \rightarrow v_j)) \times (supp_{M_{GKind}}(v_i \rightarrow v_j) - supp_{M_{GKinc}}(v_i \rightarrow v_j)),$$

le système d'équations (4.35) ne peut avoir une infinité de solution, car le second membre du système n'est pas nul. et une proposition qui en découle que pour une mesure de qualité normalisable ($supp_{M_{GK}}$) et une règle d'association $v_i \rightarrow v_j$. Les coefficients de multiplication et de centrage sont données par les expressions ci-dessous :

$$\begin{aligned} x_f &= \frac{1}{supp_{M_{GKimp}}(v_i \rightarrow v_j) - supp_{M_{GKind}}(v_i \rightarrow v_j)}, & y_f &= -\frac{supp_{M_{GKind}}(v_i \rightarrow v_j)}{supp_{M_{GKimp}}(v_i \rightarrow v_j) - supp_{M_{GKind}}(v_i \rightarrow v_j)}; \\ x_d &= \frac{1}{supp_{M_{GKind}}(v_i \rightarrow v_j) - supp_{M_{GKinc}}(v_i \rightarrow v_j)}, & y_d &= -\frac{supp_{M_{GKind}}(v_i \rightarrow v_j)}{supp_{M_{GKind}}(v_i \rightarrow v_j) - supp_{M_{GKinc}}(v_i \rightarrow v_j)}; \end{aligned}$$

En se référant aux équations (4.30) et (4.31), nous avons :

$$\begin{aligned} supp_{M_{GK}}(v_i \rightarrow v_j) &= P(v_i \cap v_j); & supp_{M_{GKinc}}(v_i \rightarrow v_j) &= 0; \\ supp_{M_{GKind}}(v_i \rightarrow v_j) &= P(v_i)P(v_j); & supp_{M_{GKimp}}(v_i \rightarrow v_j) &= P(v_i), \end{aligned}$$

alors $det(M) = P^2(v_i)P(\bar{v}_i)P(v_j)(\bar{v}_j) \neq 0$. (i) et (ii) sont vérifiées. La mesure $supp_{M_{GK}}$ est alors normalisable. D'où les coefficients de multiplication et de centrage sont :

$$\begin{aligned} x_f &= \frac{1}{P(v_i)(1-P(v_j))}, & y_f &= -\frac{P(v_i)P(v_j)}{P(v_i)(1-P(v_j))}; \\ x_d &= \frac{1}{P(v_i)P(v_j)}, & y_d &= -1; \end{aligned}$$

soit

$$supp_{(n)M_{GK}}(v_i \rightarrow v_j) = \begin{cases} \frac{P(v_i \cap v_j) - P(v_i)P(v_j)}{P(v_i)(1 - P(v_j))}, & \text{si } v_i \text{ favorise } v_j; \\ \frac{P(v_i \cap v_j) - P(v_i)P(v_j)}{P(v_i)P(v_j)}, & \text{si } v_i \text{ défavorise } v_j. \end{cases} \quad (4.38)$$

Montrons que $supp_{(n)M_{GK}}(v_i \rightarrow v_j) = M_{GK}(v_i \rightarrow v_j)$.

Ce qui nous intéresse est le cas favorisant.

Selon l'équation (4.38), on a :

$$supp_{(n)M_{GK}}(v_i \rightarrow v_j) = \begin{cases} \frac{supp_{M_{GK}}(v_i \rightarrow v_j) - P(v_i)P(v_j)}{P(v_i)(1 - P(v_j))}, & \text{si } v_i \text{ favorise } v_j; \\ \frac{supp_{M_{GK}}(v_i \rightarrow v_j) - P(v_i)P(v_j)}{P(v_i)P(v_j)}, & \text{si } v_i \text{ défavorise } v_j. \end{cases} \quad (4.39)$$

D'après le système d'équation (4.39), la composante favorisante de la normalisée du *support* basé sur M_{GK} est donnée par :

$$supp_{(n)M_{GK}}^f(v_i \rightarrow v_j) = \frac{supp_{M_{GK}}(v_i \rightarrow v_j) - P(v_i)P(v_j)}{P(v_i)(1 - P(v_j))}, \text{ si } v_i \text{ favorise } v_j. \quad (4.40)$$

Prenons l'équation (4.40), en multipliant le numérateur par $P(v_i)$, on a :

$$supp_{(n)M_{GK}}^f(v_i \rightarrow v_j) = \frac{P(v_i)(\frac{supp_{M_{GK}}(v_i \rightarrow v_j)}{P(v_i)} - P(v_j))}{P(v_i)(1 - P(v_j))}, \text{ si } v_i \text{ favorise } v_j. \quad (4.41)$$

Or $supp_{M_{GK}}(v_i \rightarrow v_j) = P(v_i \cap v_j)$ et $\frac{P(v_i \cap v_j)}{P(v_i)} = conf(v_i \rightarrow v_j)$, l'équation (4.40) devient :

$$supp_{(n)M_{GK}}^f(v_i \rightarrow v_j) = \frac{P(v_i)(conf(v_i \rightarrow v_j) - P(v_j))}{P(v_i)(1 - P(v_j))}, \text{ si } v_i \text{ favorise } v_j.$$

avec $P(v_i)(conf(v_i \rightarrow v_j) - P(v_j))$ n'est autre que la transformée affine de $conf(v_i \rightarrow v_j)$, et que l'on appelle aussi *confiance relative pondérée* par le support de la prémisse $P(v_i)$ de la règle qui rend parfois difficile l'apparition de règle ayant un faible *support* et une *confiance relative* ($conf(v_i \rightarrow v_j) - P(v_j)$) élevée.

Cependant, la mesure de qualité $supp_{(n)M_{GK}}$ est aussi une transformation affine de la *confiance*. On peut simplifier l'équation de $supp_{(n)M_{GK}}^f(v_i \rightarrow v_j)$ précédente par

$P(v_i)$, et on a :

$$supp_{(n)M_{GK}}^f(v_i \rightarrow v_j) = \frac{(conf(v_i \rightarrow v_j) - P(v_j))}{(1 - P(v_j))}, \text{ si } v_i \text{ favorise } v_j,$$

soit :

$$supp_{(n)M_{GK}}^f(v_i \rightarrow v_j) = \frac{(conf(v_i \rightarrow v_j) - supp(v_j))}{(1 - supp(v_j))}, \text{ si } v_i \text{ favorise } v_j.$$

D'après le système d'équation (4.13), on a :

$$supp_{(n)M_{GK}}^f(v_i \rightarrow v_j) = M_{GK}^f(v_i \rightarrow v_j)$$

Dans ce cas, nous obtenons le tableau ci-dessous représentant les valeurs de $supp_{(n)M_{GK}}$ -valides :

$supp_{(n)M_{GK}}$	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1			0,206		0,647			0,853		
V2							0,684		0,426	
V3	0,538				1		1	1	0,727	
V4										
V5	0,169		0,1					0,2		
V6										0,571
V7		0,236	0,1							
V8	0,744		0,333		0,667					
V9		0,426	0,211							
V10						0,25				

Tableau 4.10 – Tableau représentant $supp_{M_{GK}}$ -normalisées : $supp_{(n)M_{GK}}$.

A partir des tableaux (Tableau 4.9 et Tableau 4.10), nous pouvons construire le graphique représentant la variation des $supp_{M_{GK}}$ et $supp_{(n)M_{GK}}$ des règles valides (Cf. Figure 4.4).

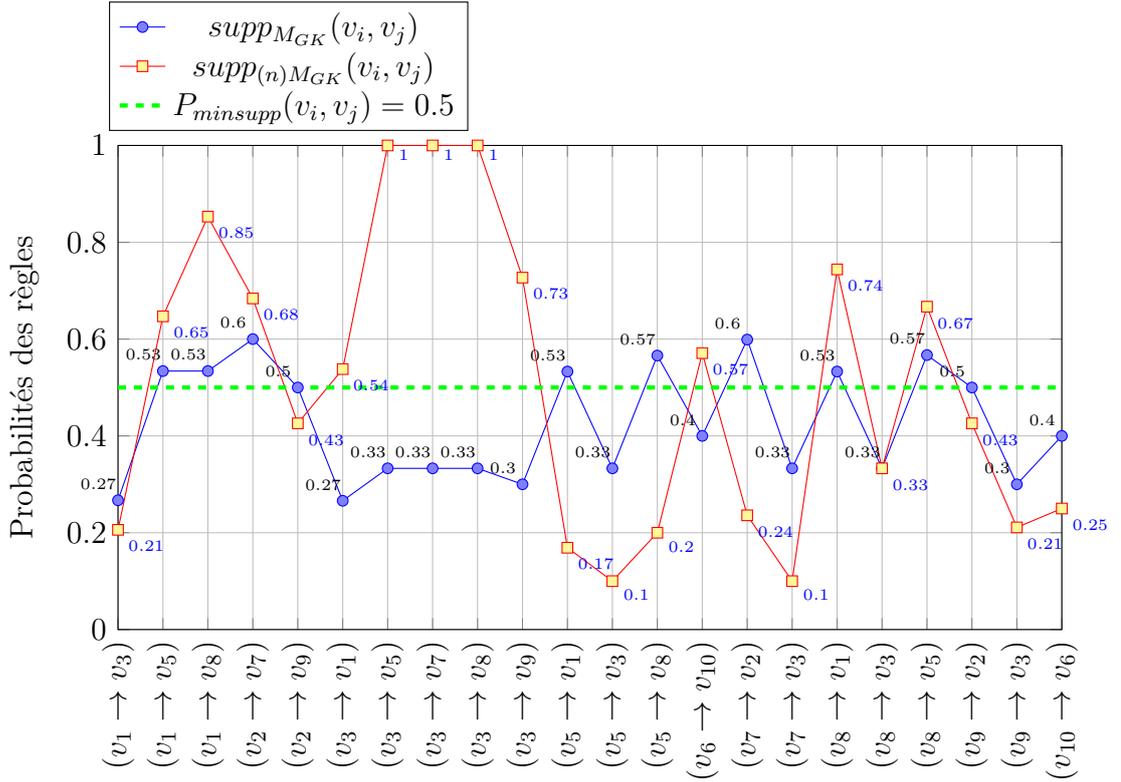


Figure 4.4 – Courbe représentative des $supp_{MGK}$ et $supp^{(n)}_{MGK}$ en fonction des couples de variables formant les règles valides à un seuil fixé au risque d'erreur $\alpha = 10\%$

Par rapport au contexte de fouille de données représenté dans (Tableau 4.2) et au risque d'erreur $\alpha = 10\%$ fixé par l'utilisateur, nous avons 22 règles valides positives (v_i favorise v_j) : $(v_1 \rightarrow v_3)$, $(v_1 \rightarrow v_5)$, $(v_1 \rightarrow v_8)$, $(v_2 \rightarrow v_7)$, ..., $(v_{10} \rightarrow v_6)$, portées sur l'axe horizontal, ainsi que leurs valeurs respectives mesurées par rapport au $supp_{MGK}$ et $supp^{(n)}_{MGK}$ portées sur l'axe vertical ($P \in [0, 1]$). Nous observons que les valeurs $supp_{MGK}(v_i \rightarrow v_j)$ -valides sont généralement faibles, car elles ne dépassent guère 0,6. Par contre, ses normalisées peuvent atteindre jusqu'à 1, malgré que la tendance de variation de ces deux mesures est presque la même. Les valeurs où $supp_{MGK}(v_i \rightarrow v_j) \geq sup^{(n)}_{MGK}(v_i \rightarrow v_j)$, sont données par les règles : $(v_1 \rightarrow v_3)$, $(v_2 \rightarrow v_9)$, $(v_5 \rightarrow v_1)$, $(v_5 \rightarrow v_3)$, $(v_5 \rightarrow v_8)$, $(v_7 \rightarrow v_2)$, $(v_7 \rightarrow v_3)$, $(v_9 \rightarrow v_2)$, $(v_9 \rightarrow v_3)$, $(v_{10} \rightarrow v_6)$; or la plupart de toutes ces règles ont des valeurs inférieures ou égales à 0,5 (voir seuil en ligne discontinue) et qui ne sont pas significatives. Par conséquent, la normalisation de la mesure $supp_{MGK}$ nous permet de renforcer (contraster) les valeurs des règles données par ladite mesure en conservant leur signification et leur portée. Cela nous

permet de faire émerger les pépites de connaissances. Une question se pose : A quoi ces augmentations des valeurs sont-elles dues ? Nous pouvons répondre à cela en présentant le même graphique avec les valeurs de confiance de chaque règle-valide ci-dessous (Cf. Figure 4.5) :

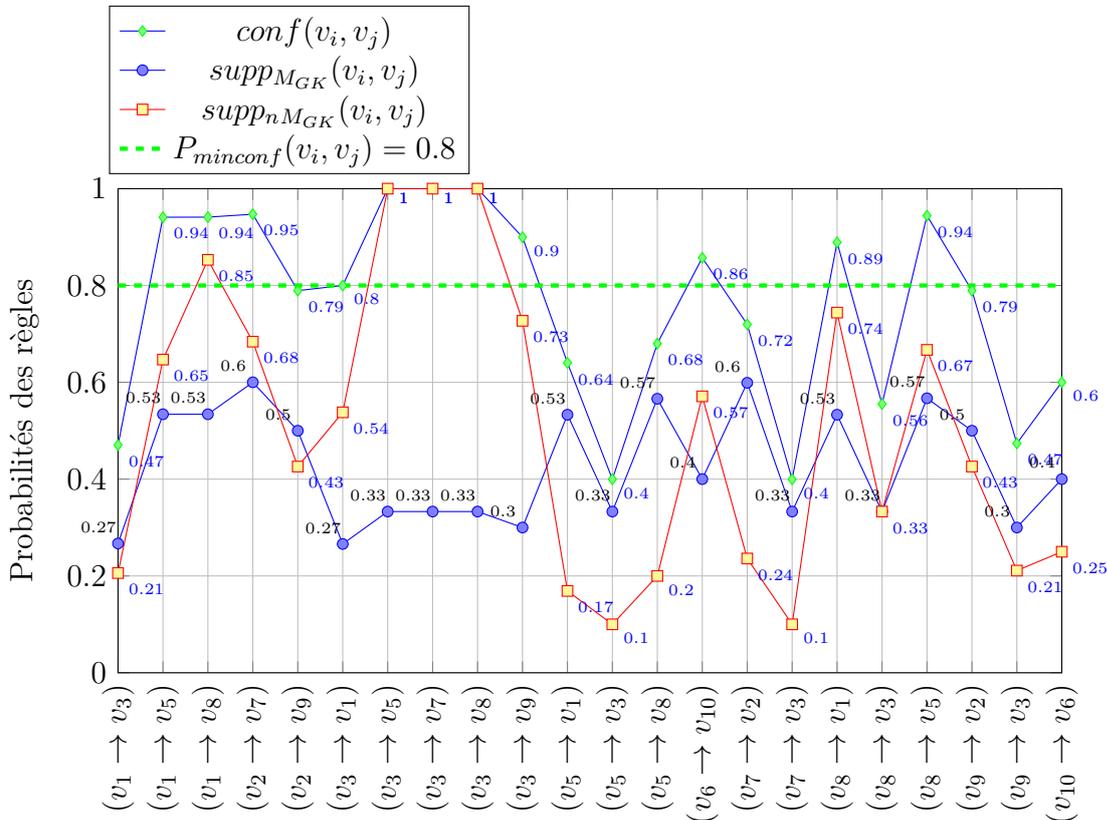


Figure 4.5 – Courbes représentatives des $supp_{M_{GK}}$ et $supp_{(n)M_{GK}}$ en fonction des couples de variables formant les règles valides à un seuil fixé au risque d'erreur $\alpha = 10\%$ comparées avec les confiances respectives

Ce graphique nous montre la variation des $supp_{M_{GK}}$, $supp_{(n)M_{GK}}$ et $conf$ des règles valides au risque d'erreur $\alpha = 10\%$ fixé par l'utilisateur. Sur l'axe horizontal, nous avons toujours les 22 règles valides positives (v_i favorise v_j) : $(v_1 \rightarrow v_3)$, $(v_1 \rightarrow v_5)$, $(v_1 \rightarrow v_8)$, $(v_2 \rightarrow v_7)$, ..., $(v_{10} \rightarrow v_6)$; sur l'axe vertical, les valeurs de $supp_{M_{GK}}$, $supp_{(n)M_{GK}}$ et $conf$ de chaque règle. Nous constatons que la variation et la tendance (montées-descentes) représentées par $supp_{(n)M_{GK}}$ et $conf$ sont très similaires (ligne avec point carré et losange) : cela veut dire qu'il y ait une influence non négligeable de la $conf$ de chacune de ces règles. Aussi, on peut remarquer :

$$\begin{cases} \text{supp}_{(n)M_{GK}}(v_i \rightarrow v_j) \geq \text{supp}_{M_{GK}}(v_i \rightarrow v_j), \text{ si } \text{conf}(v_i \rightarrow v_j) \geq 0.80; \\ \text{supp}_{(n)M_{GK}}(v_i \rightarrow v_j) < \text{supp}_{M_{GK}}(v_i \rightarrow v_j), \text{ sinon.} \end{cases}$$

On peut dire que la *confiance* qui n'est autre que la probabilité conditionnelle sachant la prémisse du conséquent, joue un rôle inévitable lors de la normalisation d'une mesure de règles d'association et doit prendre aussi son seuil minimum (notre cas $P_{\text{minconf}} = 0.80$ ligne discontinue). Ce résultat est, certes, empirique, mais il nous interroge sur son intérêt dans la recherche profonde des règles notoirement pertinentes. Reste à savoir, comment déterminer ce seuil de *confiance* pour toutes les mesures normalisables et qui sont égales à M_{GK} ?. Ce seuil dépend-il vraiment de la mesure M_{GK} -normalisables? Dans le cadre de ma thèse, ce qui m'intéresse c'est de construire une nouvelle mesure de cohésion à partir de la normalisée favorisante du *support* à base de M_{GK} ($\text{supp}_{(n)M_{GK}}$).

4.4.5 Vers un nouvel indice de cohésion implicative selon

M_{GK}

La cohésion est un indicateur d'ordre implicatif au sein d'une classe de variables. C'est une opposition au « désordre », elle nous mène à se référer à l'entropie d'une expérience aléatoire. Le mot « entropie » vient du grec entropê, qui signifie « retour ». Le concept d'entropie a été introduit en thermodynamique par Clausius. La croissance au cours du temps pour un système isolé de cette variable thermodynamique extensive est sensée exprimer l'irréversibilité, le « non-retour ». L'entropie est utilisée par Boltzmann en théorie cinétique des gaz et par Shannon (1948) en théorie de l'information apparaissant naturellement dans son article. L'entropie va permettre de capter le degré d'incertitude contenue dans une variable aléatoire. Une variable aléatoire X prenant toujours la même valeur n'a pas d'incertitude. Une telle variable aléatoire sera dite déterministe. Dans ce cas, l'entropie de X doit être égale à 0. A contrario, toute variable aléatoire qui n'est pas déterministe et alors de réalisation incertaine, doit présenter une entropie strictement plus grande que 0.

Considérons une variable aléatoire X prenant les valeurs distinctes x_1, \dots, x_n avec les probabilités P_1, \dots, P_n . On peut imaginer un texte écrit avec les n symboles de x_i , qui constituent l'alphabet utilisé. Nous désirons associer à chaque distribu-

tion (P_1, \dots, P_n) un nombre réel positif, noté $H^{(n)}(P_1, \dots, P_n)$, croissant avec le « désordre » et la « variabilité ». Si la variable aléatoire constante représente la certitude, $H^{(n)}$ apparaît comme une mesure d'incertitude, d'autant plus grande que la loi est « plus uniforme » sur son support. Alors, la quantité $H^{(n)}(X) = H^{(n)}(P_1, \dots, P_n)$ est nulle lorsque X est constante, et maximale (à n fixé) lorsque X suit la loi uniforme. Dans Shannon (1948), impose à $H^{(n)}$ les propriétés suivantes (Chafai, 2001):

Propriété 4.2. $H^{(n)}$ est une fonction positive, continue et symétrique en les variables P_i .

Propriété 4.3.

$$H^{(n)}\left(\frac{1}{n}, \dots, \frac{1}{n}\right) < H^{(n+1)}\left(\frac{1}{n+1}, \dots, \frac{1}{n+1}\right).$$

Propriété 4.4. Pour tout (b_1, \dots, b_k) tel que $b_1 + \dots + b_k = n$, on a :

$$H^{(n)}\left(\frac{1}{n}, \dots, \frac{1}{n}\right) = H^{(k)}\left(\frac{b_1}{n}, \dots, \frac{b_k}{n}\right) + \sum_{i=1}^k \frac{b_i}{n} H^{(k)}\left(\frac{1}{b_1}, \dots, \frac{1}{b_k}\right).$$

On omettra la référence à n dans la notation de $H^{(n)}$. La Propriété 4.3 ci-dessus n'est rien d'autre que la croissance du désordre avec la taille n du système. La troisième, elle correspond au partitionnement d'un système de taille n en k sous-systèmes de tailles b_i . Ces trois propriétés ne sont satisfaites que par les fonctions de la forme suivante :

$$H_b = (P_1, \dots, P_n) = - \sum_{i=1}^n P_i \log_b P_i = \sum_{i=1}^n P_i \log_b \frac{1}{P_i}. \quad (4.42)$$

Où \log_b désigne le logarithme de base $b > 0$, et $0 \log_b 0 = 0$, par définition. La fonction H_b est appelée entropie de base b : elle évalue la moyenne des incertitudes sur l'ensemble des états x_1, \dots, x_n . Elle est nulle pour la mesure de Dirac⁷ et maximale (par convexité) pour la loi uniforme, pour laquelle elle vaut $\log_b n$. A taille fixée n , elle est concave sur le simplexe (4.42) :

7. Une mesure de Dirac est une mesure supportée par un singleton et de masse unitaire

$$\left\{ (P_1, \dots, P_n); \forall i, P_i \geq 0, \sum_{i=1}^n P_i = 1 \right\}. \quad (4.43)$$

L'expérience aléatoire la plus simple qui soit comporte deux issues possibles (pour une seule issue, l'expérience n'est pas aléatoire) : c'est la loi de Bernoulli. Comme nous avons des variables binaires, les deux issues possibles sont $x_1 = 0$ et $x_2 = 1$, et apparaissent avec les probabilités respectives $P(x_1) = P$ et $P(x_2) = 1 - P$.

L'entropie s'écrit $H(X) = \lambda [-P \log(P) - (1 - P) \log(1 - P)]$.

Cette entropie est maximale lorsque les deux issues de l'expérience sont équiprobables, c'est-à-dire si $P = 1 - P = \frac{1}{2}$. Choisir λ , c'est choisir l'unité de mesure de l'incertitude. Le choix le plus couramment adopté attribue une incertitude de 1 bit à l'expérience aléatoire la plus simple qu'il soit. L'entropie de Bernoulli (entropie de Shannon d'un système à deux états) prend la forme ci-dessous :

$$H(X) = -P \log_2 P - (1 - P) \log_2(1 - P). \quad (4.44)$$

Et sa courbe représentative est montrée dans la figure ci-dessous (Cf. Figure 4.6) :

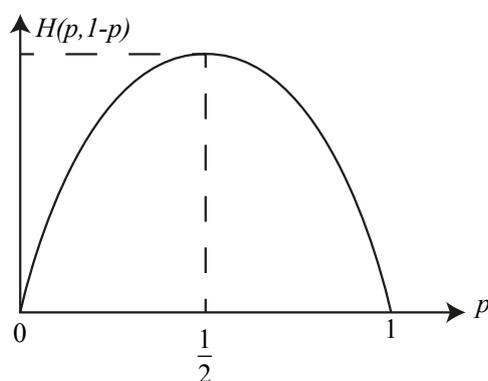


Figure 4.6 – Courbe d'entropie : cas de 2 variables formant une classe (v_i, v_j) .

Prenons le cas de deux variables qui forment une classe (v_i, v_j) . Soit X la variable aléatoire indicatrice de l'événement, pour chaque couple de variables (v_i, v_j) , P représente la *portée* ou la *fiabilité* de la règle formée par ledit couple :

$$Pr(X = 1) = \text{supp}_{M_{GK}}(v_i \rightarrow v_j) = P \text{ et } Pr(X = 0) = 1 - \text{supp}_{M_{GK}}(v_i \rightarrow v_j) = 1 - P.$$

L'entropie de cette expérience selon (4.44) est $H = -P \log_2 P - (1 - P) \log_2(1 - P)$.

Si $supp_{M_{GK}}(v_i \rightarrow v_j) = 1$, alors $H = 0$ en convenant $P \log_2 P = 0$.

Si $supp_{M_{GK}}(v_i \rightarrow v_j) = 0,5$, alors $H = 1$ (entropie maximale).

En posant $f(P) = -P \log_2 P - (1 - P) \log_2(1 - P)$, on a $f(P) = f(1 - P)$, l'entropie est symétrique par rapport à $P = 0,5$ et de plus $\frac{df}{dP} = \log_2 \frac{1-P}{P}$ pour $]0, 1[$. Alors H croît de 0 à 1 lorsque P croît de 0 à 0,5 et décroît de 1 à 0 lorsque P croît de 0,5 à 1 (Cf. Figure 4.6). Par analogie à la définition d'une cohésion implicative de deux variables de Ratsimba-Rajohn (1992), on choisit « le carré de l'entropie $supp_{M_{GK}}(v_i \rightarrow v_j)$ » pour renforcer un contraste dans $[0, 1]$ et « la racine carrée de son complément à 1 » pour donner à la cohésion la dimension de l'entropie et pour accroître sa valeur numérique (pour $x \in [0, 1], \sqrt{1 - x^2} \geq 1 - x$). Lorsque l'implication est stricte, on prend la cohésion de (v_i, v_j) égale à 1.

Définition 4.12. (Rakotomalala, Totohasina, et Diatta, 2017b) *L'indice de Cohésion du couple de variables (v_i, v_j) à base de $supp_{M_{GK}}$ tel que $v_i \leq v_j$, notée $coh_{supp_{M_{GK}}}$ est défini par :*

$$coh_{supp_{M_{GK}}}(v_i, v_j) = \begin{cases} \sqrt{1 - (supp_{M_{GK}}(v_i \rightarrow v_j))^2} & , \text{ si } supp_{M_{GK}}(v_i \rightarrow v_j) > 0,5 \\ 0 & , \text{ si } supp_{M_{GK}}(v_i \rightarrow v_j) \leq 0,5 \\ 1 & , \text{ si } supp_{M_{GK}}(v_i \rightarrow v_j) = 1 \end{cases} \quad (4.45)$$

En utilisant l'équation (4.45), nous pouvons établir le tableau 4.11 ci-dessous :

$coh_{supp_{M_{GK}}}$	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1			0		0,082			0,082		
V2							0,239		0	
V3	0				0		0	0	0	
V4										
V5	0,079		0					0,158		
V6										0
V7		0,237	0							
V8	0,079		0		0,161					
V9		0	0							
V10						0				

Tableau 4.11 – Tableau représentant les cohésions de couples de variables selon $supp_{M_{GK}}$

Nous constatons que parmi les 22 règles d'association valides, 14 sont éliminées, car elles sont toutes au-dessous du seuil 0.5 (Cf. Tableau 4.11) alors ($coh_{supp_{M_{GK}}} = 0$). D'autant plus, les valeurs des cohésions des classes constituées par deux variables

(v_i, v_j) restantes sont faibles. Passons maintenant à la cohésion selon la normalisée du support.

Définition 4.13. (Rakotomalala et al., 2017b) *L'indice de Cohésion du couple de variables (v_i, v_j) à base de la mesure normalisée, $supp_{(n)M_{GK}} \in [0.5, 1]$ tel que $v_i \leq v_j$, et notée $coh_{supp_{(n)M_{GK}}}$ est défini par :*

$$coh_{supp_{(n)M_{GK}}}(v_i, v_j) = \begin{cases} \sqrt{1 - (supp_{(n)M_{GK}}^f(v_i \rightarrow v_j))^2} & , \text{ si } supp_{(n)M_{GK}}^f(v_i \rightarrow v_j) > 0,5 \\ 0 & , \text{ si } supp_{(n)M_{GK}}^f(v_i \rightarrow v_j) \leq 0,5 \\ 1 & , \text{ si } supp_{(n)M_{GK}}^f(v_i \rightarrow v_j) = 1 \end{cases} \quad (4.46)$$

Proposition 4.12. *L'indice de cohésion à base de $supp_{(n)M_{GK}}$ est favorablement implicative :*

Si v_i favorise v_j , $coh_{supp_{(n)M_{GK}}}(v_i, v_j)$ est équivalent à sa contraposée, c'est-à-dire :

$$coh_{supp_{(n)M_{GK}}}(\bar{v}_j, \bar{v}_i) = coh_{supp_{(n)M_{GK}}}(v_i, v_j). \quad (4.47)$$

Démonstration. Nous avons déjà prouvé que : $supp_{(n)M_{GK}}^f(v_i \rightarrow v_j) = M_{GK}^f(v_i \rightarrow v_j)$. Alors, $supp_{(n)M_{GK}}^f(\bar{v}_j \rightarrow \bar{v}_i) = M_{GK}^f(\bar{v}_j \rightarrow \bar{v}_i)$.

$$coh_{supp_{(n)M_{GK}}}(\bar{v}_j, \bar{v}_i) = \sqrt{1 - (supp_{(n)M_{GK}}^f(\bar{v}_j \rightarrow \bar{v}_i))^2} = \sqrt{1 - (M_{GK}^f(\bar{v}_j \rightarrow \bar{v}_i))^2}$$

or dans Feno (2007), Totohasina (2003), Totohasina et Feno (2008) a démontré que

$M_{GK}^f(\bar{v}_j \rightarrow \bar{v}_i) = M_{GK}^f(v_i \rightarrow v_j)$, est favorablement implicative, on a :

$$coh_{supp_{(n)M_{GK}}}(\bar{v}_j, \bar{v}_i) = \sqrt{1 - (M_{GK}^f(v_i \rightarrow v_j))^2} = \sqrt{1 - (supp_{(n)M_{GK}}^f(v_i, v_j))^2}$$

Et cela nous montre aussi que :

$$coh_{supp_{(n)M_{GK}}}(\bar{v}_j, \bar{v}_i) = coh_{supp_{(n)M_{GK}}}(v_i, v_j).$$

□

Proposition 4.13. *L'indice de cohésion à base de $supp_{(n)M_{GK}}$ n'est pas symétrique : si v_i favorise v_j ,*

$$coh_{supp_{(n)M_{GK}}}(v_j, v_i) \neq coh_{supp_{(n)M_{GK}}}(v_i, v_j). \quad (4.48)$$

Démonstration. Comme M_{GK} n'est pas symétrique au cas où v_i favorise v_j , c'est-à-dire :

$$M_{GK}^f(v_j \rightarrow v_i) \neq M_{GK}^f(v_i \rightarrow v_j) \text{ alors } coh_{supp(n)M_{GK}}(v_j, v_i) \neq coh_{supp(n)M_{GK}}(v_i, v_j).$$

□

Avec l'équation (4.44), nous avons le tableau 4.12 ci-dessous :

$coh_{supp(n)M_{GK}}$	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1			0		0,35			0,798		
V2							0,436		0	
V3	0,091				1		1	1	0,534	
V4										
V5	0		0					0		
V6										0,17
V7		0	0							
V8	0,571		0		0,397					
V9		0	0							
V10						0				

Tableau 4.12 – Tableau représentant les cohésions de couples de variables selon $supp(n)M_{GK}$

A partir des tableaux (Tableau 4.11) et (Tableau 4.12), nous pouvons construire le graphique représentant la variation des $coh_{suppM_{GK}}$ et $coh_{supp(n)M_{GK}}$ des règles valides (Cf. Figure 4.7).

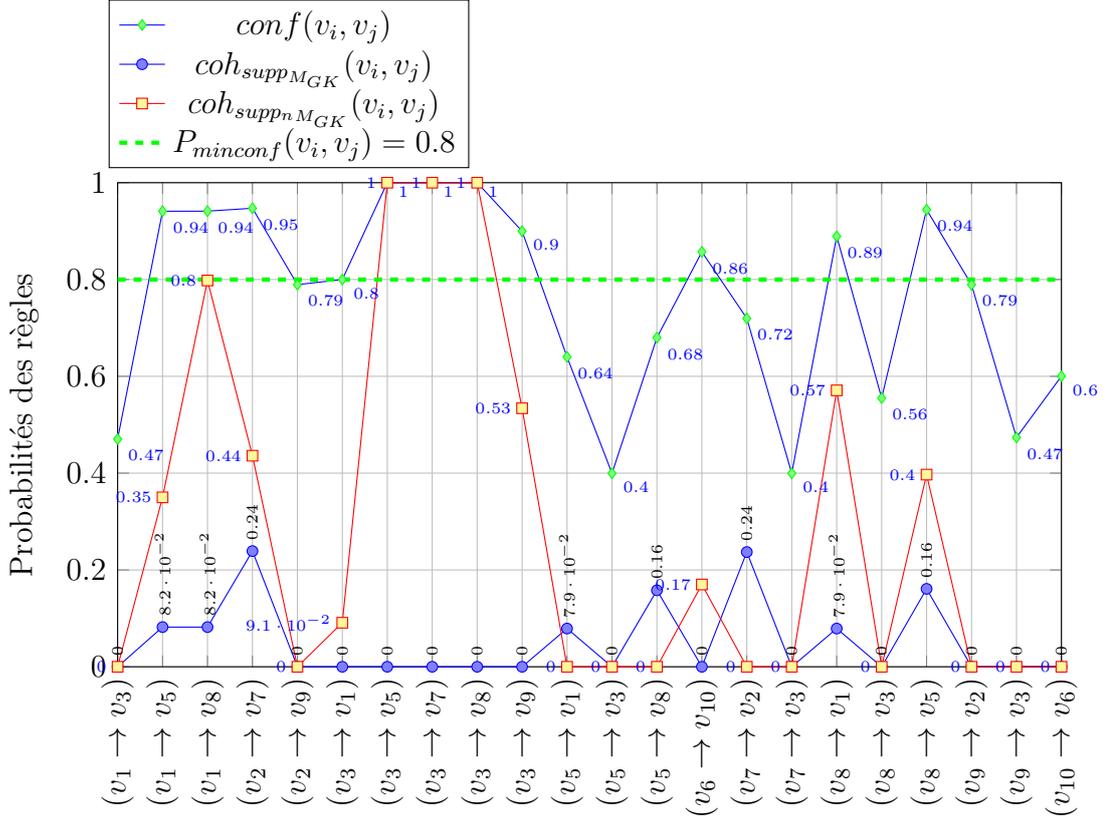


Figure 4.7 – Courbes représentatives des $coh_{supp_{MGK}}$ et $coh_{supp_{n_{MGK}}}$ et $conf$ des règles valides

Ce graphique nous montre la variation des $coh_{supp_{MGK}}$, $coh_{supp_{(n)MGK}}$ et $conf$ des règles valides à un seuil fixé au risque d'erreur $\alpha = 10\%$. Sur l'axe horizontal, nous avons toujours les 22 règles valides positives (v_i favorise v_j) : $(v_1 \rightarrow v_3)$, $(v_1 \rightarrow v_5)$, $(v_1 \rightarrow v_8)$, $(v_2 \rightarrow v_7)$, ..., $(v_{10} \rightarrow v_6)$; sur l'axe vertical, les valeurs de $coh_{supp_{MGK}}$, $coh_{supp_{(n)MGK}}$ et $conf$ de chaque règle d'association.

Nous constatons que la $coh_{supp_{MGK}}$ et $coh_{supp_{(n)MGK}}$ gardent les mêmes apparences de celles $supp_{MGK}$, et $supp_{(n)MGK}$ (comparer avec Figure 4.4). De plus, il y a quelques règles éliminées au cas où leurs confiances respectives sont inférieures à $P_{minconf} = 0.8$ (Cf. Figure 4.5) qui nous semble la clé majeure de la normalisation :

$$conf(v_5 \rightarrow v_1) = 0.64, \quad conf(v_7 \rightarrow v_2) = 0.719 \text{ alors } coh_{supp_{(n)MGK}}(v_5, v_1) = 0$$

$$\text{et } coh_{supp_{(n)MGK}}(v_7, v_2) = 0.$$

Par contre, il y en a d'autres qui apparaissent dans le cas où leurs confiances respectives sont supérieures à $P_{minconf} = 0.8$: $conf(v_3 \rightarrow v_1) = 1$, $conf(v_3 \rightarrow v_7) = 1$, $conf(v_3 \rightarrow v_8) = 1$, $conf(v_3 \rightarrow v_9) = 0.899$ et enfin $conf(v_6 \rightarrow v_{10}) = 0.857$

alors $coh_{supp(n)M_{GK}}(v_3, v_1) = 1$, $coh_{supp(n)M_{GK}}(v_3, v_7) = 1$, $coh_{supp(n)M_{GK}}(v_3, v_8) = 1$,
 $coh_{supp(n)M_{GK}}(v_3, v_9) = 0.534$,
 $coh_{supp(n)M_{GK}}(v_6, v_{10}) = 0.17$.

Nous pouvons en conclure que la *confiance* joue un double rôle, et non seulement en normalisation des mesures (faire apparaître les règles à forte confiance), mais aussi à l'évaluation des cohésions (filtre) des couples de variables données par ces dernières. Pour le même contexte de fouille de données traité par le logiciel CHIC de R. Gras, les valeurs des indices de cohésion selon la théorie classique par le modèle de Poisson est données par le tableau ci-dessous (Cf. Tableau 4.13) :

coh_{Gras}	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1			0.46		0.64	0	0	1.00	0.22	
V2					0	0.14	0.74	0	0.74	
V3	0.71				0.72	0	0.72	0.99	0.85	
V4										
V5	0.34		0.23					0.40		
V6		0.20								0.78
V7		0.46	0.23						0.16	
V8	0.99		0.78		0.69				0.35	0.13
V9	0.19	0.74	0.51				0.27	0.32		
V10						0.56		0.11		

Tableau 4.13 – Tableau représentant les valeurs de cohésion de couple de variables calculées par le logiciel CHIC de Gras.

D'après le (Tableau 4.13), les couples de variables (v_i, v_j) ayant leurs *Intensités d'Implication* respectives supérieures 0,5 sont au nombre de 32. On peut dire que le nombre de règles valides calculées à partir de l'*Intensité d'Implication* est supérieur au nombre de règles valides calculées à partir de $supp_{M_{GK}}$, ($nb II - valides > nb supp_{M_{GK}} - valides$).

Examinons maintenant le graphique ci-dessous montrant la variation des cohésions par rapport à l'*Intensité d'Implication* de R. Gras, les cohésions par rapport à $supp_{M_{GK}}$ et aussi par rapport à la normalisée du support $supp(n)M_{GK}$

La (Figure 4.8) nous montre que les valeurs de $coh_{Gras} \gg coh_{supp_{M_{GK}}} \approx$ très faible. Une raison de plus de prendre la normalisée de $supp_{M_{GK}}$. Les valeurs de $coh_{supp(n)M_{GK}}$ sont généralement supérieures aux celles de $coh_{supp_{M_{GK}}}$, malgré que, elles sont aussi généralement en dessous de coh_{Gras} , c'est-à-dire : $coh_{Gras} \geq coh_{supp(n)M_{GK}} \geq coh_{supp_{M_{GK}}}$ sauf sur des rares cas que nous explorons la(les) cause(s) prochainement. Mais, il nous paraît évident qu'au niveau de la quantité : le nombre de cohésions de couples extraites par la méthode de Gras est beaucoup plus nombreux que les cohésions cal-

culées à partir de $supp_{(n)M_{GK}}$, tandis que le nombre de cohésions extraites à partir de $supp_{(n)M_{GK}}$ est nombreux que celles calculées à partir de $supp_{M_{GK}}$, c'est-à-dire :

$$card(coh_{Gras}) \geq card(coh_{supp_{(n)M_{GK}}}) \geq card(coh_{supp_{M_{GK}}}). \quad (4.49)$$

Cette inégalité nous permet de dire que la $coh_{supp_{(n)M_{GK}}}$ prend le nombre en moyenne des règles extraites (ni trop laisser-aller, ni trop stricte). C'est pour cela qu'il nous paraît judicieux de considérer celle-ci comme "nouvel indice de cohésions, mais cette fois-ci à base de la mesure M_{GK} -normalisable".

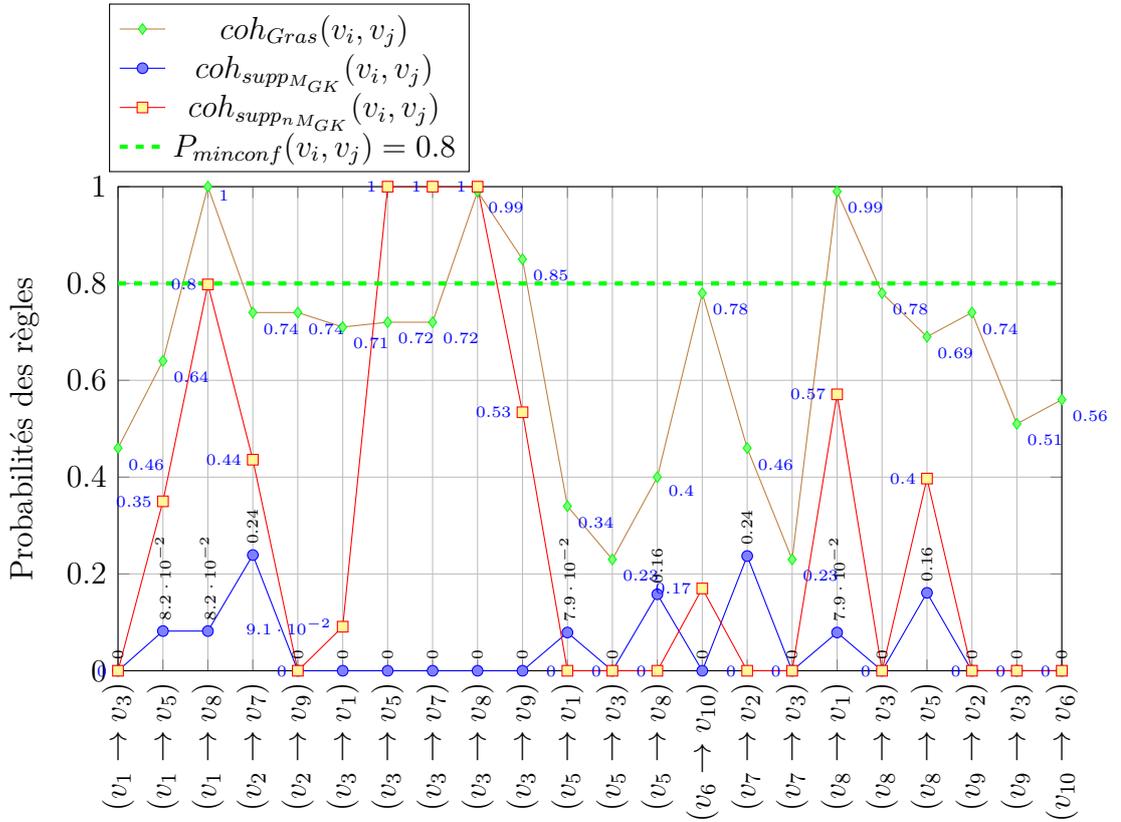


Figure 4.8 – Courbes représentatives des $coh_{supp_{M_{GK}}}$ et $coh_{supp_{n}M_{GK}}$ et $conf$ des règles valides

Définition 4.14. Soit une règle d'association $X \rightarrow Y$, telle que X favorise Y . Toute composante favorisant d'une mesure μ de qualité des règles d'associations M_{GK} -normalisable notée $\mu_{(n)M_{GK}}^f$ est un indice générique de cohésion de la règle.

4.5 Conclusion

Ce chapitre avait pour objectif de proposer un nouvel indice de cohésion à base de la mesure de qualité des règles d'association M_{GK} noté $coh_{supp(n)M_{GK}}$. D'abord, le processus d'extraction des M_{GK} -valides a été entrepris par rapport à un risque d'erreur fixé par nous-même à 10%, mais cela reste généralisable. Parallèlement à cela, on a pu mettre en évidence les supports des règles valides à base de M_{GK} notés $supp_{M_{GK}}$. Un constat nous paraît pertinent que le support des règles d'association à base de M_{GK} , vu leurs faibles valeurs risquent d'écarter des règles intéressantes. Pour remédier à cela, il a été proposé d'adopter la normalisée de celui-ci $supp(n)M_{GK}$ pour pouvoir faire apparaître des pépites de connaissances qui allait être forcément écartées. Il a été remarqué que la *confiance* qui n'est autre que la probabilité conditionnelle sachant la prémisse du conséquent, joue un rôle inévitable lors de la normalisation d'une mesure de règles d'association et prend son seuil minimum (*minconf*). Celui-ci n'est pas défini préalablement par l'utilisateur.

Ensuite, la normalisée du support à base de M_{GK} a été alors choisie pour quantifier la cohésion de couples de variables formant les règles, et il a été vu pareillement le rôle de la valeur minimale de la *confiance* (*minconf*) pour l'élagage des couples *de moins d'affinité*. Enfin, la normalisation du support des règles-valides à base de M_{GK} , ainsi que l'importance de la composante favorisante de cette mesure normalisée, nous a permis de construire l'indice de cohésion à base de M_{GK} ($coh_{supp(n)M_{GK}}$). Cet indice nouvellement créé a été comparé avec la cohésion calculée à partir de l'Intensité d'Implication de R. Gras (coh_{Gras}). Cette dernière nous paraît trop permissive ou trop poreuse. Mais malgré cela, ces deux mesures nous permettent d'engendrer des classes orientées de règles, de construire des méta-règles et enfin d'améliorer le filtre en classification hiérarchique implicative, que nous abordons dans le chapitre suivant.

Chapitre 5

Nouvelle Classification

Hiérarchique Implicative et

Cohésitive basée sur M_{GK}

*“ La hiérarchie spirituelle des
âges de la vie (enfance,
jeunesse, adulte et vieillesse) a
fait l’homme hiérarchique par
nature. ”*

Claude-may Waia Némia
Maître Reiki Kanak (1961)

Sommaire

5.1 Introduction	144
5.2 Construction statistique d’une hiérarchie implicative . .	144
5.2.1 Implication entre classes de variables	145
5.2.2 Construction de l’arbre hiérarchique et la détermination des niveaux significatifs des classifications	147
5.3 Contributions des individus	154
5.3.1 Paire orientée générique principale associée à une classe .	154
5.3.2 Vecteur puissance implicative d’une classe \underline{C}_H	155
5.3.3 Puissance implicative d’un individu et distance sur la classe \underline{C}_H	156
5.4 Conclusion	159

5.1 Introduction

L'objectif de la FD est de faire émerger par des méthodes algorithmiques des tendances à partir de grand volume de données pré-traitées. En FD, on parle souvent des règles d'association. Dans la vie quotidienne, comme les faits sont divers et nombreux, on peut aussi confronter à des nombreuses règles. Les travaux de [Totomasina \(2003\)](#); [Totomasina et al. \(2004\)](#) ont déjà proposé la théorie unificatrice des mesures de qualités normalisées, prenant ses valeurs dans l'intervalle $[-1, +1]$ et qui s'annulent en cas d'indépendance, valent $+1$ en cas d'implication logique et -1 en cas de situation d'incompatibilité : dont M_{GK} joue le rôle central et s'avérerait plus pertinente et donc meilleure pour extraire des associations orientées non symétriques répondant à des besoins de recherche de causes ou d'effets. Cette théorie règle déjà partiellement le problème majeur de la fouille des règles d'association dans un contexte binaire, à savoir la non-pertinence et la non-utilité de beaucoup de règles extraites et nous a permis d'élaborer l'algorithme d'extraction des règles d'association selon M_{GK} , et aussi de nous permettre de créer une nouvelle mesure de cohésion basée sur la mesure de qualité M_{GK} notée $coh_{supp(n)M_{GK}}$. Mais il est aussi plus astucieux de grouper les règles extraites en classes pour avoir des méta-règles et pour faciliter l'interprétation sans perdre les sens de la fouille. Dans ce chapitre, nous montrons le fondement et la méthode de la Classification Hiérarchique Implicative Cohésitive CHIC de [Gras, Couturier, et Bodin \(2003\)](#); [Ratsimba-Rajohn \(1992\)](#), basée sur l'Analyse Statistique Implicative ASI de [Gras et Régnier \(2009\)](#) avec la nouvelle indice de cohésion $coh_{supp(n)M_{GK}}$ cette fois-ci. Cela nous permet de compléter les fonctionnalités de notre outil logiciel, jusqu'à la représentation graphique de la connaissance sous forme de dendrogramme, afin de faciliter les interprétations.

5.2 Construction statistique d'une hiérarchie implicative

Le principe général de la construction d'une hiérarchie implicative binaire sur un ensemble \mathcal{V} est celui de la Classification Ascendante Hiérarchique (CAH) ([Lerman, 1981](#); [Nakache et Confais, 2005](#); [Lerman, 1983](#)). Soit un ensemble E d'entités, et supposons que E est muni d'un indice de proximité p , associant à chaque paire

$\{x, y\}$ d'éléments de E , un nombre positif $p(x, y)$ « mesurant » la proximité (ou la ressemblance) entre x et y . L'étape primordiale conceptuelle d'une CAH consiste à étendre la notion de proximité p entre éléments de E à celle \mathcal{P} entre parties disjointes de E . Ainsi, la construction ascendante hiérarchique d'un arbre de classification se déroule comme suit :

- Etat initial : Partition discrète dont chaque classe contient un seul élément, niveau :=0 ;
- Progression : Déterminer une nouvelle partition en fusionnant les paires de classes les plus proches au sens de \mathcal{P} , niveau := niveau+1 ;
- Règle d'arrêt : La partition formée est grossière et comporte une seule classe, niveau :=niveau+1.

L'arbre obtenu est un arbre à niveaux. Le niveau 0 est celui des feuilles de l'arbre et le dernier niveau atteint est celui de la racine de l'arbre. A chaque niveau se trouve associée une partition qui est moins fine que la partition du niveau précédent. L'arbre obtenu n'est pas nécessairement binaire. Il l'est si à chacun des niveaux la valeur maximale de l'indice de proximité \mathcal{P} est atteinte, soit sur une seule paire de classes du niveau, soit sur différentes paires de classes, mutuellement sans composante commune ; Si tel n'est pas le cas ([Lerman, 1989](#)), on peut toujours se ramener à une forme binaire de l'arbre des classifications et ce, en lui adjoignant une indexation. A chaque niveau, on attachera la valeur maximale de l'indice de proximité entre classes du niveau précédent. Dans ces conditions, une agrégation multiple sera décomposée en une suite d'agrégation binaires (dans n'importe quel ordre) où chacune donne naissance à un niveau ; mais avec le même valeur de l'indice de proximité. Dans notre cas, l'indice de proximité n'est autre que l'indice de cohésion à base de $supp^{(n)}M_{GK}$.

5.2.1 Implication entre classes de variables

Le regroupement des variables en classes semble nécessaire et qui apporte d'autres informations relatives à des structures sous-jacentes. [Lahrer \(1991\)](#) a étendu l'analyse implicative initialement à des implications entre classes des variables. Cette extension se base sur le nouveau concept de cohésion implicative. Pour former une classe, un sous-ensemble de variables devrait présenter un certain « *ordre implicatif* »

interne ». Ce qui signifie que si un groupe de variables est suffisamment désordonné du point de vue implicatif, il n'est pas légitime d'en constituer une classe dont on cherchait l'implication sur une autre classe ou qui serait impliquée par une autre classe.

La règle $v_i \Rightarrow v_j$ est entre deux attributs quelconques de \mathcal{V} par le couple (v_i, v_j) . Ce couple est appelé une classe de cardinal 2 et l'ensemble de tous les couples d'éléments distincts engendrés par les attributs de $V \subseteq \mathcal{V}$ est noté \underline{C}_2 .

$$\underline{C}_2 = \{(v_i, v_j), (v_j, v_i)\}$$

Considérons \underline{C}_3 des classes de cardinal 3. Chaque classe est un couple dont l'une des composantes soit un attribut et l'autre composante un élément de \underline{C}_2 .

$$\begin{aligned} \text{Si } V = \{v_i, v_j, v_k\} \text{ alors } \underline{C}_2 &= \{(v_i, v_j), (v_j, v_i), (v_i, v_k), (v_k, v_i), (v_j, v_k), (v_k, v_j)\} \text{ et} \\ \underline{C}_3 &= \{(v_i, (v_j, v_k)), ((v_i, v_j), v_k), (v_i, (v_k, v_j)), ((v_i, v_k), v_j), (v_j, (v_i, v_k)), ((v_j, v_i), v_k), \\ &(v_j, (v_k, v_i)), ((v_j, v_k), v_i), (v_k, (v_i, v_j)), ((v_k, v_i), v_j), (v_k, (v_j, v_i)), ((v_k, v_j), v_i)\} \end{aligned}$$

Définition 5.1. $\underline{C}_i = \underline{V}$. L'ensemble $\underline{C}_i, 2 \leq i \leq n$ où n est le cardinal de V ($\text{Card}(V) = n$) des classes de cardinal i est composé de couples dont les composantes distinctes appartiennent à $\underline{C}_j, 1 \leq j \leq i$ et tels que le nombre des attributs total des deux composantes est égal à i , on note $\underline{C} = \underline{C}_2 \cup \underline{C}_3 \cup \dots \cup \underline{C}_n$ le nombre de toutes les classes engendrées à partir de V et $\underline{C} = (\underline{V}, \underline{W})$ une classe quelconque de \underline{C} .

Définition 5.2. On appelle ordre implicatif $>_I$ sur l'ensemble des attributs de \underline{C} l'ordre défini par la lecture de gauche à droite de la relation d'implication.

Exemple 5.2.1. si $V = \{v_i, v_j, v_k, v_l\}$, les classes $((v_i, v_j), (v_k, v_l))$ et $(v_i, (v_j, (v_k, v_l)))$ sont des éléments de \underline{C}_4 désignant respectivement les règle $(v_i \Rightarrow v_j) \Rightarrow (v_k \Rightarrow v_l)$ et $(v_i \Rightarrow (v_j \Rightarrow (v_k \Rightarrow v_l)))$ ont le même ordre implicatif $v_i >_I, v_j >_I, v_k >_I, v_l >_I$.

Objectif: mettre en évidence les classes \underline{C} dont l'implication entre les composantes est significative au sens d'un certain critère de cohésion. La cohésion d'une classe à deux éléments est une fonction croissante de l'implication entre ces éléments.

D'une façon générale, pour un ensemble r -variables, et pour $V = \{v_i, v_j\}$, $i, j \in \{1, \dots, r\}$, [Lahrer \(1991\)](#) définit la cohésion de l'ensemble V formé la classe \underline{C}_2 comme moyenne géométrique des cohésions de chaque couple (v_i, v_j) pour lequel $n_{v_i} \leq n_{v_j}$.

$$coh_{supp(n)M_{GK}}(\underline{C}_2) = \prod_{\substack{i=1, \dots, r-1 \\ j=2, \dots, r \\ j>i}} coh_{supp(n)M_{GK}}(v_i, v_j)^{\left(\frac{2}{r(r-1)}\right)} \quad (5.1)$$

Définition 5.3. Soit \underline{C} une classe quelconque avec comme attributs dans l'ordre implicatif défini dans la classe $\underline{C}_1, \underline{C}_2, \dots, \underline{C}_k$, l'indice de cohésion ($coh_{supp(n)M_{GK}}$) de degré k est défini par :

$$coh_{supp(n)M_{GK}}(\underline{C}) = \left[\prod_{\substack{i=1, \dots, k-1 \\ j=2, \dots, k \\ j>i}} coh_{supp(n)M_{GK}}(\underline{C}_i, \underline{C}_j) \right]^{\frac{2}{k(k-1)}} \quad (5.2)$$

La cohésion implicative du groupe est nulle dès qu'un des couples a une cohésion nulle, c'est-à-dire avoir une valeur de $supp(n)M_{GK} = 0, 5$

Définition 5.4. Soient deux classes \underline{C} dans l'ordre implicatif défini par la classe $\underline{C}_1, \underline{C}_2, \dots, \underline{C}_r$ et \underline{C}' dans l'ordre implicatif défini par la classe $\underline{C}'_1, \underline{C}'_2, \dots, \underline{C}'_s$ avec des cohésions respectives $coh_{supp(n)M_{GK}}(\underline{C})$ et $coh_{supp(n)M_{GK}}(\underline{C}')$. L'implication entre les classes \underline{C} et \underline{C}' est définie par :

$$\psi(\underline{C}, \underline{C}') = \left[\max_{\substack{i \in \{1, \dots, r\} \\ j \in \{1, \dots, s\}}} (supp(n)M_{GK}(\underline{C}_i \rightarrow \underline{C}'_j)) \right]^{rs} \times \left[coh_{supp(n)M_{GK}}(\underline{C}) \times coh_{supp(n)M_{GK}}(\underline{C}') \right]^{\frac{1}{2}} \quad (5.3)$$

L'exposant rs permet de diminuer l'effet des classes à grands effectifs d'attributs.

Cet indice d'implication entre classes de variables

- croît avec les cohésions de chaque classe,
- s'annule dès qu'une cohésion est nulle,
- croît avec la liaison extrême existante,
- décroît avec les cardinaux des classes.

5.2.2 Construction de l'arbre hiérarchique et la détermination des niveaux significatifs des classifications

L'arbre de classification (dendrogramme) est construit pas à pas. Plusieurs variables sont réunies et forment une classe à chaque pas d'agrégation. Cette classe doit avoir le maximum de cohésion implicative parmi toutes celles qui auraient pu se former à ce niveau d'agrégation. L'ensemble des classes constituées respecte l'« ordre » induit

entre des variables réunies par $supp_{(n)M_{GK}}$. Le principe est de réunir à chaque pas d'agrégation le couple de variables présentant le maximum de cohésion à l'étape considérée.

5.2.2.1 Construction de l'arbre hiérarchique

La construction de l'arbre hiérarchique d'implication se fait par agrégation successive de classes d'implications (Gras, Kuntz, et Briand, 2003; Gras, 2013). Le principe est de réunir à chaque pas d'agrégation le couple de variables présentant le maximum de cohésion à l'étape considérée.

Un algorithme, basé sur le critère de la cohésion de classe, permet d'élaborer un arbre hiérarchique ascendant. Ainsi, au premier niveau, se réunissent , en une classe orientée par l'implication maximale, deux variables formant une classe dont la cohésion est maximale parmi toutes les cohésions des autres classes à deux termes. Au niveau suivant, on trouvera une classe à deux termes ou bien une classe à trois termes dont deux d'entre eux ont été réunis au niveau précédent, etc. En cas d'ex-aequo, la plus faible cardinalité définit le critère de priorité. L'élargissement d'une classe, contrairement aux hiérarchies de similarité, s'arrête dès que toute extension de cette classe conduit à une cohésion nulle.

La construction de la classe de niveau $h > 2$. On note $H_{h-1} = \{\underline{C}_1, \dots, \underline{C}_{h-1}\}$ l'ensemble des classes construites aux niveaux précédents et k le cardinal maximal de ces classes. Nous comparons les cohésions suivantes qui correspondent aux classes de cardinal 2, puis le cardinal 3 jusqu'à $2k$ dont au moins une des composantes est une classe \underline{C}_i construite précédemment :

- pour les classes de cardinal 2 : $coh_{supp_{(n)M_{GK}}}(\underline{C}_i)$ pour $\underline{C}_i \in \underline{C}_2 - H_{h-1} \cap \underline{C}_2$
- pour les classes de cardinal 3 : $coh_{supp_{(n)M_{GK}}}(v_*, \underline{C}_i)$ et $coh_{supp_{(n)M_{GK}}}(\underline{C}_i, v_*)$
pour $\underline{C}_i \in H_{h-1} \cap \underline{C}_2$ et v_* attributs de \mathcal{V} absent dans \underline{C}_i
- pour les classes de cardinal 4 : $coh_{supp_{(n)M_{GK}}}(\underline{C}_i, \underline{C}_j)$ avec \underline{C}_i et $\underline{C}_j \in H_{h-1} \cap \underline{C}_2$;
 $coh_{supp_{(n)M_{GK}}}(v_*, \underline{C}_i)$ et $coh_{supp_{(n)M_{GK}}}(\underline{C}_i, v_*)$
pour $\underline{C}_i \in H_{h-1} \cap \underline{C}_3$ et v_* attributs de \mathcal{V} absent dans \underline{C}_i
- pour les classes de cardinal $k' \leq 2k$: $coh_{supp_{(n)M_{GK}}}(v_*, \underline{C}_i)$ et $coh_{supp_{(n)M_{GK}}}(\underline{C}_i, v_*)$
pour $\underline{C}_i \in H_{h-1} \cap \underline{C}_{k'-1}$ et v_* attributs de \mathcal{V} absent dans \underline{C}_i ;
et $coh_{supp_{(n)M_{GK}}}(\underline{C}_i, \underline{C}_j)$ où \underline{C}_i et $\underline{C}_j \in H_{h-1}$ et sont telles que le cardinal de \underline{C}_i plus le cardinal de \underline{C}_j vaut k'

La classe retenue au niveau h est celle dont la cohésion est maximale et non nulle parmi l'ensemble des classes décrites ci-dessus. En cas d'ex-aequo, si deux classes $(\underline{C}_i, \underline{C}_j)$ et $(\underline{C}'_i, \underline{C}'_j)$ admettent la même cohésion au niveau h , on retient celle qui réalise une liaison implicative maximale entre les attributs respectifs des deux dernières composantes $(\underline{C}_i$ et \underline{C}_j ou \underline{C}'_i et $\underline{C}'_j)$ déjà constituées au niveau $h-1$ antérieur.

Notations :

Considérons \mathcal{I} l'ensemble des individus observés et \mathcal{V} l'ensemble de s caractères étudiés et qui sont notés $v_{i,i \in S}$ avec $S = \{1, 2, \dots, s\}$.

Pour l'analyse implicative, nous prenons les variables $v_{i,i \in S}$ deux à deux, il y en a alors $C_s^2 = \frac{s(s-1)}{2}$ paires orientés.

Nous parlons de paire mais pas de couple car à une paire notée $((v_i, v_j))$ correspondent deux couples (v_i, v_j) et (v_j, v_i) . Comme on ne prend que ce qui respecte la condition dissymétrique $n_{v_i} < n_{v_j}$, la notion de paire orienté et celle de couple se confondent au cas où l'un d'entre eux vérifie $n_{v_i} = n_{v_j}$. L'expression « **classe** » est employée dans le même sens que celui de « **paire orientée** » (Lahrer, 1991). Nous adjoignons chaque paire $((v_i, v_j))$ sa valeur $supp_{(n)M_{GK}}(v_i \rightarrow v_j)$ et sa cohésion $coh_{supp_{(n)M_{GK}}}(v_i, v_j)$. Nous nous intéressons à l'ensemble de triplets suivants :

$$\left\{ \begin{array}{l} \{((v_i, v_j), n_{v_i} \leq n_{v_j}, supp_{(n)M_{GK}}(v_i \rightarrow v_j))\} \text{ avec } v_i \in \mathcal{V}, v_j \in \mathcal{V}, i \in S, j \in S, i \neq j \\ \text{et} \\ \{((v_i, v_j), n_{v_i} \leq n_{v_j}, coh_{supp_{(n)M_{GK}}}(v_i, v_j))\} \text{ avec } v_i \in \mathcal{V}, v_j \in \mathcal{V}, i \in S, j \in S, i \neq j \end{array} \right. \quad (5.4)$$

Comme l'ordre des cohésions $coh_{supp_{(n)M_{GK}}}$ se conserve sur l'ordre des $supp_{(n)M_{GK}}$, alors on peut travailler sur le deuxième triplet $((v_i, v_j), n_{v_i} \leq n_{v_j}, coh_{supp_{(n)M_{GK}}}(v_i, v_j))$ désignant une paire orientée $((v_i, v_j))$ et l'ensemble des paires orientées $((v_i, v_j))$ est noté PO et le nombre des paires orientées est $Card(PO) \leq C_s^2$, car nous ne considérons que les règles valides, avec s est le nombre total des variables $s = Card(S)$. Tirés du tableau 4.2 de contexte binaire dans (§4.4.1), nous avons la table de l'occurrence (Tableau 5.1), pour dire à titre d'exemples que la variable v_1 apparaît dans 17 individus sur les 30 enregistrés.

Variable	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Occurrence	17	19	10	20	25	14	25	18	19	20

Tableau 5.1 – Tableau représentant l'occurrence de chaque variable

5.2.2.2 Recherche des niveaux significatifs

Etant donné la multiplicité liée directement à celle des variables des niveaux de formation des classes, il est nécessaire de dégager ceux qui sont les plus pertinents par rapport à l'intention classificatrice de l'utilisateur et eu égard aux critères choisis. Soit (PO, \leq) tel que :

$$((v_i, v_j)) \leq ((v_k, v_l)) \iff coh_{supp(n)M_{GK}}((v_i, v_j)) \leq coh_{supp(n)M_{GK}}((v_k, v_l)) \quad (5.5)$$

Définition 5.5. On appelle préordre initial et global cohésitif ou (préordonnance) sur $PO \times PO$, le préordre noté Ω induit par l'application cohésion $coh_{supp(n)M_{GK}}$ sur $PO \times PO$ et défini par la relation :

$$((v_{\tau(1)}, v_{\tau'(1)})) \leq ((v_{\tau(2)}, v_{\tau'(2)})) \leq \dots \leq ((v_{\tau(s)}, v_{\tau'(s)})) \quad (5.6)$$

où τ et τ' sont deux applications de $\{1, \dots, Card(PO)\}$ vers S et la graphe de cette relation de préordre \leq est noté $gr(\Omega)$ et $gr(\Omega) \subseteq PO \times PO$.

Tirés du tableau 4.12 dans (§4.4.5), nous avons l'ensemble des paires orientées suivant :

$$PO = \{((v_1, v_5)), ((v_1, v_8)), ((v_2, v_7)), ((v_3, v_1)), ((v_3, v_5)), ((v_3, v_7)), ((v_3, v_8)), ((v_3, v_9)), ((v_6, v_{10})), ((v_8, v_5))\}.$$

En triant ces PO par ordre croissant, nous avons le tableau 5.2 pour former le préordre initial et global cohésitif Ω .

PO	Cohesion
$((V3, V5))$	1.0
$((V3, V7))$	1.0
$((V3, V8))$	1.0
$((V1, V8))$	0.798
$((V3, V9))$	0.534
$((V2, V7))$	0.436
$((V8, V5))$	0.397
$((V1, V5))$	0.35
$((V6, V10))$	0.17
$((V3, V1))$	0.091

Tableau 5.2 – Tableau représentant les valeurs des PO triées

Le préordre Ω :

$$((v_3, v_1)) \leq ((v_6, v_{10})) \leq ((v_1, v_5)) \leq ((v_8, v_5)) \leq ((v_2, v_7)) \leq ((v_3, v_9)) \leq ((v_1, v_8)) \leq ((v_3, v_8)) \leq ((v_3, v_7)) \leq ((v_3, v_5))$$

Les classes formées à chaque niveau H sont représentées par le Tableau 5.3 ci-dessous :

Règle	H	Classe	Cohesion inter-classe $coh_{supp(n)M_{GK}}(\underline{C})$	Implication inter-classe $\psi(\underline{C}, \underline{C}')$
R(1)=	1	(V3,V5)	1.0	1.0
R(2)=	2	((V3,V5),V8)	0.734	1.0
R(3)=	3	(V1,((V3,V5),V8))	0.464	0.532
R(4)=	4	(V2,V7)	0.436	0.684
R(5)=	5	(V6,V10)	0.17	0.571

Tableau 5.3 – Tableau montrant les classes constituées, $coh_{supp(n)M_{GK}}(\underline{C})$ et $\psi(\underline{C}, \underline{C}')$

Au niveau H de la hiérarchie des cohésions implicatives, s'est formée une classe s_H variables ($s_H \leq s$) obtenue à partir de l'algorithme dans §5.2.2.1 et dont la cohésion est la plus forte que celle des classes qui ne sont pas encore formées et qui auraient pu se construire à ce niveau H . Notons \underline{V}_H cette classe de s_H variables et $S_H = \{1, \dots, s_H\}$ et soit $V_H = \{v_{f(1)}, v_{f(2)}, \dots, v_{f(s_H)}\}$ forme une ensemble PO_H de paires orientées où f est une injection de S_H dans S et $Card(PO_H) = C_{s_H}^2$.

Soit $PO_H = PO \cap (V_H \times V_H)$, la partition sur V définie au niveau H constituée des classes qui y sont déjà formées et des singletons non encore associés (Cf. 2.12, §2.7.4).

$PO_H = \{(v_{f(k)}, v_{f(l)}), k \in S_H, l \in S_H, l > k\}$; PO_H est le sous ensembles des éléments de PO formé par les **paires orientées de variables réunies à ce niveau H** .

Les variables qui ne sont pas dans V_H sont des éléments de \overline{V}_H et $\overline{S}_H = \{s_{H+1}, \dots, s\}$ et nous avons noté f l'injection $S_H \xrightarrow{f} S$ et notons f^* l'injection $\overline{S}_H \xrightarrow{f^*} S$.

$$v_{f^*(l)} \in \overline{V}_H, l \in \overline{S}_H \text{ et } Card(\overline{S}_H) = s - s_H.$$

Notons POS_H les sous-ensemble des éléments de PO tel que : $POS_H = PO \cap (V_H \times \overline{V}_H)$ et $POS_H = \{(v_{f(k)}, v_{f^*(l)}), k \in S_H, l \in \overline{S}_H, l > k\}$ désigne l'ensemble des **paires orientées de variables séparées par le niveau H** .

On considère la réunion de sous-ensembles de paires orientées jusqu'au niveau H :

$$POR_H = \bigcup_{k=1}^H PO_k. \text{ (Cf. Définition 2.46)}$$

Les **paires orientées de variables nouvellement réunies au niveau H** :

$$POR'_H = POR_H - POR_{H-1}$$

L'ensemble produit $POS_H \times POR'_H$ est un ensemble de couples de paires orientées : le premier élément du couple est une paire orientée de variables séparées par le niveau H et le deuxième élément une paire orientée de variables nouvellement réunies au niveau H .

L'ensemble $gr(\Omega) \cap [POS_H \times POR'_H]$ est constitué des couples de couples qui au niveau H respectent le préordre initial.

Il y aura deux cas extrêmes :

- **Cas 1** : $gr(\Omega) \cap [POS_H \times POR'_H] = \emptyset$
- **Cas 2** : $gr(\Omega) \cap [POS_H \times POR'_H] = POS_H \times POR'_H$ c'est-à-dire $POS_H \times POR'_H \subseteq gr(\Omega)$.

Dans cas 1 : le niveau H a introduit des paires de variables nouvellement réunies qui, avec les paires de variables séparées à ce niveau, ne respectent pas l'ordre initial et global Ω des PO , c'est-à-dire toute POS_H a une cohésion plus forte que celle que possède toute POR'_H pour la première fois. Donc, le niveau qui vient de se former n'apporte pas d'informations conformes à l'ordre initial et global des PO . Ce niveau H **n'apporte aucune information qui va dans le sens du préordre initial et global Ω** . On peut dire que, la cohésion et le préordre ne sont pas en résonance.

Dans cas 2 : toute POS_H a une cohésion plus faible que celle de toute POR'_H . Ainsi, cet ordre institué par le niveau H est conforme à l'ordre initial et global Ω . A ce niveau H , l'ordre **va totalement dans le sens de l'ordre initial et global Ω** . On peut dire que, la cohésion et le préordre sont en résonance.

Afin d'exprimer ces différents états possibles compris entre ces deux cas extrêmes, nous introduisons l'indice brut $\mu(H) = Card(gr(\Omega) \cap [POS_H \times POR'_H])$.

- **Dans cas 1 extrême**: $\mu(H) = 0$;
- **Dans cas 2 extrême**: $\mu(H) = Card(POS_H \times POR'_H)$

Dans tous les cas $\mu(H) \in \{0, 1, \dots, \text{Card}(POS_H \times POR'_H)\}$
avec $\text{Card}(POS_H \times POR'_H) \leq \text{Card}(gr(\Omega)) = C_s^2$

Associons au $\text{Card}(gr(\Omega) \cap [POS_H \times POR'_H])$ l'indice aléatoire $\text{Card}(gr(\Omega^*) \cap [POS_H \times POR'_H])$ où Ω^* est une préordonnance aléatoire dans l'ensemble, muni d'une probabilité uniforme, de toutes les préordonnances de même type cardinal que Ω .

Cet indice a pour espérance mathématique $\frac{\text{Card}(POS_H \times POR'_H)}{2}$

et de variance $\frac{\text{Card}(POS_H \times POR'_H)(\text{Card}(gr(\Omega)+1))}{12}$

Pour simplifier les notations, posons :

$\text{Card}(POS_H) = P_H$, $\text{Card}(POR'_H) = Q_H$, $\text{Card}(gr(\Omega)) = G = P_H + Q_H$
et $\text{Card}(POS_H \times POR'_H) = P_H \cdot Q_H$

Soit $s(\Omega, H)$ l'indice centré et réduit observé au niveau H . Ainsi on a,

$$s(\Omega, H) = \frac{\text{Card}(gr(\Omega) \cap [POS_H \times POR'_H]) - \frac{1}{2}P_H \cdot Q_H}{\sqrt{\frac{P_H \cdot Q_H (G+1)}{12}}} \quad (5.7)$$

Comme dans l'analyse des similarités de I-C. Lerman, cet indice sert de statistique globale des niveaux ; On peut poser finalement :

$$v(\Omega, H) = s(\Omega, H) - s(\Omega, H - 1) \quad (5.8)$$

Définition 5.6. On appelle **niveau significatif** tout niveau correspondant à un maximum local de $s(\Omega, H)$ au cours de la constitution de la hiérarchie implicative. Nous dirons dans ce cas que la partition PO_H est en résonance partielle avec Ω . Si de plus, $gr(\Omega) \cap [POS_H \times POR'_H] = POS_H \times POR'_H$, nous dirons que la partition PO_H est en résonance totale avec Ω .

Définition 5.7. On appelle **nœud significatif** tout nœud formé à un niveau correspondant à un maximum local de $v(\Omega, H)$

H	Classe	$s(\Omega, H)$	$v(\Omega, H)$
1	(V3,V5)	0.019	0.019
2	((V3,V5),V8)	0.02	0
3	(V1,((V3,V5),V8))	0.02	0.001
4	(V2,V7)	-0.013	-0.033
5	(V6,V10)	-0.014	-0.001

Tableau 5.4 – Tableau représentant de l'indice centré et réduit observé (*niveau significatif*) et l'indice de statistique globale (*nœud significatif*) de chaque niveau de classification.

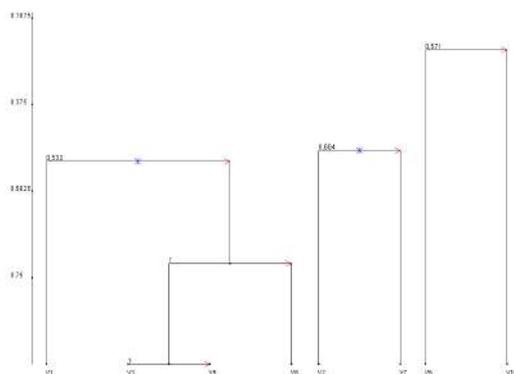


Figure 5.1 – Dendrogramme représentant les méta-règles du tableau 5.3

5.3 Contributions des individus

5.3.1 Paire orientée générique principale associée à une classe

Des informations complémentaires sont nécessaires pour pouvoir donner une analyse approfondie et enrichissante. C'est ainsi, la participation des individus afin de fournir des informations non négligeables dans l'analyse. Cette fois-ci, nous cherchons l'individu ou groupe d'individus qui contribuent à la classe constituée d'un niveau donné plutôt significatif. Ce qui veut dire, de déterminer le rôle de cet individu ou du groupe dans l'élaboration d'une classe d'un niveau considéré. Nous nous référons toujours à I-C. Lerman dans ses méthodes de classification.

Soit une classe d'implications à un niveau H et notée \underline{C}_H . L'ensemble des paires orientées qui constituent au même niveau H est $PO_H = \{(v_{f(k)}, v_{f(l)}), k \in S_H, l \in S_H, l > k\}$; où $(v_{f(k)}, v_{f(l)})$ se décompose comme suit : f est l'injection de S_H dans S , S étant l'ensemble qui indexe les s -caractères étudiés et S_H celui qui indexe les

s_H -variables qui forment la classe au niveau H . $PO_H = PO \cap (V_H \times V_H)$ où V_H est la classe de s_H -variables dont la caractéristique est d'avoir la cohésion la plus forte parmi toutes les classes non encore formées et qui auraient pu se construire à ce niveau.

Définition 5.8. *On appelle **paire orientée générique principale** de la classe \underline{C}_H , la paire orientée de cette classe dont la $supp_{(n)M_{GK}}$ est la plus grande parmi les paires nouvellement constituées. C'est-à-dire :*

Pour une classe $\underline{C}_H = ((\underline{C}_I, \underline{C}_J))$ où \underline{C}_I et \underline{C}_J sont deux classes d'implication formées à des niveaux inférieurs. Ainsi, nous avons, $I < H$ et $J < H$, alors la paire orientée générique principale de \underline{C}_H sera :

$$((v_i, v_j)) \in V_I \times V_J \text{ si } \forall v_k \in V_I, v_l \in V_J: supp_{(n)M_{GK}}(v_i \rightarrow v_j) \geq supp_{(n)M_{GK}}(v_k \rightarrow v_l).$$

D'après l'Equation 5.3, la cohésion entre ces deux classes est donnée par :

$$\psi(\underline{C}_I, \underline{C}_J) = [supp_{(n)M_{GK}}(v_i \rightarrow v_j)]^{Card(POR'_H)} \times [coh_{supp_{(n)M_{GK}}}(\underline{C}_I) \times coh_{supp_{(n)M_{GK}}}(\underline{C}_J)]^{\frac{1}{2}} \quad (5.9)$$

où $((v_i, v_j))$ est la paire orientée générique principale de la classe \underline{C}_H . La paire orientée générique principale de la classe est une information nécessaire pour caractériser ladite classe, car la $supp_{(n)M_{GK}}$ maximale a été prise pour se substituer aux autres dans le calcul de la cohésion de cette classe. Or chaque sous-classe formant la classe \underline{C}_H admet sa paire orientée générique principale et cette suite générique porte l'historique et chronologique de la construction de la classe \underline{C}_H . On appelle **paires génériques** de \underline{C}_H notées PG_H les éléments de cette suite générique et NG_H le nombre des paires génériques de la classe, c'est-à-dire le nombre de sous-classes constituant \underline{C}_H (\underline{C}_H incluse). Si une classe est formée de n -sous classes y compris la classe toute entière, on aura n -paires génériques et la suite générique est formée de n -éléments. Nous avons :

- $PG_H \subseteq PO_H$,
- $NG_H = Card(PG_H)$.

5.3.2 Vecteur puissance implicative d'une classe \underline{C}_H

Pour pouvoir déterminer la qualité des liaisons au sein des couples génériques (paires génériques) d'une classe, nous allons définir un vecteur comme un instrument de

représentation que nous attachons à la classe \underline{C}_H . Ces différentes images vectorielles d'éléments des graphes et hiérarchie permettent d'introduire une métrique sur des espaces de représentation où les individus pourront être projetés.

Définition 5.9. On appelle **vecteur puissance implicative** d'une classe \underline{C}_H , noté $VPC_{\underline{C}_H}$, l'élément de $[0, 1]^{Card(PG_H)}$ égal à la suite $(\Phi)_n, n \in \{1, \dots, NG_H\}$, c'est-à-dire : $(\Phi_1, \Phi_2, \dots, \Phi_{NG_H})$.

$VPC_{\underline{C}_H}$ est constitué de Φ_n qui est la $(supp_{(n)M_{GK}})_n$ de la $n^{\text{ème}}$ paire générique de \underline{C}_H .

5.3.3 Puissance implicative d'un individu et distance sur la classe \underline{C}_H

5.3.3.1 Respect d'un individu vis-à-vis d'une classe

La problématique c'est de savoir : *quel est l'individu ou groupe d'individus qui contribuent à l'élaboration de la classe implicative ?*. Ainsi, nous allons dans un premier temps d'aborder la notion de « respect » d'un individu par rapport à la classe, grâce à une mesure Φ_x de la définition 5.10. Alors, le respect qu'a un individu x , avec $x = \{i_1, i_2, \dots, i_n\}$ envers une classe \underline{C}_H est mesuré par l'application notée Φ_x définie sur l'ensemble des individus vers $[0, 1]$:

\underline{C}_H est une classe formée des deux classes \underline{C}_I et \underline{C}_J au niveau H , $(\underline{C}_I \rightarrow \underline{C}_J)$ et (v_i, v_j) : la paire générique principale de \underline{C}_H , $v_i \in V_I$ et $v_j \in V_J$. Un individu x quelconque respecte ou non l'implication de la paire générique $((v_i, v_j)) \in V_I \times V_J$ d'une classe \underline{C}_H si celui-ci est responsable dans la relation implicative de v_i sur v_j . Φ_x est donnée par le Tableau 5.5 ci-dessous :

x	v_i	v_j	$v_i \rightarrow v_j$	Φ_x
i_1	1	1	1	1
i_2	1	0	0	0
i_3	0	1	1	1
i_4	0	0	1	0,5

Tableau 5.5 – Tableau représentant le respect de l'individu x , noté Φ_x vis-à-vis de la classe formée par la paire générique $((v_i, v_j))$.

Pour un individu x , notons $v_i(x)$, la valeur que prend x sur la variable v_i et $v_j(x)$, la valeur que prend x sur la variable v_j .

Définition 5.10. Le respect Φ_x de l'individu x envers la paire générique $((v_i, v_j))$ est défini par :

$$\Phi_x = \begin{cases} 0 & , \text{ si } v_i(x) = 1 \text{ et } v_j(x) = 0 \\ 1 & , \text{ si } v_i(x) = 1 \text{ ou } v_i(x) = 0 \text{ et } v_j(x) = 1 \\ p = 0,5 \in [0,1] & , \text{ si } v_i(x) = 0 \text{ et } v_j(x) = 0 \end{cases} \quad (5.10)$$

p est une valeur arbitraire que l'utilisateur peut changer selon le contexte. Dans (Ratsimba-Rajohn, 1992), on prend $p = 0,5$, pour garantir la neutralité d'un individu ne satisfaisant pas les deux variables au regard de l'implication entre elles.

5.3.3.2 Vecteur puissance implicative d'un individu sur une classe

Nous avons donné dans Définition 5.10 une valeur Φ_x qui associe un individu x à une paire orientée. Par analogie, nous mesurons la valeur qui donnera le profil de l'individu sur une classe d'implications. Nous prenons la suite de respects de l'individu x envers les paires génériques de la classe \underline{C}_H .

Définition 5.11. Soit PG_H les paires génériques de la classe \underline{C}_H . Le **vecteur puissance implicative de l'individu x** , noté $VPI_H(x)$ est le NG_H -uplets $(\Phi_x PG_H)_n$, $n \in \{1, \dots, NG_H\}$ qui est le **respect** qu'a l'individu envers la $n^{\text{ème}}$ paire générique de \underline{C}_H . C'est-à-dire :

$$VPI_H(x) = (\Phi_x PG_H)_n = (\Phi_x((v_i, v_j)))_n, n \in \{1, \dots, NG_H\}.$$

5.3.3.3 Individu idéal théorique et individu idéal observé d'une classe

Le candidat idéal théorique est l'individu qui respecte totalement l'ensemble des paires générique de la classe, c'est-à-dire : $(\Phi_x((v_i, v_j)))_n = 1, n \in \{1, \dots, NG_H\}$, soit $VPI_H(w) = (1, \dots, 1)$.

Exemple 5.3.1. Soit $x = \{i_1, i_2, i_3, i_4, i_5, i_6\}$, l'individu idéal théorique qui respecte la classe est alors, $\forall n \in \{1, \dots, 6\}, i_n = (1, 1, 1, 1, 1, 1)$

Définition 5.12. On appelle **individu idéal observé**, l'individu x qui vérifie effectivement $VPI_H(x) = (1, \dots, 1)$.

5.3.3.4 Distance d'un individu à une classe

La problématique, c'est de définir une valeur quantifiable la proximité d'un individu à une classe d'implication donnée. Pour cela, nous avons une paire $((v_i, v_j))$ et un individu de x désigné dont les valeurs qui les caractérisent respectivement sont : $supp_{(n)M_{GK}}(v_i, v_j)$ et Φ_x par rapport à la classe formée par $((v_i, v_j))$. Au niveau H d'une classe d'implications, nous avons ses paires génériques PG_H et $NG_H = Card(PG_H)$ (Cf. Définition 5.8) et les deux vecteurs puissances implicatives VPC_H (Cf. Définition 5.9) qui caractérise la classe $VPI_H(x)$ (Cf. Définition 5.11) qui définit le rapport de l'individu x à cette classe.

Nous choisissons une distance de χ^2 dans l'espace des vecteurs puissances $[0, 1]^{Card(PG_H)}$ ou bien $[0, 1]^{NG_H}$. Et aussi, nous avons intérêt à prendre le complément à 1 de VPC_H , ceci afin de mettre en relief un phénomène qui est d'autant plus important que l'implication au voisinage de 1.

Définition 5.13. *On appelle **distance implicative d'un individu x à la classe \underline{C}_H au niveau H** la quantité définie par :*

$$d(x, \underline{C}_H) = \left[\frac{1}{NG_H} \sum_{i=1}^{NG_H} \frac{(\overline{VPI_H(x)} - \overline{VPC_H})^2}{\overline{VPI_H(x)}} \right]^{\frac{1}{2}} \quad (5.11)$$

avec $\overline{VPI_H(x)} = 1 - VPI_H(x)$ et $\overline{VPC_H} = 1 - VPC_H$.

5.3.3.5 Individu idéal empirique d'une classe

Dans (§5.3.3.3), nous avons défini un individu idéal théorique pour la classe \underline{C}_H . Soit $d(w, \underline{C}_H)$ la distance de cet individu à la classe \underline{C}_H . Alors, l'individu x dont la distance à la classe \underline{C}_H est inférieure à $d(w, \underline{C}_H)$ est en prendre en compte : i.e. on prend $d(x, \underline{C}_H) \leq d(w, \underline{C}_H)$. Puisque la distance de x à la classe \underline{C}_H est faible par rapport à la théorique, cet individu est alors conforme à l'ensemble des observations de l'expérience. D'une autre manière, ce sont les individus qui interviennent le plus dans la construction de \underline{C}_H .

Définition 5.14. *On appelle **individus idéaux empiriques de la classe \underline{C}_H** , l'ensemble des individus défini par :*

$$X_i(H) = \{x \in \mathcal{I} / d(x, \underline{C}_H) \leq d(w, \underline{C}_H)\} \quad (5.12)$$

5.4 Conclusion

Ce chapitre nous a permis de décrire quelques notions utiles pour une classification hiérarchique implicative et cohésitive selon la mesure de qualité M_{GK} . Il a été évoqué la construction de l'arbre hiérarchique orientée appelé « dendrogramme », la détermination des niveaux significatifs et les nœuds significatifs du dendrogramme d'une part, ainsi que l'établissement des relations pour la contribution des individus et les groupes d'individus par rapport à chaque classe formée d'autre part.

À l'insu de ces théories, l'implémentation de l'outil CHIC- M_{GK} se trouve dans l'Annexe B.1, qui nous permet de l'appliquer par rapport aux quelques expérimentations dans les chapitres qui suivent.

Troisième partie

Application en didactique de l'informatique : didactique de l'algorithmique

Chapitre 6

Pédagogie, didactique, didactique des disciplines

*“ La didactique est à la
pédagogie, ce que la théorie est à
la pratique. ”*

Mathieu Foria

Sommaire

6.1 Pédagogie	164
6.1.1 Pédagogie : une théorie éducative	165
6.1.2 Pédagogie : une pratique éducative	165
6.1.3 Évolution des conceptions pédagogiques	166
6.2 Didactique	169
6.2.1 Didactique générale	169
6.2.2 Le triangle didactique	169
6.2.3 Limites du triangle didactique	170
6.2.4 Trois dimensions de la recherche didactique	171
6.3 Didactique des disciplines	173
6.3.1 Le triangle didactique par rapport à l'introduction des TIC	174
6.3.2 Apport des théories constructivistes en psychologie à la didactique	179
6.3.3 Notion et concept d'activité	187
6.3.4 Modèle adopté dans le système d'activités	197

6.1 Pédagogie

Le mot pédagogie tire son origine de la Grèce antique *paidagogós*. Au début, le terme étant composé par *paidos* « enfant » et par *gogía* « mener » ou « conduire », son concept désignait l'esclave qui accompagnait les enfants à l'école. C'est une direction, éducation des enfants¹.

« La pédagogie désigne l'ensemble des méthodes et des pratiques d'enseignement et d'éducation en tant que phénomène typiquement social et spécifiquement humain. Il s'agit d'une science appliquée à caractère psycho-social, dont l'objet d'étude est l'éducation. La pédagogie reçoit des influences de plusieurs sciences, telles que la psychologie, la sociologie, l'anthropologie, la philosophie, l'histoire et la médecine, parmi d'autres »².

La pédagogie est alors un savoir ou un art pour certains chercheurs et une science pour d'autres.

La pédagogie peut se catégoriser selon plusieurs critères. On a tendance à parler de la pédagogie générale (qui a à voir avec les questions universelles et globales de la recherche et de l'action sur l'éducation) ou des pédagogies spécifiques (ayant systématisé tout un autre corps de la connaissance en fonction de plusieurs réalités historiques expérimentées). Ne pas confondre non plus la pédagogie traditionnelle avec la pédagogie contemporaine.

Il est important de faire la différence entre la pédagogie, en tant que science qui étudie l'éducation, et la didactique, en tant que discipline ou qu'ensemble de techniques qui aident à l'apprentissage. C'est pour cela que l'on peut dire que la didactique n'est qu'une discipline à l'intérieur de la pédagogie.

La pédagogie a déjà été comparée à l'andragogie, qui est la discipline éducative chargée d'instruire et d'éduquer l'homme en permanence à n'importe quel moment de son développement en fonction de sa vie culturelle et sociale.

« Qu'est-ce que la pédagogie ? C'est l'enveloppement mutuel et dialectique de la théorie et de la pratique éducatives par la même personne, sur la même personne. Le pédagogue est un praticien-théoricien de l'action

1. Dictionnaire de l'Académie française :9^{ème} édition

2. Définition de pédagogie - Concept et Sens. Disponible sur : <http://lesdefinitions.fr/pedagogieixzz3OUFrM9VA>, consulté le 01/06/2017

éducative. Il cherche à conjoindre la théorie et la pratique à partir de sa propre action, à obtenir une conjonction parfaite de l'une à l'autre, tâche à la fois indispensable et impossible en totalité (sinon il y aurait extinction de la pédagogie). »(Houssaye, 1993, p.13)

Le mot « Pédagogie » est apparu pour la première fois dans la langue française en 1485, avant d'entrer dans le Dictionnaire de l'Académie Française en 1761 où il se renvoyait « *soit aux pratiques du pédagogue soit au discours plus ou moins savant qu'on peut tenir sur elles* » . . . « *Au début du XX^{ème} siècle, E. Durkheim³ s'en tient, en l'opposant au terme éducation, à cette seconde acception mais il nous semble que dans la plupart des discours ordinaires ou savants, c'est la première acception qui l'emporte* » (Bessé, 1995). Il en est de même dans le Dictionnaire de didactique des langues, où pédagogie a deux sens : une **théorie** et une **pratique éducatives**

6.1.1 Pédagogie : une théorie éducative

La pédagogie est une réflexion théorique à caractère philosophique et psychologique sur les méthodes d'enseignement, les actions à exercer en situation d'apprentissage, leurs finalités et leurs orientations. Les auteurs donnent comme exemples : la simplicité, la progressivité, l'émulation (Galisson et Coste, 1976).

6.1.2 Pédagogie : une pratique éducative

La pédagogie est aussi une action pratique « constituée par l'ensemble des conduites de l'enseignant et des enseignés dans la classe ». La pédagogie renvoie à un ensemble de méthodes, de procédés, de techniques mis en œuvre, selon la situation d'enseignement, par l'enseignant pour instruire et former les élèves⁴ (Altet, 1994).

Exemple 6.1.1. *La pédagogie audiovisuelle, pédagogie de projet, pédagogie active, etc. Dans le même sens, Marguerite Altet définit la pédagogie comme étant « (...) ce champ de la transformation de l'information en savoir par la médiation de l'en-*

3. David Émile Durkheim, né le 15 avril 1858 à Épinal est un sociologue français considéré comme l'un des fondateurs de la sociologie moderne.

4. « L'enseignement peut être conçue comme un processus de traitement de l'information et de décision en classe où le rôle de la dimension relationnelle et de la situation vécue demeure essentiel ; c'est ce vécu interactif en situation d'enseignement-apprentissage qui constitue le champ de la pédagogie. »

seignant, (...) » et le pédagogue comme « celui qui facilite la transformation de l'information en savoir... »⁵.

La pédagogie comporte alors un aspect théorique et un autre pratique. Ces deux aspects sont loin d'être séparés, mais ils s'enveloppent l'un à l'autre⁶. Or, étant donné que « pédagogie » renvoie tantôt à une philosophie de l'éducation appliquée ou à une psychologie appliquée, tantôt à un art d'enseigner, ce terme a été discrédité en France surtout par Michel Dabène et Christian Puren qui lui substituent le terme de *didactique* (Cuq et Gruca, 2002).

6.1.3 Évolution des conceptions pédagogiques

Ladjili (2009) fournit une explication intéressante entre la pédagogie traditionnelle et la pédagogie nouvelle :

L'apposition entre éducation traditionnelle et éducation nouvelle n'est faite par rapport à la relation binaire entre le professeur (P) et le savoir (S) qui a caractérisé le premier courant, puis le professeur et l'élève qui a caractérisé le second. Le premier courant prône un modèle transmissif du savoir, par le professeur, d'une manière organisée, et formalisée, à un élève récepteur selon un schéma béhavioriste (S-P) pour lequel apprendre c'est répéter, reproduire. A l'antipode, le courant dit moderne privilégie l'activité du sujet apprenant. La pédagogie est devenue active, inspirée par les théories de l'apprentissage et les thèses constructivistes de la psychologie génétique. De nos jours, une autre façon de décrire cette évolution considère que la situation pédagogique ne peut être définie que par un triangle dans lequel le professeur, le savoir et l'élève occuperaient chacun un des sommets. Ce schéma illustre mieux cette évolution car il permet de dégager, en gros, trois courants en fonction des segments liant les pôles du triangle deux à deux. (p.1-2)

5. Pédagogie, didactique générale et didactiques disciplinaires, Saïd Tasra, <https://halshs.archives-ouvertes.fr/halshs-0153181>, soumis le 02/06/2017, p.2

6. « Qu'est-ce que la pédagogie ? C'est l'enveloppement mutuel et dialectique de la théorie et de la pratique éducatives par la même personne, sur la même personne. Le pédagogue est un praticien-théoricien de l'action éducative. Il cherche à conjoindre la théorie et la pratique à partir de sa propre action, à obtenir une conjonction parfaite de l'une à l'autre, tâche à la fois indispensable et impossible en totalité (sinon il y aurait extinction de la pédagogie). », Jean Houssaye (dir.), *La pédagogie : une encyclopédie pour aujourd'hui*, Paris : ESF, 1993, p. 13

6.1.3.1 Le triangle pédagogique

Jean Houssaye inscrit son triangle dans la situation pédagogique. Il y est question de déterminer la place, le rôle dévolu ou revendiqué par les trois actants. Selon la

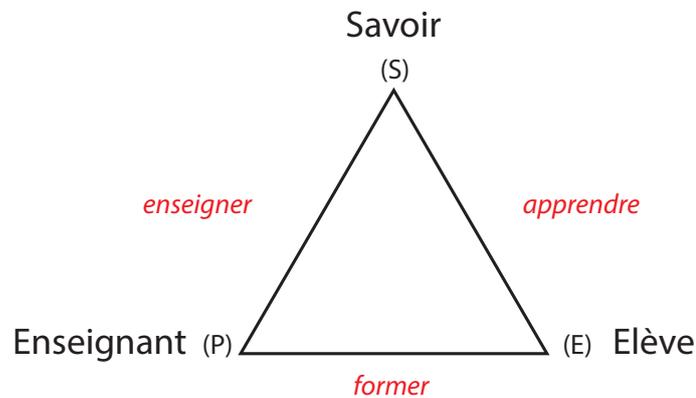


Figure 6.1 – Triangle pédagogique de Houssaye

configuration de la triade qui est ainsi déterminée par la relation privilégiée de deux des trois actants, [Houssaye \(1988, p.267\)](#) détermine trois processus pédagogiques possibles :

- **enseigner** : sur l’axe Savoir (S) – Enseignant (P), lorsque, par exemple, l’élève est exclu du jeu ;
- **former** : axe Enseignant (P) – Elève (E), lorsque, par exemple, l’appropriation du savoir n’est plus la priorité ;
- **apprendre** : axe Elève (E) – Savoir (S), lorsque, par exemple, l’enseignant s’efface pour interférer le moins possible entre l’élève et le savoir.

Trois classifications de la pédagogie ont été induites selon toujours [Ladjili \(2009\)](#)

1. *La pédagogie traditionnelle qui a privilégié le rapport du professeur au savoir et a relégué l’élève au second plan, à un « degré zéro » de la culture . Le professeur s’érige en modèle intermédiaire entre l’élève et les grands chefs d’œuvre de l’humanité.[...] De ce fait, l’élève n’a aucune existence propre et c’est la vénération devant un savoir sacré et un modèle sans faille. La simple maîtrise, par*

l'enseignant, des contenus de savoir est supposée être le garant de l'apprentissage chez l'élève.[...]

2. *La pédagogie relationnelle qui, comme son nom l'indique, s'est préoccupée de gérer la relation professeur-élève considérant que tout apprentissage s'explique par la relation « séductrice » de l'enseignant avec l'enseigné.[...] Le professeur devient animateur. On suppose, dans ce cas, que tout apprentissage passe à travers cette relation pédagogique. C'était la mode des courants libertaires et de la dynamique des groupes.*
3. *La pédagogie inspirée de la psychologie de l'apprentissage qui met en relation directe l'enfant avec le savoir en maintenant le professeur dans une position d'accompagnateur dans la situation d'apprentissage.(p.3-4)*

Dans ce cas, l'enseignant est alors un garant de l'apprentissage chez les élèves en maîtrisant les contenus de ce qu'il allait enseigner, prend le rôle de séducteur pour que les élèves aient la motivation et s'intéressent à l'apprentissage et enfin prend le rôle d'accompagnateur aux élèves durant l'apprentissage.

Apprendre c'est quoi alors ?

Pourquoi n'y a-t-il pas un mot en anglais pour l'art d'apprendre ? [...]

La pédagogie, l'art d'enseigner, sous ses différents noms, a été adopté par le monde universitaire comme un domaine respectable et important.

L'art d'apprendre est un orphelin universitaire.(Traduction libre Papert, 1993, p.82)

La notion de l'apprentissage dépend des auteurs et évolue dans le temps. Dans ce système triadique et dynamique l'élève devient acteur de sa formation . Il s'agit plus d'une action formatrice qu'informatrice. Cette action est inspirée des théories constructivistes qui soutiennent que la prise en charge par le formé de sa propre formation renforce l'apprentissage. Nous imaginons déjà l'incidence de cette approche sur la formation des enseignants et le mode de leur préparation à la mise en place de situations d'apprentissage (connaissance de l'élève, mécanismes d'apprentissage, pédagogie différenciée, évaluation etc....). C'est ce système didactique dans ses définitions et les concepts qu'il s'est forgés que nous allons présenter dans la section suivante.

6.2 Didactique

6.2.1 Didactique générale

La didactique est une discipline qui s'occupe de l'enseignement/apprentissage d'un certain contenu (la didactique générale), et de l'enseignement-apprentissage des connaissances déterminées relevant d'une discipline déterminée (Didactique spécialisée ou disciplinaire) et de leurs interrelations. La question centrale dans toute didactique est celle des savoirs, de leur enseignement-apprentissage dans un contexte scolaire précis. Sans cela, il n'y a pas de didactique. L'origine de la réflexion didactique remonte vers la fin des années 60, à l'occasion de la « réforme des mathématiques modernes ». Et cette réflexion, dit Samuel Johsua, « a constitué une véritable révolution dans un secteur entier de l'enseignement ». Le terme de didactique a, lui aussi, une histoire, ajoute Johsua. En fait, à l'époque, on hésitait entre le terme de didactique et celui d'« épistémologie expérimentale ». « Épistémologie », parce qu'il s'agit pour Guy Brousseau de l'organisation des savoirs ; et cette épistémologie est dite « expérimentale » étant donné qu'elle devait être mise en œuvre « en situation ». Mais en fin de compte et pour plusieurs raisons, c'est le terme de didactique qui a été retenu et popularisé grâce aux travaux de [Brousseau \(1978\)](#) ; [Brousseau, Wittwer, Delis, et Caid \(1969\)](#), dans le domaine de la didactique des mathématiques. Par la suite, le terme s'est imposé dans des disciplines proches des mathématiques : dans la didactique de la physique avec [Halbwachs \(1975\)](#) et peu après dans d'autres didactiques. Les principaux domaines d'investigation de la didactique sont : l'étude du curriculum (relation Enseignant-Contenu), l'étude de l'enseignement (relation Élève-Enseignant), l'étude de l'apprentissage (relation Élève-Contenu), et les interrelations entre ces trois domaines. Ces trois pôles de réflexion sont représentés historiquement par le « triangle pédagogique » (Cf. [Figure 6.1](#)). Grâce aux développements récents de la didactique, le « triangle pédagogique » s'est progressivement transformé en « triangle didactique général » (Cf. [Figure 6.2](#)) et puis en « triangle didactique », tel qu'illustré par [Chevallard \(1991\)](#) (Cf. [Figure 6.3](#)).

6.2.2 Le triangle didactique

Le triangle didactique, quant à lui, s'inscrit dans une structure systémique, appelée système didactique. Le système didactique détermine également trois axes, toujours

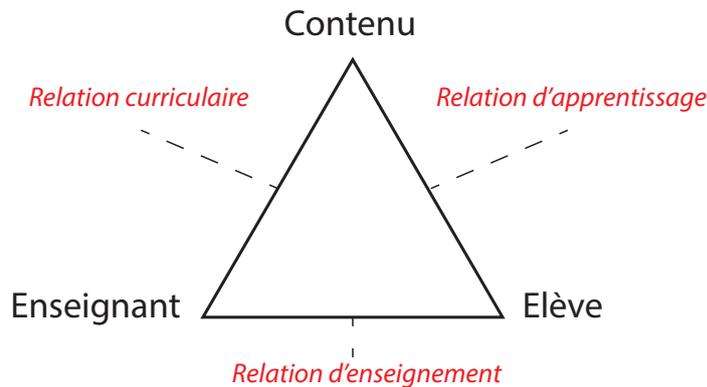


Figure 6.2 – Didactique générale

à partir des relations nouées entre les trois pôles, mais ces interactions vont servir à caractériser des points de vue particuliers quant au rapport au savoir. Elles vont conditionner des heuristiques selon trois approches disciplinaires à la fois distinctes et complémentaires (Cf. Figure 6.3) :

- **approche épistémologique** : sur l'axe Savoir - Enseignant
- **approche psychologique** : axe Enseignant - Élève
- **approche pédagogique** : axe Élève - Savoir

Ces trois axes offrent autant de dimensions à explorer. Ils servent à désigner des recherches, des travaux, des pistes, des entrées possibles à la réflexion dont le système didactique est l'objet. Ici, la question des méthodes pédagogiques n'est pas primordiale (Reuter, Cohen-Azria, Daunay, Delcambre-Delville, et Lahanier, 2007). La réflexion va plutôt s'orienter sur les interactions systémiques de ces trois dimensions à l'œuvre dans toute situation d'enseignement-apprentissage, ainsi que sur le rapport au savoir que ces interactions interrogent.

6.2.3 Limites du triangle didactique

Les didacticiens invoquent des critiques, des réserves ou des rappels à l'ordre quant à l'appréhension et l'utilisation du triangle didactique (Cf. Figure 6.3).

Le système didactique ne doit pas être réduit à l'espace de la classe ni au seul temps du cours (Reuter et al., 2007). Plus largement, il se dessine dans toute situation où une personne apprend intentionnellement quelque chose à une autre. La notion

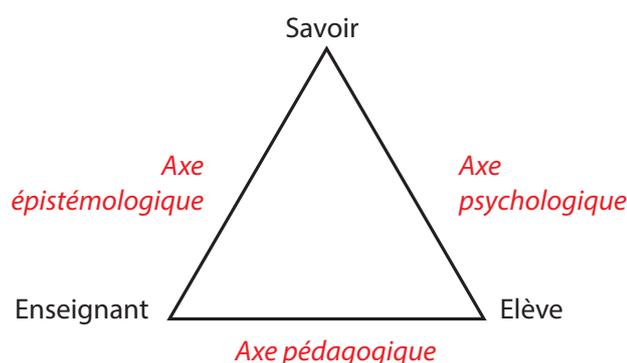


Figure 6.3 – Triangle didactique de Chevillard

de « *Savoir* » ne saurait être appréhendée de manière générique. Il convient plutôt de distinguer ses différents états : *savoir à enseigner/savoir effectivement enseigné*, i.e. présenté à la classe (Reuter et al., 2007). C'est d'ailleurs le but du système didactique que de produire ces derniers. De même, il importe d'identifier son statut épistémique, en différenciant les contenus d'enseignement des savoirs de référence (savants, experts, sociaux).

Enfin, la critique interroge la pertinence, au sommet du triangle, du pôle « *Savoir* ». Peut-il être considéré acteur au même titre que les deux autres ? Certains chercheurs préféreraient voir là l'« *Institution* » (Cornu et Vergnion, 1992), tandis que d'autres proposent d'inscrire le triangle dans une figure plus vaste qui permettrait ainsi de l'inclure dans un contexte éducatif et social (Audigier, 1996; Reuter et al., 2007).

6.2.4 Trois dimensions de la recherche didactique

Malgré la limite du triangle didactique, on pourra établir trois grands axes de recherche en didactique (Cf. Figure 6.4). S'agissant de l'axe 1, la dominante épistémologique est largement reconnue pour désigner l'élaboration didactique des contenus d'enseignement. Il est également observé une stabilité pour la dominante psychologique, ou psycho-cognitive (Axe 2), désignant la problématique d'appropriation de ces contenus par les élèves. Mais s'agissant du troisième domaine (Axe 3), la diversité des appellations recensées révèle deux aspects. Le premier témoigne de la pluralité des approches quant aux objets de l'investigation ; le second désigne la base du triangle didactique comme le lieu de la polémique entre pédagogie et didactique. Ainsi

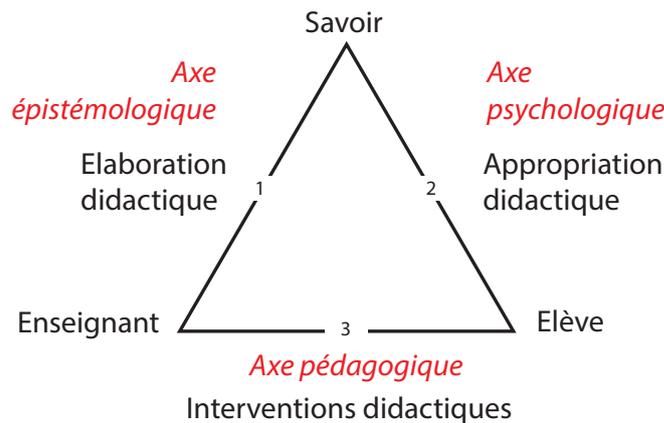


Figure 6.4 – Les trois axes de recherche en didactique

les appellations convoquées pour nommer ce champ de recherche divergent-elles : champ de la pédagogie et des Sciences de l'éducation (Develay, 1992), ingénierie pédagogique (Develay, 1992), pôle de la situation de formation (Meirieu, 2005 [ref. du 06-05-2018]), pôle Former (Bronckart et Chiss, 2005 [ref. du 06-05-2018]), dominante praxéologique (Halte, 1992), domaine de l'intervention didactique (Halte, 1992; Langlade, 1997). Pour Altet (1994), qui reconnaît pourtant aux didactiques l'empire sur les deux axes épistémologique et psychologique, la base du triangle ne peut relever que du domaine de la pédagogie.

6.2.4.1 La dimension épistémologique (Axe 1 : Savoir-Enseignant)

L'axe 1 que nous appelons : élaboration didactique (Halte, 1992) des savoirs à enseigner. Les didacticiens se proposent d'y examiner les objets d'enseignement et, en particulier, d'y répertorier les principaux concepts de la discipline, d'étudier leurs relations, leur structuration et leur hiérarchisation à l'intérieur du domaine considéré. La question de la référence et de l'origine des savoirs y est également posée avec, notamment, l'histoire des savoirs référents, qu'ils soient savants, experts ou sociaux. S'agissant des savoirs scientifiques, il reste à déterminer quel **cheminement** ils ont suivi et quels **obstacles** ils ont rencontrés et surmontés. Il convient en effet de sortir du « *hiatus (coupure) pédagogique* » qui fait que les savoirs scolaires sont « *présentés comme des faits établis* » sans histoire, sans dimension culturelle, humaine ou sociale (Giordan, 1994) et surtout à ne pas ignorer la dimension économique et politique que nous avons déjà rappelé à l'introduction comme sources des obstacles : c'est

l'ESEPC. S'agissant des savoirs scolaires et de leur histoire institutionnelle, on étudiera la manière dont ils ont été transposés, comment et pourquoi ils sont apparus « habitus » (Astolfi et Develay, 1989; Halte, 1992; Giordan, 1996).

6.3 Didactique des disciplines

Pendant longtemps ce concept n'a été employé que comme adjectif qualificatif. Mais en tant que substantif il n'est apparu que les années 1950 pour désigner l'art d'enseigner.

- Employé comme adjectif, il apparaît, dans le Grand Larousse Encyclopédique, comme étant ce qui sert ou qui est propre à l'enseignement.
De son côté, le Quillet lui donne le sens de ce qui est propre à enseigner. On parle alors de procédé didactique, d'une démonstration didactique ou encore de matériel didactique.
- Employé comme substantif, ce terme fait considéré pendant longtemps comme synonyme de pédagogie. La didactique apparaît alors comme une science s'occupant des problèmes de l'enseignement.

Mais en réalité, la didactique n'est pas synonyme de pédagogie. Elle ne constitue, selon Lacombe (1989) « *ni une discipline, ni une sous-discipline, mais une démarche, ou plus précisément, un mode d'analyse des phénomènes de l'enseignement.* » Et de ce fait elle requiert un statut particulier autonome parmi les sciences de l'éducation opposant la pédagogie à la didactique, Avanzini (1976) donne la précision suivante : « *La pédagogie désigne seulement la réflexion sur l'éducation de l'enfant. L'étude des méthodes d'enseignement ou de travail scolaire constitue l'objet de la didactique* ». Il n'y a pas un fait de théorie didactique et encore moins de recettes didactiques. Mais il y a, comme le dit Revuz (1980) « *une culture didactique qui ne dictera jamais dans le détail sa conduite au professeur, mais éclairera suffisamment la démarche pour qu'il puisse prendre en connaissance de cause de bonnes décisions* ». Pour Vergnaud (1983), « *la didactique n'est réductible ni à la connaissance d'une discipline, ni à la psychologie, ni à la pédagogie, ni à l'histoire, ni à l'épistémologie. Elle suppose tout cela, elle ne s'y réduit pas* ».

D'après Astolfi et Develay (1989), la didactique : « *c'est prendre des distances avec*

la pratique...pour mieux y revenir » et l'approche didactique travaille à la fois en amont et en aval de la réflexion pédagogique.

- En amont, l'intérêt est porté sur les contenus de l'enseignement comme objet d'étude. Cette préoccupation nous met d'emblée dans un champ épistémologique. La didactique s'intéresse aux concepts qui fonctionnent dans une discipline donnée : leur histoire, leurs rectifications successives et les modalités de leur introduction dans l'enseignement. Cette démarche faite d'avancées et de reculs s'explique par les obstacles rencontrés qui justifient ces rectifications, ces ruptures. Et c'est à juste titre que [Bachelard \(1938\)](#) dit : « *c'est en termes d'obstacles qu'il faut poser le problème de la connaissance scientifique* » (les causes d'inertie étant appelées obstacles épistémologiques).
- En aval, l'intérêt est porté sur l'élève et la compréhension des phénomènes de l'intérieur et non plus en tant que résultat. Ce qui intéresse le didacticien c'est l'étude des représentations chez les élèves, leurs réactions, leurs erreurs et leurs modes de raisonnement ainsi que les réactions de l'enseignement à l'égard de ces phénomènes.

6.3.1 Le triangle didactique par rapport à l'introduction des TIC

Intégrer les TIC dans ses enseignements est de mettre les élèves au service de ses apprentissages. C'est aussi chercher une efficacité et des gains au niveau des temps d'apprentissage, des activités des apprenants et de leur motivation. C'est aussi un vecteur de changement des pratiques pédagogiques des enseignants. L'utilisation des techniques de communication modifie, d'une certaine façon, les rapports entre l'enseignant et les apprenants. En effet, ***délaisser momentanément le contact direct, le tableau et la craie*** pour le compte d'une espace de travail synchrone impose à l'enseignant la mise en place d'un dispositif commun qui peut devenir collectif.

Que ce passe t-il du côté de la relation didactique lorsque l'enseignant introduit l'ordinateur dans ses stratégies d'enseignement ? Quels sont les différents rôles que peut jouer l'ordinateur dans le processus d'enseignement-apprentissage ? Quels changements pédagogiques implique l'intégration des TICE dans ses propres enseigne-

ments ?.

Le recours aux TIC modifie complètement le rapport entre le formateur et le groupe d'information et chaque personne formée

« (...)le formateur est donc confronté à une procédure de production de son action de formation qui ne lui appartient plus entièrement car elle doit être coordonnée en fonction des contraintes des matériels et équipement, de la disponibilité d'autres professionnels ». (Cazellet, 2003)

L'enseignant devient un accompagnateur qui assure une médiation à distance via l'ordinateur et internet dans l'objectif de favoriser les apprentissages de ses étudiants. L'enseignant est dans une situation qui lui impose la mise en place de pratiques qui ne sont pas, de nature, siennes. Il se doit d'organiser son action en fonction des contraintes des matériels et des équipements ainsi que de la nature du savoir à dispenser.

L'ordinateur a un rôle et une place complexes. Il ne peut pas prendre la place du professeur et le remplacer ni occuper un sommet du triangle didactique. Il est dans un premier cas conçu comme diffuseur de savoirs et se substituer, par conséquent au pôle discipline ou matière. Il est aussi conçu comme ayant la possibilité de prodiguer des consignes et se substituer au pôle enseignant. Et il est, en dernier lieu, conçu comme outil de communication se situant ainsi sur les cotés du triangle.

L'utilisation où l'ordinateur est considéré comme passif à un autre où certains types d'interactions lui confèrent le caractère actif. Nous pouvons détailler ci-dessous les différents rôles de l'ordinateur en tant que l'une des composantes principales des TIC.

- **L'ordinateur outil (OO)**: Dans ce cas, l'ordinateur n'est porteur d'aucune pédagogie. C'est l'enseignant qui assure toutes les actions et les tâches. Il l'utilise comme outil pour véhiculer des informations de manière claire et rapide. « L'ordinateur reste un outil de présentation des cours, illustrer des images, graphiques et des expériences, ça aide aux explications ». L'ordinateur ne peut pas remplacer la tâche d'un professeur.
- **L'ordinateur comme tuteur (OT)**: Dans ce cas, l'ordinateur guide les élèves vers certains types d'apprentissages. Il les amène à suivre un nombre d'étapes où essais et erreurs se succèdent pour aider les étudiants à reconnaître leurs lacunes et acquérir des savoirs sans « l'intervention » de l'enseignant.

Alors, les élèves, qui ont mal compris les explications du professeur, peuvent se rabattre sur certains logiciels bien conçus qui permettent la répétition et qui peuvent faire l'objet d'utilisations externes à la classe. Cela fait renforcer les enseignements déjà dispensés auparavant ». C'est ainsi que les temps d'apprentissages se trouvent prolongés ; donnant plus de chances aux apprenants de maîtriser les notions dispensés par les professeurs. L'interactivité que présente l'ordinateur permet de se passer « momentanément » de la présence de l'enseignant. Cependant, même si on gagne beaucoup au niveau de la qualité des interactions avec l'ordinateur, vu que tous les étudiants sont, à un certain moment, actifs, on perd au niveau de la qualité vu que logiciel ne peut pas prendre en compte les différences des apprenants et leurs divers questionnements. « Seul le professeur peut reconnaître ses étudiants... Cette fois-ci encore, l'ordinateur ne peut pas remplacer la tâche d'un professeur.

- **L'ordinateur enseigné (OE)**: Dans ce cas, c'est l'étudiant qui « programme » l'ordinateur. Dans le cadre d'une situation particulière, l'apprenant implémente les variables que la machine prendra en compte et c'est lui qui dicte l'opération à faire et la démarche à suivre. La logique constructiviste prend le dessus sur celle behavioriste décrite par le rôle précédent. Les élèves sont actifs et simulent le fonctionnement d'une expérience ou d'une machine. Ils se doivent de comprendre son fonctionnement pour pouvoir visualiser les résultats escomptés. « *Ce rôle de l'ordinateur, intéressant d'un point de vue constructiviste, demande cependant beaucoup d'engagement de la part de l'enseignement* » (Mangenot, 1996). En effet, les sujets interrogés insistent à mettre à la lumière l'effort qu'ils doivent fournir pour initier leurs apprenants à l'utilisation des logiciels de simulation qu'ils utilisent.
- **L'ordinateur comme environnement pédagogique (OEP)**: Dans ce cas, l'ordinateur fait partie intégrante de la stratégie pédagogique de l'enseignant. Il a un rôle fédérateur puisque c'est à travers lui qu'un environnement pédagogique spécifique est mis en place. « *l'ordinateur est là pour constituer une continuité à sa pédagogie (enseignant) et à la compléter* ». Son rôle est celui d'aider l'enseignant à aller plus loin dans ses enseignements et dans les apprentissages qu'il dispense.

Le triangle didactique classique unissant l'enseignant, l'apprenant et le savoir pour-

rait évoluer pour prendre en compte un quatrième « pseudo-pôle » : les TIC, désormais omniprésents et constituant un axe par lequel passent les savoirs ainsi que les interactions. Paquelin (2002) dit que « les TIC agissent comme outils cognitifs, d'aide à l'apprentissage et comme outils médiateurs et transformateurs des activités naturelles de connaissance et de communication humaine ». Il propose de transformer le triangle didactique classique en un *triangle didacTIC* (Cf. Figure 6.5) « qui pose les TIC comme un objet vecteur d'apprentissage et de nouveaux rapports socio-pédagogiques » et où apparaissent trois secteurs qui « en complémentarité, organisent l'apprentissage : l'acquisition, la communication ».

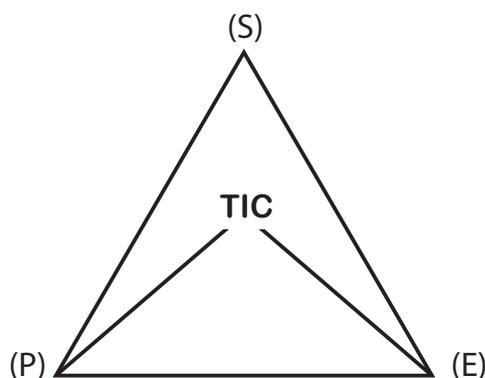


Figure 6.5 – Le triangle didacTIC

Comme le dispositif Technologique⁷ externalise les choix pédagogiques (Barcheath et Pouts-Lajus, 1990) et les rend visibles et donc analysables. Plus précisément la médiatisation (Mangenot, 1999) « rend durable, fixés par leur implémentation technologique et donc analysable les choix stratégiques effectifs des acteurs de la séquence pédagogique, qu'on peut alors comparer aux intentions annoncées. »

De nombreux enseignants expriment encore actuellement, lorsqu'ils abordent l'usage scolaire des technologies, pensent que la machine reste toujours une machine. Tantôt imaginent-ils que ce dispositif technologique serait neutre et porteur d'aucune intention pédagogique, tantôt ils semblent penser que les technologies s'opposeraient aux valeurs humanistes telles que la pédagogie, la culture littéraire ou la relation aux élèves. Ils ne sont donc guère d'humeurs à analyser les technologies sur un plan

7. Ce terme est parfois employé pour décrire le support technologique dans lequel s'incarnent les choix et stratégies pédagogiques de(s) auteurs, notamment, et il se réfère alors à un site Web ou un cd-rom, une vidéo, etc.

médiation, communication, ni même en termes de pédagogie, de didactique, de stratégies d'apprentissages. Afin de développer une attitude réflexive (Schön, 1994), il est temps d'exposer à ces enseignants en formation à un outillage de réflexion qui rende explicite les interactions entre les acteurs principaux de la relation pédagogique avec les Technologies.

Lombard (2003) considère que le triangle pédagogique de Houssaye devrait prendre une nouvelle dimension et se transformer en un tétraèdre sous l'influence des TIC. Pour cet auteur, l'apport des TIC enrichit la situation pédagogique d'un quatrième pôle d'interaction. Il propose ainsi de développer le triangle pédagogique selon la troisième dimension et prendre en compte une nouvelle variable d'une importance capitale : on a dès lors un tétraèdre pédagogique d'intégration des technologies ou plus simplement un tétraèdre d'intégration. A partir de cette conception, Lombard développe l'idée d'un tétraèdre, plaçant les TIC comme le quatrième pôle, appelé « *dispositif cyber-prof* ».

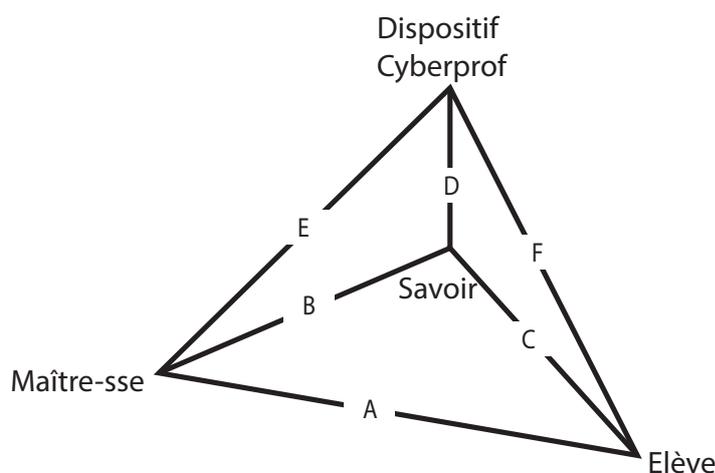


Figure 6.6 – Le tétraèdre de Lombard

Ce tétraèdre permet d'analyser les interactions entre les acteurs de la relation pédagogique : *maître-élève-savoir* et *dispositif-cyber-prof*. Il se compose de quatre faces triangulaires et six arêtes : Enseigner - Apprendre - Former - Faciliter - Partager - Participer. « *C'est un outil permettant de disséquer et d'explicitier la circulation entre les pôles de la situation pédagogique au cours de l'activité pour mettre en évidence ses contradictions, éclairer ses incohérences ou illustrer les cheminements féconds* » (Lombard, 2003).

Les travaux de Faerber (2002) ; Cissé (2008) montrent que le triangle pédagogique de Houssaye est affaibli voire « incapable de rendre compte de l'ensemble de cette situation pédagogique qui ne met plus les deux protagonistes en situation d'interagir face à face ». Il observe que les relations entre les différents pôles de ce triangle (*élève-enseignant, élève-savoir et enseignant-savoir*) se transforment pour prendre en compte d'autres nouvelles. En effet, en plus des trois pôles classiques, le *groupe* mérite une place avec les autres pôles du triangle. Son introduction comme pôle à part entière conduit à une figure de forme tétraédrique où les interactions dans le groupe « *communauté éducative* » vont dans tous les sens et où le contexte de médiations s'étend à tous les individus.

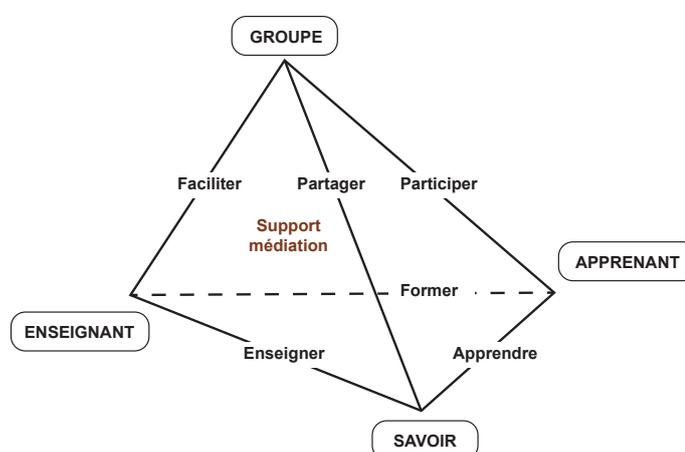


Figure 6.7 – Le tétraèdre pédagogique de Faerber

6.3.2 Apport des théories constructivistes en psychologie à la didactique

Dans le modèle traditionnel de la classe : « *modèle transmissif* », c'est l'Enseignant (P) qui parle la plupart du temps, il dispose les savoirs, les explique et les démontre. Il attribue les tâches et régule les apprentissages. Il gère et suggère. Il est en avance par rapport à ses Elèves (E), puisqu'il est le seul maître qui connaît la suite de son enseignement. Il est le seul émetteur et destinataire pour ses élèves.

Les théories constructivistes soutiennent que la prise en charge par le formé de sa propre formation renforce l'apprentissage. Nous allons aborder ces théories pour mieux en saisir les fondements et présenter les positions de Philippe Jonnaert, parmi

des auteurs constructivistes dans les littératures pour y déterminer le rôle de l'enseignant et de l'apprenant, la pertinence et la faisabilité du constructivisme en prenant le contexte éducatif en informatique, plus particulièrement en algorithmique dans la classe de seconde dans la zone Nord de Madagascar.

6.3.2.1 Le constructivisme selon Philippe Jonnaert

L'enseignant et la mise en place des situations d'apprentissage : La mise en place de situation d'enseignement et d'apprentissage est indispensable pour faciliter la construction des connaissances par l'élève. Jonnaert (1996) définit la situation « *comme une catégorie d'activités proposées aux élèves au même titre que le sont les exercices systématiques, les problèmes à résoudre, les activités de synthèse ou de structuration, le jeux,...* ». Le rôle de l'enseignant est alors de contextualiser l'objet à apprendre pour que les élèves puissent construire leurs connaissances dans des situations qui paraissent être significatives pour eux. Et on peut réutiliser ces connaissances acquises dans d'autres situations. Les situations qui ont du sens pour l'apprenant doivent être reconnues, ainsi l'auteur décrit les critères qui les caractérisent : les rapports entre les situations et les connaissances du sujet, les situations et les savoirs à apprendre et les situations et les résultats attendus. Mais les situations d'apprentissage et les compétences des élèves sont généralement incohérentes et inadaptées. Pour Jonnaert (2009), les compétences ne sont pas seulement cognitives. Elles se manifestent dans des situations comportant des dimensions contextuelles importantes, à savoir : l'environnement, l'informatique, la bibliothèque, les logiciels, le temps consacré à la tâche, ... De plus, il faut considérer, d'après l'auteur, les côtés affectifs et sociaux dont l'objectif est de mettre en œuvre un maximum de ressources pertinentes afin d'être « *compétent* » en situation. Dans ce cas, l'apport du projet personnel de l'élève n'est pas du tout négligeable, mais comment le motiver ? C'est là, que le monde de son environnement (côté affectif et social) rentre en jeu. Le résultat d'une étude visant à analyser la façon dont fonctionnent les sujets en situation (Jonnaert, 2009), justifie qu'une compétence ne peut se produire qu'en situation. Pour permettre aux élèves de dégager leur compétence dans l'exécution d'une tâche d'apprentissage, l'enseignant doit présenter les éléments qui caractérisent le **contenu de la tâche**. « *Une des fonctions de l'enseignante ou de l'enseignant est de mettre les élèves en situation pour qu'ils se construisent des connaissances à propos des savoirs*

codifiés dans les *programmes d'études*»(Jonnaert, 2009). Par conséquent, il doit y avoir une ligne directrice claire (curricula) pour l'enseignant pour pouvoir mettre en situation les élèves. Et là, l'enseignant prend le rôle de médiateur pour permettre à l'élève de développer ses connaissances en situation et doit mobiliser plusieurs types de savoirs en interaction, tels que : celui de savoir-objet d'enseignement (les programmes d'études et les manuels scolaires), ce qui nous manque énormément à Madagascar, celui des connaissances de l'élève et des représentations qu'il met en jeu et enfin celui des exigences et des contraintes de la situation (Jonnaert, 1996). Le but des situations que l'enseignant construit dans sa classe est de permettre aux élèves de réaliser des activités afin de construire leurs connaissances. L'enseignant suit les 3 phases suivantes pendant son intervention pour faciliter les apprentissages des élèves :

1. la phase pré-active pendant laquelle l'enseignant planifie les situations ou les conditions d'apprentissages ;
2. la phase interactive, il contrôle et régularise les stratégies d'apprentissages utilisées par les élèves ;
3. la phase post-active, il procède à l'évaluation des résultats et du processus d'apprentissage pour pouvoir corriger les erreurs et apporter des corrections pour le futur si les objectifs ne sont pas atteints.

Dans ce modèle, l'élève a plus de responsabilité pour l'acquisition de ses connaissances sous le contrôle de l'enseignant. Cependant, les fonctions de l'élève se décrivent par Jonnaert et V-Borghet (2003):

1. La mise en interaction de ses propres savoirs avec les savoirs à apprendre ;
2. l'adaptation de ses connaissances au savoir à apprendre ;
3. l'adaptation du savoir à apprendre à ses connaissances ;
4. la création de nouvelles connaissances en coordonnant le résultat de cette double adaptation pour répondre aux contraintes actuelles de la situation à laquelle il est confronté.

Les savoirs scolaires et les connaissances des élèves Entre les connaissances des élèves et les savoirs scolaires. Les savoirs se réfèrent ici aux savoirs codifiés et organisés dans les programmes d'études afin d'être transmis aux apprenants. En

plus des savoirs scolaires, l'auteur souligne qu'il existe aussi des savoirs que les élèves peuvent acquérir hors de l'école. Ce sont, selon lui, les savoirs quotidiens et c'est le rôle important de TIC (Cf Figure 6.5) et les environnements sociaux. Contrairement aux savoirs scolaires, les connaissances font partie du « *patrimoine* » mental de la personne. Les connaissances du sujet lui sont personnelles. Les savoirs sont extérieurs au sujet. Une fois reconstruits par les élèves, ils deviennent leurs connaissances. Les caractéristiques de ces connaissances sont (Jonnaert, 2009):

1. Elles sont construites et non transmises ;
2. Elles sont temporairement viables ;
3. Elles nécessitent une pratique réflexive et ne sont pas admises comme telles sans remise en cause ;
4. Elles sont situées dans des contextes et des situations et non décontextualisées.

Les connaissances acquises selon la vision constructiviste s'articulent autour des savoirs personnels de l'apprenant, des savoirs contenus dans les programmes d'études et des interventions de l'enseignant. Nous prenons en compte dans notre thèse la mise en place des savoirs contenus et des interventions de l'enseignant au profit de l'apprentissage de l'algorithmique en classe de seconde à Madagascar.

Selon toujours Jonnaert et V-Borghet (2003), les trois grandes dimensions du modèle socioconstructiviste Interactif (SCI) interagissent afin de faciliter les apprentissages scolaires. Ces trois dimensions situent les relations que les élèves et les maîtres entretiennent autour de l'objet à apprendre. La dimension socio explique la façon dont l'enseignant organise les activités d'apprentissage pour créer des conditions permettant aux élèves d'interagir entre eux.

Dans ce modèle, le rôle que joue l'enseignant est essentiel pour la construction des connaissances par les élèves. L'intervention du maître aide les élèves à mettre en place un dialogue entre les savoirs antérieurement construits et les savoirs en construction. En fonction des savoirs à construire, l'enseignant doit créer des **situations** capables de rendre les élèves actifs et constructifs.

De plus, l'élève doit procéder à une analyse de ses démarches et des caractéristiques des savoirs à apprendre afin de tisser des **liens de cause à effet**. C'est cette analyse, qui débouchera sur l'acquisition de connaissances, que nous avons présentée comme un élément constitutif du patrimoine cognitif du sujet. L'intervention de l'enseignant est cruciale pour faciliter cette mise en relation entre les savoirs préalables et

les nouveaux savoirs envisagés. Au cours de l'apprentissage, il doit penser aux structurants préalables⁸ qu'il doit mobiliser chez l'élève afin d'assurer sa compréhension des apprentissages futurs. Ceci montre que ce ne sont pas tous les savoirs déjà maîtrisés qui peuvent faciliter l'acquisition de nouvelles connaissances. Les conditions d'apprentissage en contexte scolaire dans le modèle socioconstructiviste interactif (SCI) : L'auteur a précisé que *ce n'est pas parce que l'élève apprend par ses propres opérations cognitives que la fonction de l'enseignant est moins déterminante*. Selon lui, *la réalisation de l'apprentissage en contexte scolaire est liée aux interventions pédagogiques de l'enseignant. Effectivement, c'est grâce à ses interventions que se créent les interactions entre les connaissances des élèves et les savoirs à apprendre* (Jonnaert, 2009). L'enseignant organise des activités d'apprentissage dans lesquelles les élèves interagissent avec lui, mais aussi avec leurs pairs.

Dans le modèle socioconstructiviste interactif, les situations d'apprentissage, les connaissances préalables de l'élève se trouvent en conflit avec les savoirs codifiés. Ces situations génèrent une interaction entre les connaissances anciennes de l'élève, les savoirs véhiculés dans les manuels, le contexte physique et social de l'école. Mais quel est l'apport de l'interaction entre l'enseignant et l'élève dans les acquisitions de connaissances ? L'auteur présente les trois ancrages dans lesquels se construisent les connaissances en contexte social d'apprentissage :

1. Celui des interactions sociales qui provoquent les conflits sociocognitifs inter [individuel] et intra-individuel ;
2. Celui de l'inscription de l'école et des savoirs codifiés dans l'environnement social finalisé ;
3. Celui, éthique, de la responsabilité de l'apprenant face aux connaissances qu'il construit, qui sont nécessairement les siennes.

Quand les élèves travaillent en coopération i.e. en groupe (Cf. Figure 6.6), ils dégagent une critique mutuelle, ce qui rend plus objectives leurs réflexions. Le travail en équipe a non seulement des retombées positives sur le développement cognitif des élèves, mais il facilite aussi le développement de leurs habiletés sociales. En combinant les deux tétraèdres 6.5, 6.6, nous avons le pentaèdre didactique suivant (Cf. Figure 6.8) :

8. Le contenu des structurants préalables est essentiellement constitué de notions préalables à l'apprentissage envisagé.

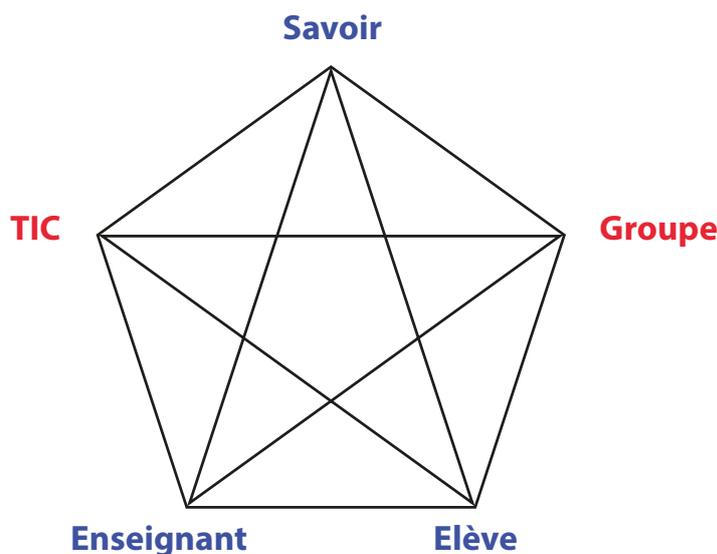


Figure 6.8 – Pentaèdre didactique

Le modèle constructiviste donne une signification à l'activité avec le concept d'équilibration qui définit les rapports du sujet au réel dans la construction des connaissances et les structures mentales. Ceci se fait par un va et vient entre « *l'assimilation du réel aux schèmes du sujet et l'accommodation de ces schèmes au réel* ».

En tenant compte du rôle que joue l'apprenant dans l'installation de la relation entre situation et réponse, on parle d'un apprentissage par construction de la réponse. D'où la théorie constructiviste prônée par Piaget, Bruner, Vygotsky .

En effet, les enfants apprennent en modifiant de vieilles idées, plutôt qu'en accumulant des parcelles supplémentaires d'information.

Compte tenu de cette approche cognitive, non seulement le sujet devient l'acteur de son propre progrès vers le savoir, mais l'objectif de l'enseignement n'est plus d'abord et seulement l'acquisition du savoir, mais la construction des compétences cognitives (savoir-faire) puisqu'il s'agit désormais « *d'apprendre à apprendre* ».

Ainsi on assiste au déplacement de l'intérêt des « structures » aux « procédures » et on s'intéresse plus spécialement à l'étude du fonctionnement cognitif du sujet en situation et la mise en œuvre de sa stratégie de résolution.

D'où le rôle de médiation de l'enseignement . Cette médiation est bien nécessaire car, tout en impliquant le sujet dans sa propre formation, l'apprentissage n'est ni solitaire, ni autonome . Le rôle joué par le médiateur est important du fait que

l'apprentissage ne se réduit pas à un simple passage du réel à la connaissance, mais « *il met en œuvre un ensemble complexe de processus mentaux qui ne s'exercent pas directement sur le réel* ». D'où l'importance accordée à celles-ci quand on décide de se préoccuper davantage de l'activité intellectuelle plutôt que du résultat.

Il existe plusieurs façons d'aborder les problèmes liés au processus d'enseignement/aprentissage de savoirs et, beaucoup de courants théoriques ont été élaborés pour éclairer des situations éducatives orientées dans ce sens. L'approche socioconstructiviste une forme de révolution paradigmatique dans le monde de l'enseignement/aprentissage scolaire, justifiée par le fait que les connaissances ne sont pas transmises mais sont construites par le sujet à travers les expériences vécues dans son milieu. L'enseignement de l'informatique, pour mettre leurs élèves ou étudiants en activité, se situe dans une approche plutôt socioconstructiviste : l'approche par projets et le travail en groupes fait référence à ce modèle. Le cadre théorique qui a guidé notre regard et notre réflexion est la théorie de l'activité (TA), que nous abordons après. Ce choix a été motivé par le fait qu'avec le modèle théorique socioconstructiviste correspondant à l'enseignement de l'informatique, les élèves, les étudiants et leurs professeurs sont tous inscrits dans un système d'activités. Malgré des rôles respectifs, ils sont appelés à entrer en interaction tout en respectant certaines règles en vue de la production d'un résultat attendu qui est l'objectif des projets. Pour toutes ces raisons, la théorie de l'activité nous semble bien appropriée pour cette étude qui, rappelons-le, porte sur les pratiques d'apprentissages des savoirs informatiques chez les élèves ou étudiants débutants dans un contexte d'apprentissage par projets.

6.3.2.2 Approche par projets et interactions

Le public cible, ce sont les élèves en classe de seconde au lycée catholique d'Antsirana débutants en informatique. Les éléments qui interviennent dans les situations d'apprentissage sont : les élèves, le professeur, le TIC et le savoir, auxquels s'ajoutent leurs interactions liées au contexte de projets effectués en groupe (Cf. Figure 6.8). C'est une approche didactique, car elle concerne la construction de savoirs disciplinaires, essentiellement informatiques et plus particulièrement algorithmiques par l'informatique, i.e par l'intermédiaire de TIC. Ceci a pour but de clarifier le rapport de l'apprenant (élève) à l'objet de connaissance (Brun, 1994) dans un contexte plus vaste de l'approche par projets, et apporter un éclairage sur les pratiques des

élèves, de leur professeur et les interactions en jeu. L'approche par projets est un apprentissage pluridisciplinaire inscrit sur un processus de longue durée. D'autre part, le travail se fait en groupe qui favorise la coopération et soutien mutuel des élèves dans leurs apprentissages. L'apprentissage coopératif suit les principes : une interdépendance positive, une responsabilité individuelle, un développement des habilités coopératives et une objectivation qui est un bilan-synthèse des apprentissages fait par l'enseignant pour terminer l'activité (Robitaille, 2007). Malgré, la grande responsabilité prise par les élèves, cela n'empêche d'avoir un tuteur en tant que chef d'orchestre (le professeur). Il prend le rôle central dans le travail de projets : il prescrit, conseille, oriente, aide en cas de nécessité. Lorsque le problème est complexe, l'engagement de l'enseignant est nécessaire, ne serait-ce que soutenir et encourager les élèves à être toujours motivés jusqu'à la fin du projet.

6.3.2.3 Approche par projets et théorie de l'activité

Au cours d'une action didactique, les pratiques des élèves et des enseignants s'influencent mutuellement (Robert et Rogalski, 2002; Rogalski, 2012). L'enseignant doit détecter les points forts et les points faibles des élèves pour construire les situations didactiques : détermination du sujet de projet, la façon de tenir son encadrement, la nature des aides fournies. . . Dans ses pratiques, le professeur doit prendre en compte les capacités des élèves, les questions posées par eux-mêmes, les tâches déjà étudiées et effectuées, les compétences des élèves à aller plus loin, leurs interactions mutuelles et leurs difficultés. . . La théorie de l'activité a alors sa place dans cette approche, car d'après Lewis (1998): « *La théorie de l'activité met en évidence les caractéristiques des communautés de travail efficaces* », mais cela peut servir dans le cadre de l'analyse des pratiques pour un travail relatif aux communautés d'apprentissages. Dans la communauté de travail à laquelle s'intéresse la théorie de l'activité, les différents membres peuvent travailler sur des tâches différentes en termes d'exigences en fonction de leurs compétences ou leur disponibilité. Dans la communauté d'apprentissage (en groupes d'élèves/étudiants), les tâches sont en général partagées équitablement au sein des membres de chaque groupe. Pour que cette approche soit efficace dans le contexte d'apprentissage, les tâches proposées doivent avoir les caractéristiques suivantes (Lewis, 1998):

1. amener les apprenants à s'y reconnaître pour susciter leur motivation ;

2. nécessiter une approche collaborative/coopérative dans leur exécution pour stimuler les interactions entre les apprenants.

Dans ce cas, l'apprentissage de l'informatique en contexte de projets, les tâches prescrites aux élèves/étudiants remplissent-elles les deux caractéristiques de Lewis pour justifier le choix de ce cadre théorique ?

6.3.3 Notion et concept d'activité

6.3.3.1 Définition d'une activité

Les travaux de Engeström (1987), de Nardi (1996) et de Kuutti (1996) inspirés par des psychologues soviétiques tels que Leontiev (1978) et Luria (1976) disciples de Vygotsky vont particulièrement dans le sens d'une considération de l'activité humaine selon une approche systémique. Ces recherches, dont l'origine se trouve dans les travaux de Vygotsky (1934), visent à étudier la nature des interactions entre l'ensemble des éléments constitutifs de l'activité humaine. Le paradigme théorique dans lequel se placent ces recherches est communément appelé « *la théorie de l'activité* ». La Théorie de l'Activité (TA) prend ses sources dans l'école russe de psychologie du développement humain à partir de la thèse historico-culturelle défendue par L.Vygotski basée sur une théorie anthropo-psychologique dans le contexte du matérialisme dialectique et du marxisme, et diffusée en Europe dans le courant des années 60 : l'apprentissage, avant d'être le fait d'individus isolés, serait d'abord un phénomène social puisqu'il se déroule dans des contextes culturellement déterminés imprimant leurs marques spécifiques aux groupes humains dans leurs comportements les plus quotidiens. Le type fondamental de contexte est appelé une activité, qui est l'unité de base du développement et de la vie humaine, et donc forme la base de l'étude de toute contextualité. L'activité humaine s'y organise et s'y déploie au moyen des outils dont disposent les individus, qu'il s'agisse du langage ou de tout autre artefact présent dans les milieux où ils évoluent. L'activité est une caractéristique particulière et vitale de l'homme lui permettant en tant qu'actant à être capable d'initier des projets d'action pour satisfaire ses besoins sur le monde réel. Leontiev (1975) définit l'activité comme étant « *une organisation systémique permettant le développement de la conscience* ». Pour comprendre l'activité, il préconise une approche de sa structuration en trois niveaux : l'activité proprement dite, les

actions associées à sa conduite et les opérations exécutées. Mais leurs frontières entre eux sont difficiles à identifier : elles restent floues et mouvantes ([?Beauné, 2010](#)).

[Leontiev \(1981\)](#), l'un des fondateurs de la théorie de l'action, distingue trois niveaux d'activité : les **activités** proprement dites, les **actions** et les **opérations**.

Les **activités** sont en relation étroite avec un *but conscient*, une *motivation* et peuvent donner lieu à une multiplicité d'actions. « *An activity is a form of doing directed to an object, and activities are distinguished from each other according to their object* ». [Une activité est une forme de faire dirigée vers un objet, et les activités se distinguent les unes des autres en fonction de leur objet].([Kuutti, 1996](#)).

Les **actions** s'effectuent par des **opérations** qui sont des *procédures compilées* et *inconscientes*. Une action peut servir plusieurs activités (faire cuire de l'eau peut servir à faire du thé, faire des inhalations pour guérir d'une grippe, pour faire cuire des oeufs etc ...). « *Actions have both an intentional, orientational aspect (what is to be done) and an instrumental aspect (how to do it : anticipated plan and general method to reach intermediary goals)* », [Les actions ont à la fois un aspect intentionnel, orientationnel (ce qui doit être fait) et un aspect instrumental (comment le faire : plan anticipé et méthode générale pour atteindre les objectifs intermédiaires)]([Linard, 1994](#)).

Les actions et opérations sont dans une relation dynamique qui permet à une action de devenir une opération. Une activité est associée à un motif, une action à un but et une opération à des conditions nécessaires à son exécution. « *Initially each operation is a conscious action, consisting of both the orientation and execution phases, but when the corresponding model is good enough and the action has been practiced long enough, the orientation phase will fade and the action will be collapsed into an operation* ». [Initialement, chaque opération est une action consciente, comprenant à la fois les phases d'orientation et d'exécution, mais lorsque le modèle correspondant est assez bon et que l'action a été pratiquée assez longtemps, la phase d'orientation s'estompera et l'action sera effondrée].([Kuutti, 1996](#))

Au fur et à mesure que des actions deviennent des opérations, le sujet peut s'occuper d'actions de plus haut niveau. Lorsque les conditions d'exécution d'une opération ont changé, celle-ci peut à nouveau obtenir le statut d'action pour être spécialisée et adaptée à ces nouvelles conditions.

Une activité possède alors un **objet** (motive) qui permet la distinction entre ac-

tivités (clause d'existence?), un **sujet** *individuel* ou *collectif* (non compréhension de l'objet=aliénation). C'est un phénomène collectif, de développement historique. L'activité existe dans un environnement **matériel** et le **transforme**.

6.3.3.2 Modélisation de l'activité

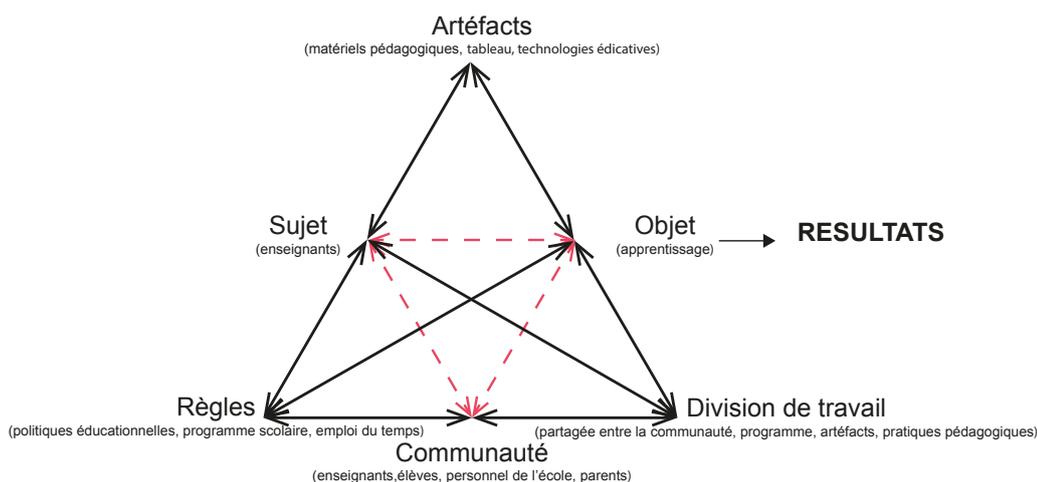


Figure 6.9 – Les éléments d'un système d'activité dans le milieu scolaire

Dans ce modèle (Cf. Figure 6.9), le sujet réfère à l'acteur (ou au groupe d'acteurs) impliqué(s) dans l'activité et dont on prend le point de vue. L'objet constitue la cible, la visée de l'activité et l'oriente en vue de l'obtention d'un certain résultat. Pour y parvenir, l'acteur a à sa disposition des instruments, artefacts matériels ou symboliques (dont le langage), qui vont assurer une médiation entre le sujet et l'objet de l'activité. La transformation de l'objet de l'activité en résultat constitue le moteur (mobile) de l'activité. Dans cette perspective, le sujet est rarement seul, il fait partie de la communauté de personnes dont l'activité a un objet identique (communauté d'intérêts, de pratiques ou de culture). Aussi, Engeström propose deux autres formes de médiation : d'une part les règles explicites ou tacites (normes, conventions, procédures administratives, pratiques de travail, relations sociales, qui constituent un « *héritage culturel* » propre à la communauté) médient les interactions entre sujet et communauté ; d'autre part, la division du travail (organisation que la communauté se donne pour satisfaire l'objet de l'activité en distribuant les rôles et les responsabilités) médie les relations entre sujet et objet de l'activité. Une telle représentation de l'activité permet de saisir simultanément le travail d'un individu ou d'un collectif

restreint et son inscription dans une organisation.

L'ensemble de ces composantes et leurs interactions constituent un système d'activité, « *which is taken as the prime unit of analysis against which scripted strings of goal-oriented actions and automatic operations are interpreted* » [qui est considérée comme l'unité d'analyse principale par rapport à laquelle les chaînes d'actions orientées vers les objectifs et les opérations automatiques sont interprétées] (Engeström, 2000a). Considéré donc comme unité d'analyse de l'activité (?), ce système « *connects the psychological, cultural and institutional perspective to analysis. The study of activity ceases to be psychology of an individual but instead focuses on the interaction between an individual, systems of artefacts and other individuals in historically developing institutional settings* » [relie la perspective psychologique, culturelle et institutionnelle à l'analyse. L'étude de l'activité cesse d'être la psychologie d'un individu, mais se concentre plutôt sur l'interaction entre un individu, des systèmes d'artefacts et d'autres individus dans des contextes institutionnels en développement historique] (Miettinen, 1997). Ces différents éléments s'influencent les uns les autres, font l'objet de contradictions internes, entrent éventuellement en contradiction les uns avec les autres ou avec d'autres systèmes d'activité, amenant continuellement ajustements et évolutions. Le système d'activité est donc dynamique et permet le développement des acteurs et des pratiques : « *the systemic contradictions, manifested in disturbances and mundane innovations, offer possibilities for expansive developmental transformation* » [les contradictions systémiques, manifestées dans les perturbations et les innovations mondaines, offrent des possibilités de transformation expansive du développement] (Engeström, 2000a). Les systèmes d'activité ne sont pas isolés les uns des autres et pour mieux comprendre ce qu'il se passe au sein de l'un d'entre eux, on peut examiner les interactions qu'il développe avec ses voisins (Engeström, 2005). Cet examen apporte souvent un éclairage complémentaire mais devient indispensable quand plusieurs systèmes sont associés dans un travail identifié comme conjoint 6. Pour comprendre ce qui se joue lors des interactions entre ces systèmes, Engeström développe trois concepts principaux. Le « *boundary crossing* » [franchissement de frontière] (Engeström et Karkkainen, 1995) recouvre les processus en jeu dans ces interactions et pourrait être analysé « *as a process of collective concept formation* » [comme un processus de formation de concept collectif]. Deux de ces processus, le « *knotworking* » [travail en noeud]

et la « *co-configuration* », ont fait l'objet d'analyses. « *The notion of knot refers to rapidly pulsating, distributed and partially improvised orchestration of collaborative performance between otherwise loosely connected actors and activity system* » [La notion de nœud fait référence à une orchestration rapide, distribuée et partiellement improvisée de la performance collaborative entre des acteurs autrement connectés de manière lâche et un système d'activité] (Engeström, 2000a). Dans ce cadre, les fils de l'activité sont noués, dénoués, renoués, le centre de l'initiative évoluant au cours du temps et ne pouvant être rattaché à un individu ou un groupe particulier. L'objectif est ici de résoudre un problème temporaire, les frontières entre les systèmes étant franchies de manière fugace même si la collaboration est inscrite dans le temps. La co-configuration (Engeström, 2000b) constitue une nouvelle forme de travail dans laquelle un fabricant développe dans la durée un partenariat interactif avec ses fournisseurs et ses clients pour concevoir, en les co-configurant, des produits ou des services complexes qui évoluent régulièrement dans le temps. Selon Engeström (2000a), « *knotworking may be seen as the emerging interactional core of co-configuration* » [Le travail en nœud peut être vu comme le noyau interactionnel émergent de la co-configuration].

Dans la conception dudit système, le sujet et l'objet sont situés en son cœur. Si l'activité est orientée par son **objet**, le rapport à cet objet est médiatisé par des **artefacts** (instruments) et participe au développement d'une activité socialement réglée et normée. Au sein du système, le **sujet** fait partie d'une **communauté** qui partage avec lui le même objet d'activité. Les relations **communauté-sujet** et **communauté-objet** sont médiatisées par les concepts de **règles** et de divisions du travail véhiculant eux aussi un héritage culturel de la situation (?). Ainsi Engeström donne la définition d'un système d'activités comme étant « *un système cohérent du point de vue de l'activité orientée vers des objets, des sujets agissant par l'intermédiaire des artefacts, et organisés collectivement au sein d'une division du travail, des règles qu'ils emploient et de leur communauté (d'intérêts, de pratiques ou de cultures)* » (Owen, 2008). La Figure 6.10 modélise le schéma actuel de la structure de base d'un système d'activités selon la conception d'Engeström.

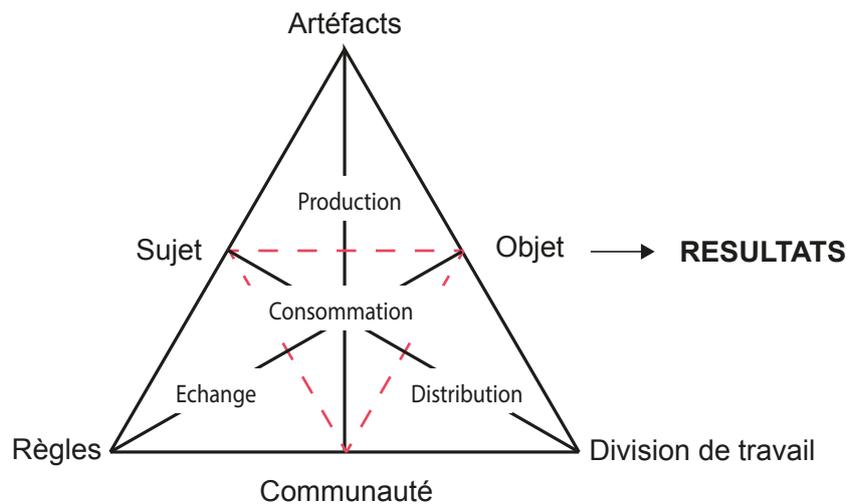


Figure 6.10 – Structure de base de l'activité humaine d'Engeström (1987)

La génération de la TA d'Engeström se synthétise en cinq principes (Barma, 2010):

1. Le système d'activité constitue l'unité d'analyse. Orienté vers l'objet d'étude, ce sont les artefacts qui assurent la médiatisation dans son rapport avec l'objet. Les systèmes d'activités sont inter-reliés : chacun est en interrelation avec au moins un des autres systèmes ;
2. Un système d'activité reflète divers points de vue, divers traditions et intérêts. Cette situation est rendue possible par la division du travail au sein du système, ce qui donne à chaque membre du système un rôle à jouer ;
3. Les systèmes d'activités ne restent pas figés une fois formés : ils se renouvellent et leurs transformations peuvent durer de longues périodes ;
4. Il existe des tensions au sein d'un système d'activités. Souvent de nature structurelle, ces tensions, accumulées depuis longtemps au sein des systèmes d'activités, jouent un rôle essentiel dans la transformation et l'innovation ;
5. Dans un système d'activités, l'innovation se produit lorsqu'il y a une transformation, c'est-à-dire qu'il y a reconceptualisation de l'objet et du motif de l'activité pour un enjeu plus large comparativement au système d'activité d'avant.

6.3.3.3 Médiations dans le système de l'activité

La recherche de la finalité de l'activité par l'individu est influencée par les autres éléments du système de l'activité. Nous parlons de « médiation » lorsqu'un élément du système de l'activité vient faciliter l'accès de l'individu à la finalité de l'activité. Selon le modèle d'Engeström, les différents types de médiations pouvant survenir dans le système de l'activité d'apprentissage sont les suivants :

- la médiation par les instruments (artefacts) ;
- la médiation par les communautés ;
- la médiation par les règles ;
- la médiation par la division du travail.

Par « instruments », il faut entendre non seulement les instruments de nature matérielle, mais aussi ceux qui ont une nature psychologique.

Par « communauté », il faut entendre tous les individus liés à l'activité, constitués ou non en groupes et/ou en sous groupes.

La médiation par les « règles », se fait par les rapports entre le sujet et l'objet directement et indirectement. En effet, les règles définissent, d'une part, la façon dont le sujet peut accéder à l'objet et, d'autre part, la façon dont le sujet peut se comporter à l'intérieur des communautés sociales liées à la réalisation de l'activité.

Quant à la médiation par la « division du travail », elle s'opère à deux niveaux : individuel et collectif. Au niveau individuel, elle permet au sujet de diviser l'activité nécessaire pour atteindre l'objet en une série d'actions constituées elles-mêmes d'opérations élémentaires et de rendre ainsi réalisable une activité complexe. Au niveau collectif, elle permet au sujet de bénéficier des actions effectuées par d'autres membres de la communauté dont il est membre dans le sens des résultats attendus de l'activité.

6.3.3.4 La notion de contradiction

Les systèmes d'activité sont caractérisés par des contradictions (ou tensions internes) qui peuvent favoriser et déclencher l'innovation et le changement et peuvent être source de développement. Le terme de « *contradiction* » ne doit pas être compris en tant que problème, obstacle ou conflit, mais plutôt, comme des « tensions accumulées historiquement à l'intérieur d'un système d'activités ou entre différents systèmes

d'activités » (Engeström, 2001). Ces tensions indiquent « *un manque de compatibilité à l'intérieur des éléments, entre les éléments, entre différentes activités et entre différentes phases d'une seule activité* ». Dans le contexte des pratiques de travail, les contradictions « *se manifestent comme des problèmes, des ruptures, des pannes* » (Kuutti, 1996) ou comme des perturbations qui interrompent le flux normal du travail.

Dans (Engeström, 1987), il distingue 4 types de contradictions :

1. Contradictions dans chaque noeud de l'activité centrale : par exemple, il peut y avoir des contradictions entre les différentes communautés en jeu dans une activité, entre les différentes règles en vigueur ou encore entre les différents instruments mis en œuvre dans le cadre de l'activité. Le « *conflit instrumental* » dans des situations d'enseignement/apprentissage au sein d'institutions éducatives relève de ce type de contradictions.
2. Contradictions entre les nœuds constituant l'activité centrale : par exemple, l'adoption d'environnements virtuels d'apprentissage au sein d'institutions éducatives peut créer des tensions entre les nouvelles démarches d'enseignement/apprentissage liées à ces environnements et les rôles traditionnels des apprenants et des enseignants. Les règles et la division du travail traditionnellement en vigueur peuvent ne plus correspondre aux spécificités des instruments techniques utilisés pour faciliter l'apprentissage.
3. Contradictions entre l'objet de la forme dominante de l'activité centrale et l'objet d'une forme culturellement plus avancée de l'activité centrale : le décalage entre les connaissances acquises dans le cadre d'institutions éducatives et les connaissances nécessaires pour agir dans les contextes réels du travail et de la vie quotidienne constitue un exemple de ce type de contradictions. Alors que les activités d'apprentissage à l'école et à l'université visent traditionnellement la maîtrise de contenus disciplinaires de nature « savante », l'objet de l'activité d'apprentissage devient aujourd'hui de plus en plus l'acquisition de compétences exploitables dans des situations professionnelles ou de la vie quotidienne au sein de la société. Ce changement de l'objet de l'activité d'apprentissage crée des contradictions et des tensions qu'il est nécessaire de résoudre en repensant le système de l'activité d'apprentissage au sein d'institutions éducatives.

4. Contradictions entre l'activité centrale et des activités voisines : par exemple, dans une institution éducative, un apprenant qui a passé un premier niveau d'études avec succès pourrait ressentir des difficultés à s'adapter à la différence des méthodes éducatives mises en œuvre au niveau supérieur. Les enseignants pourraient considérer que les activités d'apprentissage effectuées au niveau précédant par d'autres enseignants n'ont pas pu préparer le sujet (l'apprenant) aux activités d'apprentissage auxquelles il est censé prendre part. Il s'agit ici d'une contradiction entre plusieurs activités d'apprentissage.

Ces contradictions produisent des déséquilibres adaptatifs dans le système de l'activité. Ce dernier est en effet dans une dynamique permanente de recherche d'équilibre en surmontant les tensions et les perturbations qui se produisent suite aux contradictions internes (Taurisson, 2005). La recherche d'équilibre par les acteurs de l'activité est source d'innovations et d'améliorations et par conséquent source d'apprentissages. Elle conduit à modifier la structure de l'activité ou certains de ses composantes (Taurisson, 2005; Meyers, 2007). Cette évolution se fait par l'intégration de nouveaux concepts et de nouveaux savoir-faire dans le système de l'activité. Engeström (1987, 2001) appelle ce type d'apprentissage, « *apprentissage par expansion* ».

La compréhension de ces tensions est essentielle pour appréhender les difficultés des enseignants lorsqu'ils doivent intégrer les TIC dans leurs pratiques courantes. Parmi toutes les contradictions théoriques possibles et après une revue bibliographique assez large, Bonneau (2011) considère que 5 d'entre elles sont déterminantes dans la sphère scolaire (Cf. Figure 6.11) :

- Contradiction entre l'artefact et la communauté ;
- Contradiction entre les sujets et les règles du système d'activité ;
- Contradiction à l'intérieur de l'artefact ;
- Contradictions entre l'artefact et l'objet d'activité ;
- Contradiction dans la division du travail.

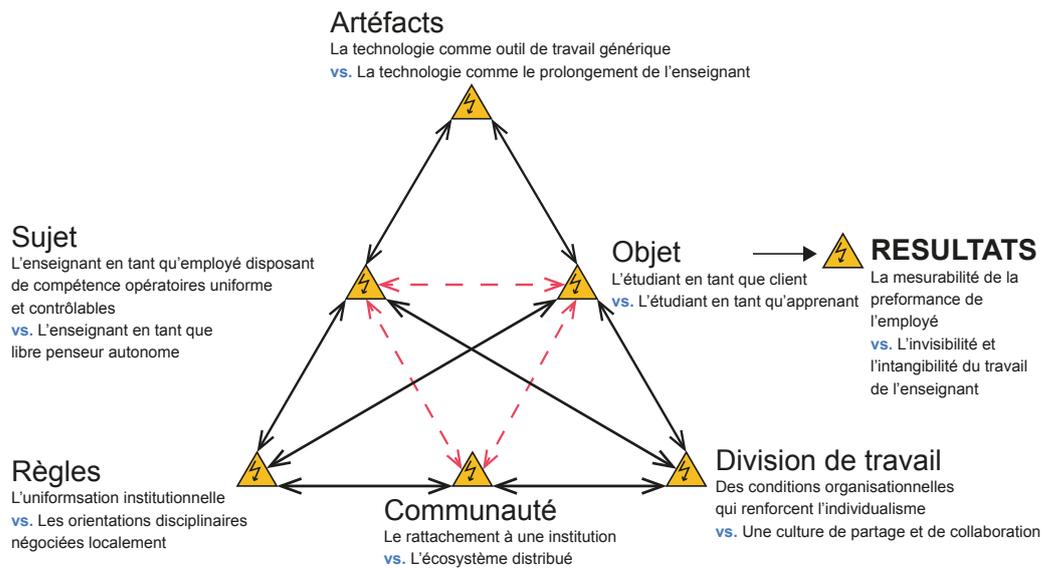


Figure 6.11 – La contradiction primaire dans chaque constituante du système d'activité, (*un extrait emprunté à C Bonneau p.27*)

Les contradictions peuvent se présenter entre deux éléments d'un système d'activité ou à l'intérieur d'un élément (Cf. Figure 6.11). Les enseignants peuvent rencontrer des contradictions entre les pratiques éducationnelles véhiculées par les outils traditionnels et les pratiques véhiculées par les TIC (contradiction entre l'artefact et l'objet du travail). Par exemple, les enseignants souhaitent que les enfants continuent à copier leurs devoirs (valorisation de l'autonomie et de l'effort), chose qui n'est plus nécessaire avec l'arrivée des cahiers de textes numériques (Voulgre, 2011). Les contradictions peuvent également se manifester à l'intérieur d'un élément du système d'activité. Certains auteurs (Blin et Munro, 2008) parlent des différences entre les représentations du rôle d'enseignant véhiculées par les technologies et celles véhiculées par la tradition scolaire. L'enseignant joue traditionnellement un rôle actif, de transmetteur d'information « *sage on the stage* » [sage sur la scène], alors que les technologies de l'information lui confèrent un rôle d'accompagnateur des élèves sur l'ordinateur (l'outil) « *guide on the side* » [guide sur le côté] (Chaptal, 2007). Il existe alors une contradiction dans la division du travail du système scolaire (Cf. Figure 6.11).

L'étude de ces contradictions aide à révéler les différences qui peuvent exister entre les politiques éducationnelles et les réalités concrètes de l'école, avec les enseignants qui doivent faire face à ces injonctions. Ainsi, l'analyse de ces contradictions est utile

pour identifier les obstacles et les freins dans l'adoption des TIC en classe et pour comprendre certains phénomènes de résistance au changement.

6.3.4 Modèle adopté dans le système d'activités

Nous montrons ci-dessous notre modèle des pratiques expérimentales liées à l'enseignement de l'algorithmique au sein du curriculum lycéen malagasy en informatique. En effet, les activités expérimentales apparaissent comme un paradigme organisateur de l'enseignement de l'algorithmique (Host, 1991). Ces activités peuvent prendre des formes très diversifiées (Coquidé, 2000) : des activités d'investigation qui offrent aux élèves de véritables moments d'expérimentation scientifique avec un réel non trop aménagé, des expériences réalisées par l'enseignant, évocation d'expériences historiques, pratiques réalisées par les élèves, etc. Coquidé (2000) a signalé que, tout au long du cursus scolaire, le rapport expérimental avec la réalité répond à plusieurs fonctions. Construire un rapport expérimental avec la réalité répond ainsi à des fonctions de scientificité. Il permet, en effet, de poser et de résoudre des problèmes. Il permet aussi de constituer un référent empirique (Martinand, 1996) dans l'élaboration des concepts et des modèles scientifiques, et il contribue à la construction d'une rationalité expérimentale. Les élèves participent aux expériences sous forme de groupes en utilisant l'outil TIC pour mieux conserver les logiques de raisonnement de résolution de problèmes posés. Cela entraîne un certain conflit d'idées afin d'en construire une nouvelle base de connaissances, avec l'aide du Professeur en tant que régulateur de séance, qui partage les tâches constituant un projet d'apprentissage pour atteindre un objectif précis pour chaque groupe (CF. Figure 6.12).

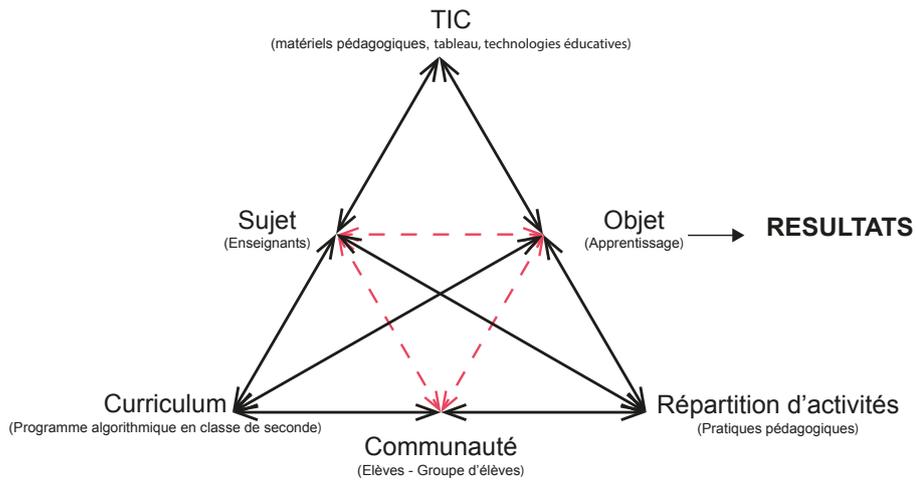


Figure 6.12 – Le modèle didactique associé à l'activité algorithmique en classe de seconde

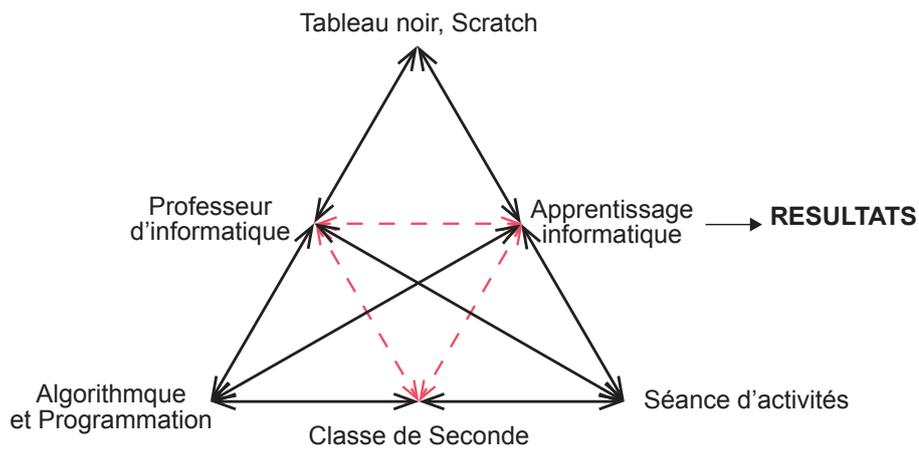


Figure 6.13 – Le modèle réduit associé à l'activité algorithmique en classe de seconde

Chapitre 7

La technologie de l'éducation et les technologies de l'information et de la communication

“ Il est hélas devenu évident aujourd’hui que notre technologie a dépassé notre humanité. ”

Albert Einstein

Sommaire

7.1 Introduction	200
7.2 La technologie de l'éducation	200
7.2.1 Technologie	200
7.2.2 Technologie éducative	201
7.2.3 Technologie de l'éducation	202
7.3 Les Technologies de l'Information et de la Communication	205
7.4 Étude sur les stratégies à mettre en œuvre pour l'intégration effective des TIC dans la pratique pédagogique de la région DIANA	206
7.4.1 Matériels et méthodes	208
7.4.2 Résultats et discussions	211
7.4.3 Stratégie à mettre en œuvre	223
7.4.4 Conclusion partielle	224
7.5 Analyse et diagnostic sur la connaissance en informatique des étudiants de l'enseignement supérieur à Madagascar	224
7.5.1 Matériels et méthodes utilisés pour la fouille des données	225

7.5.2 Interprétation de quelques règles	232
7.6 Conclusion	233

7.1 Introduction

Les technologies de l'information et de la communication (TIC) prennent toujours leur part inévitable dans l'amélioration de l'éducation. L'UNESCO étant un organisme international dans le domaine de l'éducation, ne ménage guère ses efforts à faire comprendre les acteurs internationaux et les pays membres, le rôle majeur des TIC pour atteindre l'Objectif de développement durable ODD N°4 jusqu'en 2030, dont l'objectif est d'avoir un accès à tous une éducation de qualité, sur un pied d'égalité, et promouvoir les possibilités d'apprentissage tout au long de sa vie. Des attributions de prix sont aperçues sur les programmes innovants pour l'insertion des TIC dans l'éducation. Des projets d'appui à la réforme éducatif en insérant la technologie de l'éducation sont en pompe. A Madagascar, où nous sommes ?. Un état de lieu global s'avère nécessaire.

7.2 La technologie de l'éducation

7.2.1 Technologie

La technologie est l'élaboration et le perfectionnement des méthodes permettant l'utilisation efficace des techniques diverses prises isolément, en groupe ou dans leur ensemble, qu'il s'agisse de techniques ou mécaniques, physiques ou intellectuelles, en vue d'assurer le fonctionnement des mécanismes de la production, de la consommation, de l'information, de la communication, des loisirs, de la construction et de la destruction, ainsi que des activités de la recherche artistique et scientifique. On distingue trois technologies :

1. la technologie instrumentale qui se développe depuis 3 millions d'années. Elle va des outils les plus simples jusqu'aux microprocesseurs ;
2. la technologie sociale qui se manifeste par l'insertion d'idées ou d'objets dans le circuit social, sur les plans financiers, industriels et politiques, par l'intermé-

diaire de ceux qui représentent les différents secteurs et les différents groupes de pression ;

3. la technologie économique qui concerne les principes et les techniques de rentabilisation des investissements, ainsi que les études théoriques et leur mise au point pratique par des techniciens à la disposition des décideurs.

« *La technologie est l'application de la connaissance aux buts de la vie humaine, ou de changer et manipuler l'environnement de l'homme.* »¹ Le mot technologie est employé pour dénoter les sens suivants :

- L'utilisation des outils et matériels issus de l'application de la technologie ;
- L'application du savoir pour créer les outils et pour faciliter la vie ;
- Les techniques, les méthodes, les procédures et les compétences utilisées pour augmenter la productivité, rendre les systèmes d'organisation plus efficaces et la vie plus aisée ;
- La manipulation des sources de l'énergie pour rendre la vie plus aisée.

7.2.2 Technologie éducative

La technologie éducative est l'intégration de la technologie dans l'enseignement et l'emploi des méthodes pédagogiques et organisationnelles plus innovées pour améliorer la qualité de l'enseignement. Elle implique l'application des théories de l'apprentissage, le changement des pratiques pédagogiques, la restructuration du système d'enseignement et l'utilisation de la technologie pour des fins pédagogiques, scolaires et académiques. Pour cela, la technologie éducative est l'organisation systématique des activités de la formation et de l'apprentissage à travers l'application des connaissances, l'innovation des techniques et l'intégration des TIC pour améliorer la qualité de l'enseignement ou combler les faiblesses de l'enseignement classique.

Dans les documentations de la technologie de l'enseignement : technologie éducative = technologies pour l'apprentissage = technologie de l'enseignement.

D'autres auteurs ont défini la technologie éducative comme suit : « *les technologies pour l'apprentissage sont des modèles spécifiques et fiables d'enseignement et d'apprentissage qui servent à accomplir un apprentissage efficace* » (Heinich, Molenda, Russek, et Smaldino, 2002). « *La technologie éducative : c'est une théorie et une*

1. Encyclopaedia Britannica (Technology, 2006).

pratique de conception, de développement, d'utilisation, de gestion, et d'évaluation des processus et ressources pour l'apprentissage »².

7.2.3 Technologie de l'éducation

Les technologies de l'éducation au sens large désignent tous les outils, artefacts qui sont mobilisés dans la situation d'enseignement (Cf. Figure 6.10). Les outils dits traditionnels sont à ce point intégrés à la situation d'enseignement qu'ils en deviennent transparents : le crayon de l'élève (!), les cahiers, le tableau noir... dont on pense d'ailleurs, sans doute à tort, que l'usage en est trivial et ne nécessite pas de formation pour les futurs enseignants. Lorsqu'un nouvel outil apparaît, il doit « trouver » sa place dans la classe, et ce à trois niveaux : les *pratiques enseignantes*, les *interactions élèves-élèves-enseignant* et les *processus d'apprentissage*. Dans ce qui suit seront détaillées pour chaque niveau les implications et les conséquences de l'introduction d'un nouvel outil. Mais parfois, la « technologie » est confondue avec « outil » ou encore avec « technique ». « *Technology means the systematic application of scientific knowledge to practical tasks* » : La technologie signifie l'application systématique des connaissances scientifiques aux tâches pratiques et « *educational technology is based on theoretical knowledge drawn from different disciplines (communication, education, psychology, sociology, philosophy, artificial intelligence, computer science, etc.) plus experiential knowledge drawn from educational practice* » : la technologie éducative est basée sur des connaissances théoriques tirées de différentes disciplines (communication, éducation, psychologie, sociologie, philosophie, intelligence artificielle, informatique, etc.) plus des connaissances expérientielles tirées de la pratique éducative (Schneider, 2003).

- **Pratiques enseignantes** : le nouvel outil doit s'intégrer dans les pratiques existantes, de manière fluide et cohérente et sans rupture physique (changement de salle...) ou pédagogique. Sur ce dernier point, il est important de réfléchir à quel format pédagogique l'outil convient le mieux (travail en petit groupe vs grand groupe, méthode par la découverte ou l'enquête vs. Maîtrise de compétence, etc.). Le fait même de proposer de jouer sur les formats pédagogiques en fonction des activités et des objectifs d'apprentissage, en dehors

2. Association pour la Communication Educative et la Technologie (AECT)

de tout nouvel outil, ne va pas de soi, pour des raisons institutionnelles et idéologiques ;

- **Interactions entre élèves et élèves-enseignant:** On oublie souvent l'effet de l'outil sur les interactions dans les situations de classe, bien que cela soit un des sujets les plus étudiés dans la recherche en technologies éducatives. La plupart des outils pour l'apprentissage ont, entre autres, une fonction de stockage de l'information et de restitution, soit à l'enseignant, soit à un groupe d'élève, soit à la classe entière. Une activité faite sur un cahier, un panneau affiché, ou au tableau noir n'a pas le même impact sur les interactions. On n'aurait pas l'idée d'utiliser une ardoise pour montrer à la classe entière, et si un groupe travaille ensemble, leur espace d'interaction doit être suffisamment vaste. La question importante ici est qui va utiliser l'outil, et pour montrer à qui ;
- **Processus d'apprentissage:** l'objectif de l'éducation ou de la formation initiale est de fournir compétences et connaissances que les apprenants soient capable de mobiliser dans les situations adéquates. D'un point de vue cognitif, un outil au service des apprentissages est un outil qui favorise les processus cognitifs que l'on souhaite développer chez les élèves, qui soutient les processus prérequis et qui réduit les obstacles à ces processus. Ainsi, l'outil tableau noir comme support à la copie d'un texte soulage la mémoire de travail, soutient les processus de perception visuelle et permet d'amenuiser les différences entre vitesse d'écriture.

7.2.3.1 Sens de la technologie de l'éducation

La technologie de l'éducation a deux sens :

- **Sens de rationalisation de l'activité de formation** par l'intermédiaire de la définition précise des objectifs, de l'élaboration des stratégies adaptées au public et au type d'apprentissage visé. « *Technologie de l'éducation* » : *préoccupation par l'efficacité de l'enseignement*
- **Sens d'utilisation des machines** pouvant aider les enseignants (tableau interactif, vidéo, Data-show, projecteur de cinéma, magnétophone, etc.). « *Technologie en éducation* » : *l'utilisation des machines et techniques*

7.2.3.2 Domaine de la technologie de l'éducation

Les recherches dans le domaine de la technologie éducative s'intéressent à :

- L'application des théories de l'enseignement pour améliorer la qualité de l'enseignement et pour résoudre les problèmes éducatifs ;
- L'application des principes de la communication et de la présentation pour assurer la création des meilleurs outils pédagogiques ;
- La découverte des meilleures façons de modeler l'environnement éducatif pour assurer un meilleur apprentissage ;
- L'imagination de la potentialité pédagogique des innovations technologiques ;
- L'analyse des besoins des écoles ou des systèmes éducatifs pour proposer des solutions pédagogiques aux problèmes éducatifs à travers la mise en œuvre de la technologie de l'enseignement ;
- La détermination des changements nécessaires dans les programmes scolaires pour une meilleure intégration de la technologie ;
- L'évaluation des obstacles qui limitent l'efficacité des programmes d'intégration de la technologie dans l'enseignement ;
- L'utilisation de la technologie pour créer des possibilités d'enseignement et d'apprentissage.

7.2.3.3 Rôles des spécialistes de la technologie éducative

Les spécialistes de la technologie de l'enseignement :

1. Font la conception pédagogique ;
2. Planifient et pilotent les programmes de formation à distance ;
3. Associent au développement des solutions d'e-formation (e- Learning) ;
4. Élaborent des curriculums et des stratégies d'apprentissage en intégrant la technologie dans l'enseignement ;
5. Travaillent comme des consultants de formation pour les organisations et les entreprises ;
6. Participent à la création des plateformes et outils pédagogiques tels que les logiciels de formation interactive sur Internet, l'enseignement assisté par ordinateur, et les audiovisuels pédagogiques ;

7. Supervisent et gèrent les programmes de mise en œuvre de la technologie de l'enseignement ;
8. De plus important, font les recherches dans le domaine de la technologie éducative pour créer des possibilités et résoudre des problèmes.

7.3 Les Technologies de l'Information et de la Communication

Les technologies de l'information et de la communication « TIC » jouent le rôle inévitable dans le développement socio-économique d'un pays. Les TIC offrent des possibilités techniques avantageuses, qui facilitent les tâches complexes face à l'explosion des connaissances mais aussi, plus largement, par la volonté de créer un développement durable pour toute la société. Notre société du XXI^{ème}, se base surtout sur le traitement et la gestion de l'information numérique. Les TIC nous accompagnent dans notre vie quotidienne, car ils ont une influence importante sur les sociétés permettant de : raccourcir les distances, d'accéder à des informations encyclopédique.

« Les technologies de l'information et de la communication renvoient à un ensemble de technologies fondées sur l'informatique, la microélectronique, les télécommunications (notamment les réseaux), le multimédia et l'audiovisuel, qui, lorsqu'elles sont combinées et interconnectées, permettent de rechercher, de stocker, de traiter et de transmettre des informations, sous forme de données de divers types (texte, son, images fixes, images vidéo, etc.), et permettent l'interactivité entre des personnes, et entre des personnes et des machines »(Basque, 2005).

En pédagogie, les TIC permettent d'apprendre, de comprendre, d'entreprendre, de motiver, de partager, d'interagir, de communiquer, d'échanger, de collaborer, d'exposer, de transmettre et de distribuer de savoir. Utiliser les TIC en éducation, c'est aussi faire passer à l'apprenant des « connaissances spontanées » aux « connaissances « manipulées », puis aux « connaissances intégrées ». Il s'agit de bien utiliser l'outil informatique car il peut générer des simulations, reconstruire des lois, des mo-

dèles et des théories. Il permet plus rapidement à l'apprenant de mettre en relation les éléments nouveaux avec les idées déjà établies dans sa propre structure cognitive, ce qui détermine l'acte d'apprendre. C'est la raison pour laquelle, l'utilisation des TIC en éducation est primordiale pour plusieurs pays souhaitant se développer. D'ailleurs, nous assistons à un engouement des pays pour l'acquérir des TIC avec une volonté de les intégrer dans l'éducation, plus particulièrement les pays en voie de développement.

L'intégration des TIC dans l'éducation a été une réussite dans des pays industrialisés, mais ceci n'a pas été une opération facile à réaliser, car elle demande la contribution de tous les acteurs de l'éducation et une forte volonté politique de l'état. Nous proposons dans cette section, une étude scientifique orientée sur la théorie d'analyse multivariée basée sur la technique d'analyse factorielle multiple sur les stratégies de mises en œuvre pour l'intégration des TIC dans l'éducation à Madagascar.

7.4 Étude sur les stratégies à mettre en œuvre pour l'intégration effective des TIC dans la pratique pédagogique de la région DIANA

L'introduction de l'informatique à Madagascar a été dominée par une politique de maîtrise de coût, de l'utilisation exclusive des compétences nationales et de la volonté de disposer d'un parc de matériels compatibles. Au-delà des politiques nationales d'informatisation plus ou moins pertinentes, ce sont aujourd'hui les conséquences de la mondialisation de l'économie soutenue par la vague de réformes économiques appliquées aux pays en voies de développement sous le conseil des institutions financières internationales qui ouvrent de nouvelles perspectives. L'ordinateur qui a été, pendant longtemps, considéré comme un produit de luxe, et taxer en conséquence, est en train de changer de statut. Car en 2004, le gouvernement malagasy a pris une décision pour la détaxation totale des matériels informatiques dans l'espoir de rattraper le retard avec cette technologie moderne quantitativement puis qualitativement. Le retard sur l'enseignement de l'informatique est d'une quarantaine d'années par rapport aux pays avancés ([Baron et Brouillard, 1996](#)), car aucun niveau était concerné sur l'utilisation de l'informatique jusqu'en 2000. On ne peut

pas parler d'informatique sans l'ordinateur en tant qu'outil basique et inévitable des technologies de l'information et de la communication (TIC). Les TIC renvoient à un ensemble de technologies fondées sur l'informatique, la microélectronique, les télécommunications (notamment les réseaux), le multimédia et l'audiovisuel, qui, lorsqu'elles sont combinées et interconnectées, permettent de rechercher, de stocker, de traiter et de transmettre des informations, sous forme de données de divers types (texte, son, images fixes, images vidéo, etc.), et permettent l'interactivité entre des personnes, et entre des personnes et des machines (Basque, 2005). Il est à noter que la TIC joue un rôle de transmission efficace et rapide d'information, qui est un message pouvant être colligé, analysé, synthétisé et transmis sous forme de données (Legendre, 1993). L'usage des TIC en tant que véhicules de messages pédagogiques évoque une crainte pour certains de voir le formateur (professeur), considéré ici comme le principal, sinon l'unique, transmetteur de connaissances, remplacé par les TIC au sein de l'acte pédagogique. Il y a encore beaucoup à faire avant que les TIC deviennent pour les professeurs, non pas des remplaçants, mais de véritables « partenaires pédagogiques » et avant qu'eux-mêmes se perçoivent comme des « professeurs plus » (Perkins, 2005). Une question se pose : Est-ce qu'un professeur plus peut-il toujours former un élève plus?. Logiquement, la réponse est « oui », si et seulement si l'émetteur de connaissances (formateur) et le récepteur (formé) sont tous en bon état, et que tous les deux fonctionnent efficacement et en synergie totale, mais dans le cas contraire, le résultat n'est pas bon : « élève moins ». Le TIC alors-t-il parmi les moyens perspicaces pour enseigner mieux et/ou pour apprendre mieux ? Il s'agit cependant de savoir comment peut-on faire pour que le formateur et le formé soient informés des outils TIC (ordinateur,...) ?, quels moyens faut-il mettre pour qu'ils aient accès à ces outils ?, que faut-il faire pour qu'ils s'intéressent à ces outils ?, comment peut-on faire pour qu'ils sachent manipuler ces outils ?, que faut-il faire pour qu'ils bénéficient de formations sur ces outils ? L'objectif est alors d'avoir une idée précise sur l'accès, le niveau de connaissance, l'intégration et l'utilisation du TIC des formateurs et/ou des formés dans les lycées techniques et à l'université d'Antsirananana ville. C'est la politique du TIC pour Tous dans la perspective de l'amélioration du système éducatif à Madagascar, plus particulièrement le système d'enseignement.

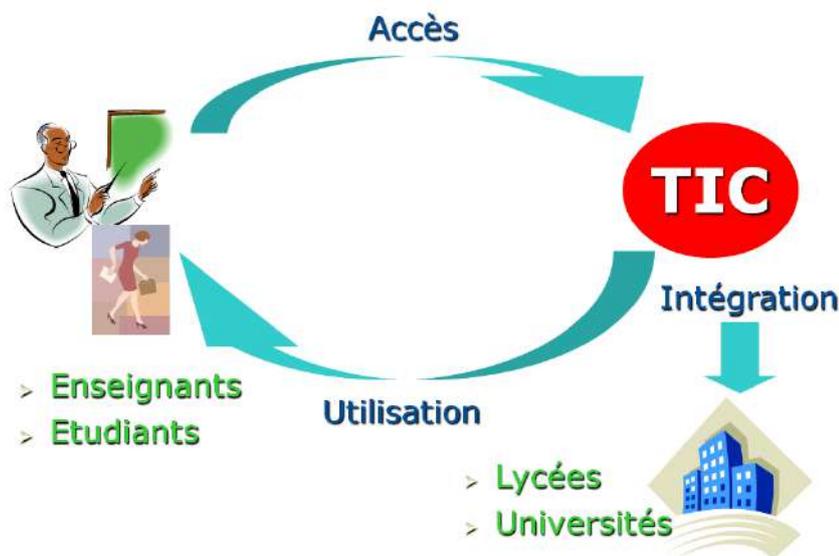


Figure 7.1 – Accès et utilisation TIC des élèves/enseignants dans les Lycées/Universités

7.4.1 Matériels et méthodes

7.4.1.1 Recueils des données

Nous avons mené des enquêtes corporelles auprès de l'Université Nord d'Antsiranana et dans quelques Lycées Publics et Privées de Diégo Suarez en vue d'élaborer une étude sur les stratégies à mettre en œuvre pour l'intégration effective des TIC dans la pratique pédagogique de l'enseignement supérieur technique de la ville. Pour cela, nous avons élaboré un modèle de questionnaire qui est constitué de 4 grandes parties dont :

- La partie qui caractérise les individus enquêtés (enseignants, étudiants et élèves) ;
- La partie sur les connaissances des enquêtés et leur utilisation de l'ordinateur et de l'internet comme composantes TIC ;
- La partie sur les connaissances et utilisation du TICE ;
- La partie sur l'opinion et le rôle du TIC dans l'enseignement et apprentissage de l'informatique.

Le tableau 7.1 illustre un extrait de la première partie :

PROFIL DE L'ENSEIGNANT	
5	Matière enseignée ?
6	Nombre d'années dans l'enseignement ?
7	Nombre d'années dans L'Etablissement ?
PROFIL DE L'ÉLÈVE/ETUDIANT	
8	Classe fréquentée ? Année
9	Série ? Filière
10	(GÉNÉRAL=1, TERTIAIRE=2, INDUSTRIEL=3, TECHNIQUE=4, TECHNOLOGIQUE=5, PROFESSIONNEL=6)

Tableau 7.1 – Extrait du questionnaire C.1: partie qui caractérise les individus enquêtés

En tout, 53 questions ont été réparties dans les quatre parties susmentionnées. Le tableau 7.2 résume la répartition des questions selon les parties.

Partie	Catégorie de questions	Nombre de questions	
I	Individus	5	
	Profil des individus	Enseignant	3
		Elève-étudiant	3
II	Connaissances et utilisation de l'ordinateur et/ou de l'internet (TIC) et formation en informatique	16	
	Formation en informatique	4	
III	Connaissances et utilisation du TICE dans l'enseignement/apprentissage	17	
IV	Place et le rôle de l'ordinateur et/ou de l'internet (TIC) dans l'apprentissage de l'informatique	5	
TOTAL DES QUESTIONS		53	

Tableau 7.2 – Répartition des questions selon les parties

7.4.1.2 Méthodes et matériels utilisés pour l'exploitation des données

Les questionnaires ont permis d'effectuer des enquêtes portées en mai 2012 sur les enseignants, les étudiants de l'Université Nord d'Antsiranana et les élèves des établissements publics et privés de Diégo Suarez concernant leur situation, soit personnelle soit universitaire, et d'une part leur connaissance sur le TIC (Rakotomalala et al., 2012). Le questionnaire était composé de 53 questions, groupées en quatre parties (Cf. tableau 7.2). 300 fiches ont été distribuées à 6 enquêteurs et nous avons eu en retour 170 questionnaires, soit 56.6% de l'ensemble.

Les versions papiers ont été numérisées en se servant de l'outil Excel pour constituer une base de données pouvant nous permettre d'effectuer les différentes analyses qui sont exposées par la suite.

7.4.1.3 Mise en œuvre des analyses multidimensionnelles

Disposant d'une masse conséquente d'informations (170 individus enquêtés dans des établissements publics et privés dans la ville d'Antsiranana), nous nous sommes orientés vers les méthodes d'analyses multidimensionnelles mises en œuvre en deux étapes successives, qui sont respectivement l'Analyse Factorielle Multiple (AFM) ainsi que la Classification Ascendante Hiérarchique (CAH).

Mise en œuvre de l'AFM

Une AFM permet de discriminer de façon optimale les unités statistiques, qui sont ici les individus ou parties des individus, sur un ensemble de variables synthétiques des thématiques étudiées (axes factoriels). Ceci est une technique de représentation des données ne nécessitant ni hypothèses ni modèles particuliers et où l'on décrit les observations à partir de variables structurées en groupes correspondant aux thématiques retenues. L'AFM est une méthode d'analyse simultanée de K tableaux. Elle est donc une généralisation des techniques de couplage de tableaux (Escofier et Pagès, 1984), (Escofier et Pagès, 1986) et (Escofier et Pagès, 1994). C'est une des méthodes les plus pratiques parmi les méthodes d'ordination simultanée de K tableaux ayant en commun n lignes-individus (Tableau 7.3), chacun d'eux correspondant à un groupe de variables-colonnes (Chessel, 1997).

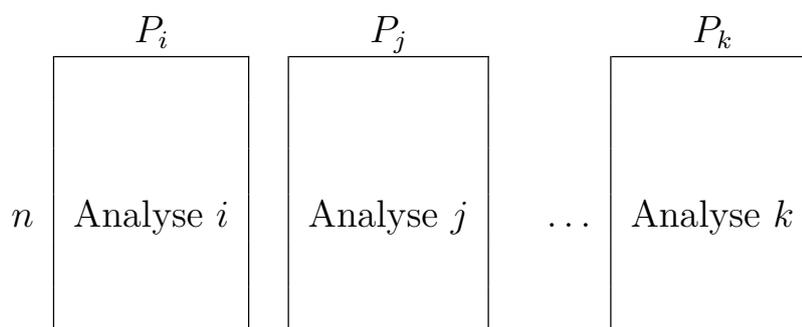


Tableau 7.3 – Ensemble de k -tableaux ayant en commun n individus

P_i	Catégorie de questions	Abréviation
P_1	Les individus enquêtés	INDIV
P_2	Profil des individus Enseignant	ENS
P_3	Profil des individus Elève-étudiant	EL_ETU
P_4	Connaissances et utilisation de (TIC) dans l'enseignement	CON_ROLE_TIC
P_5	Formation en informatique	FORM_INFO
P_6	Connaissances et utilisation du (TICE) dans l'enseignement/apprentissage	CON_ROLE_TICE
P_7	Place et le rôle de l'ordinateur et/ou de l'internet (TIC) dans l'apprentissage de l'informatique.	ROLE_TIC_APPR_INFO

Tableau 7.4 – Répartition des questions par rapport aux groupes de variables

Mise en œuvre de la CAH

La CAH des individus est ensuite réalisée sur l'ensemble des axes factoriels et une typologie est proposée par coupure optimale de l'arbre construit. Cette typologie est appliquée sur les composantes principales des AFM avec agrégation selon le critère de WARD qui minimise l'inertie intra-classes et maximise l'inertie inter-classes.

7.4.2 Résultats et discussions

7.4.2.1 Résultats et discussions de l'utilisation de l'AFM

Caractéristiques des individus observés

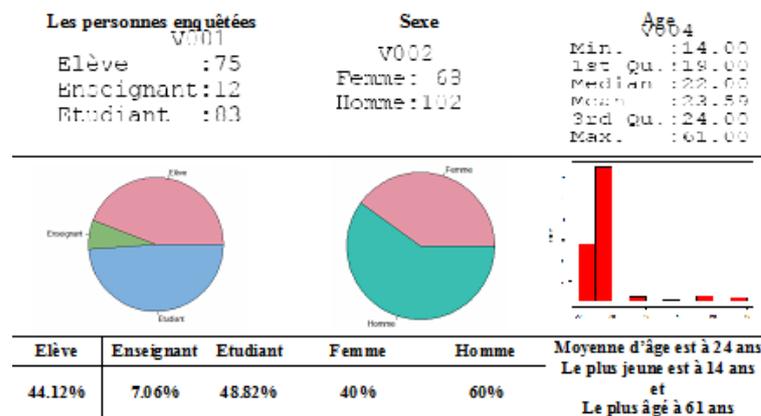


Figure 7.2 – Caractéristiques et graphes respectifs des personnes enquêtées

Valeurs propres des paramètres de données

Le tableau 7.5 donne les valeurs de deux séries de paramètres dont nous n'avons pas

encore parlé : les valeurs propres, désignées par λ (comprises entre 0 et 1.783).

	Valeurs propres	Pourcentages de variance	Pourcentages de variance cumulés
comp 1	1.783	4.366	4.366
comp 2	1.483	3.631	7.997
comp 3	1.322	3.237	11.235
.....
comp 169	0.0000657	0.000161	100

Tableau 7.5 – Extrait des valeurs propres et pourcentages de variance (ou d'inertie)

Sur la figure 7.3, les valeurs propres valent ici $\lambda_1 = 1.783$ pour le premier axe et $\lambda_2 = 1.483$ pour le second ; les pourcentages de variance, appelés encore pourcentages d'inertie (rapport de chacune des valeurs propres à leur somme globale (Cf. C.2, T-C.2) correspondant à ces valeurs propres valent à leurs tours 4.366% et 3.631% pour ces deux mêmes axes. En pratique, la stabilité du plan factoriel se démontre par des techniques de simulation (perturbations aléatoires du tableau de données) ou de validation par échantillon-test (positionnement d'individus n'ayant pas participé à la construction des axes). Le choix du nombre d'axes à retenir est prépondérant. Après plusieurs simulations selon plusieurs paquets d'axes retenus, à partir de 3 axes, les projections ne suivent plus les tendances. Pour la suite de l'étude, nous avons adopté deux axes selon les valeurs propres susmentionnées.

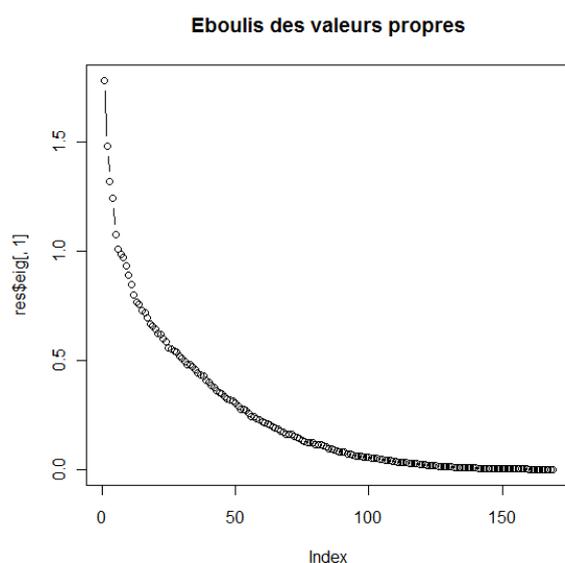


Figure 7.3 – Eboulis des valeurs propres

Pourcentages de variance (ou d'inertie)

Les pourcentages de variance mesurent l'importance relative de chaque valeur propre dans le tracé. Le tableau 7.5 montre les différentes valeurs des 3 premiers plans factoriels. Celle du 1er plan factoriel représente 4.366% de la variance totale. En général, ces pourcentages sont au contraire une mesure pessimiste de la part d'information représentée. De nombreux contre-exemples dans la littérature montrent que des pourcentages médiocres correspondent parfois à des représentations qui rendent compte de façon satisfaisante de la structure des données. Cette affirmation concorde bien avec notre cas.

Coordonnées de groupes

Les coordonnées des tableaux sont affichées et utilisées pour créer les graphiques afin de mieux représenter et interpréter le comportement des groupes. Ces graphiques permettent notamment de visualiser la distance entre les tableaux.

	Dim.1	Dim.2
ENS	0.008	0.6377
EL_ETU	0.8657	0.3421
CON_UTIL_TIC	0.2157	0.2024
CON_UTIL_TICE	0.3653	0.1387
ROLE_TIC_APPR_INFO	0.3282	0.1617

Tableau 7.6 – Coordonnées de groupes sur les axes

7.4.2.2 Autres aides à l'interprétation

Deux séries de paramètres permettent d'interpréter avec plus de sûreté les résultats, en complétant l'information donnée par les coordonnées des éléments sur les axes factoriels : les *contributions* et les *cosinus carrés*.

- Les *contributions* (ou *contributions absolues*), qui décrivent la part prise par un élément (ligne ou colonne) dans la construction d'un axe factoriel. Les contributions (ou contributions absolues), qui décrivent la part prise par un élément (ligne ou colonne) dans la construction d'un axe factoriel.

	Dim.1	Dim.2
ENS	0.4506	43.0116
EL_ETU	48.5516	23.0763
CON_UTIL_TIC	12.0996	13.6509
CON_UTIL_TICE	20.4899	9.3524
ROLE_TIC_APPR_INFO	18.4083	10.9087

Tableau 7.7 – Contributions des groupes sur les 2 axes factoriels

Les contributions sont une aide à l'interprétation. Les tableaux influençant le plus la construction des axes sont deux dont les contributions sont les plus élevées. Le tableau 7 montre que les groupes de variables EL_ETU et ENS sont respectivement ceux qui ont les plus de contribution sur les axes 1, 2.

- Les *cosinus carrés* (ou *contributions relatives*) qui mesurent la qualité de la représentation de chaque élément par les axes. Comme pour les autres méthodes factorielles, l'analyse des cosinus carrés permet d'éviter des erreurs d'interprétation dues à des effets de projection. Si les cosinus carrés associés aux axes utilisés sur un graphique sont faibles, on évitera d'interpréter la position du tableau en question. Le comportement des cosinus carrés (Cf. Tableau 7.8) suit celui des contributions.

	Dim.1	Dim.2
ENS	0	0.1143
EL_ETU	0.1294	0.0202
CON_UTIL_TIC	0.0143	0.0126
CON_UTIL_TICE	0.0348	0.005
ROLE_TIC_APPR_INFO	0.0216	0.0052

Tableau 7.8 – Cosinus carrés des groupes sur les 2 axes factoriels

Coefficient de similarité \mathcal{L}_g

Les coefficients \mathcal{L}_g de liaison entre les tableaux permettent de mesurer à quel point les tableaux sont liés deux à deux. La liaison sera d'autant plus forte que l'ensemble des variables d'un tableau seront liées à celle du second. Pour un groupe donnée ($\mathcal{L}_g(X1; X2)$) vaut la somme des carrée des valeurs propres du groupe (indicateur de multidimensionalité) ; mais entre deux groupes il est égal à la somme pondérée des carrées des covariances entre chaque colonne des deux groupes (équation 7.1).

$$\mathcal{L}_g(K_j, K_m) = \sum_{k \in K_j} \sum_{l \in K_m} cov^2 \left(\frac{x_k}{\lambda_1^k}, \frac{x_l}{\lambda_1^l} \right) \quad (7.1)$$

\mathcal{L}_g est d'autant plus élevé que les deux groupes comparés ont des directions d'inertie importantes communes. Dans notre cas, le groupe d'élèves et d'étudiants EL_ETU nous donne une bonne description (5.0012 est la plus grande valeur de \mathcal{L}_g) (Tableau 7.9).

Coefficient de corrélation vectoriel RV

Les coefficients RV de liaison entre les tableaux sont une autre mesure de la liaison entre les tableaux. Les coefficients RV (Tableau 7.10) dont la valeur est comprise

	AGE	ENS	EL_ETU	CON_UTIL_TIC	CON_UTIL_TICE	ROLE_TIC_APPR_INFO	INDIV_INDIV	FORM_INFO_INFO	AFM
AGE	1	0,7444	0,251	0,0639	0,0691	0,0292	0,5337	0,0496	0,6493
ENS	0,7444	3,4261	0,2754	0,2607	0,1485	0,1379	0,5817	0,0709	2,3828
EL_ETU	0,251	0,2754	5,0012	0,7761	1,2712	0,7778	0,9533	0,4189	4,5439
CON_UTIL_TIC	0,0639	0,2607	0,7761	1,9399	0,6107	0,6055	0,2052	0,303	2,3516
CON_UTIL_TICE	0,0691	0,1485	1,2712	0,6107	2,3456	0,6192	0,2803	0,4843	2,8016
ROLE_TIC_APPR_INFO	0,0292	0,1379	0,7778	0,6055	0,6192	3,8228	0,1886	0,302	3,3445
INDIV	0,5337	0,5817	0,9533	0,2052	0,2803	0,1886	1,4256	0,1206	1,239
FORM_INFO	0,0496	0,0709	0,4189	0,303	0,4843	0,302	0,1206	2,0216	0,8856
AFM	0,6493	2,3828	4,5439	2,3516	2,8016	3,3445	1,239	0,8856	8,6508

Tableau 7.9 – Indicateur de similarité \mathcal{L}_g

entre 0 et 1, correspondent à une normalisation des coefficients \mathcal{L}_g :

$$0 \leq \mathcal{L}_g(v_1, K_j) = \frac{1}{\lambda_1^j} \underbrace{\sum_{k \in K_j} cov^2(x_k, v_1)}_{\leq \lambda_1^j} \leq 1 \quad (7.2)$$

- mesure la liaison entre deux tableaux de données portant sur les mêmes individus
- RV=1 si les nuages associés sont homothétiques
- RV positif ou nul c'est la standardisation de L_g

	AGE	ENS	EL_ETU	CON_UTIL_TIC	CON_UTIL_TICE	ROLE_TIC_APPR_INFO	INDIV_INDIV	FORM_INFO_INFO	AFM
AGE	1	0.4022	0.1123	0.0459	0.0451	0.0149	0.447	0.0349	0.2207
ENS	0.4022	1	0.0665	0.1011	0.0524	0.0381	0.2632	0.0269	0.4377
EL_ETU	0.1123	0.0665	1	0.2492	0.3712	0.1779	0.357	0.1317	0.6908
CON_UTIL_TIC	0.0459	0.1011	0.2492	1	0.2863	0.2224	0.1234	0.153	0.574
CON_UTIL_TICE	0.0451	0.0524	0.3712	0.2863	1	0.2068	0.1533	0.2224	0.6219
ROLE_TIC_APPR_INFO	0.0149	0.0381	0.1779	0.2224	0.2068	1	0.0808	0.1086	0.5816
INDIV	0.447	0.2632	0.357	0.1234	0.1533	0.0808	1	0.0711	0.3528
FORM_INFO	0.0349	0.0269	0.1317	0.153	0.2224	0.1086	0.0711	1	0.2118
AFM	0.2207	0.4377	0.6908	0.574	0.6219	0.5816	0.3528	0.2118	1

Tableau 7.10 – Indicateur RV

Le groupe ayant la valeur la plus proche de celle de l'AFM est l'EL_ETU. Ce dernier constitue par conséquent l'inertie de l'ensemble des variables du groupe dont la coordonnée vaut 0.8657 (Tableau 7.6). La figure 7.4 représente les groupes de variables projetés sur les deux premiers axes. On peut y observer l'inertie d'EL_ETU sur l'axe 1 et ENS sur l'axe 2.

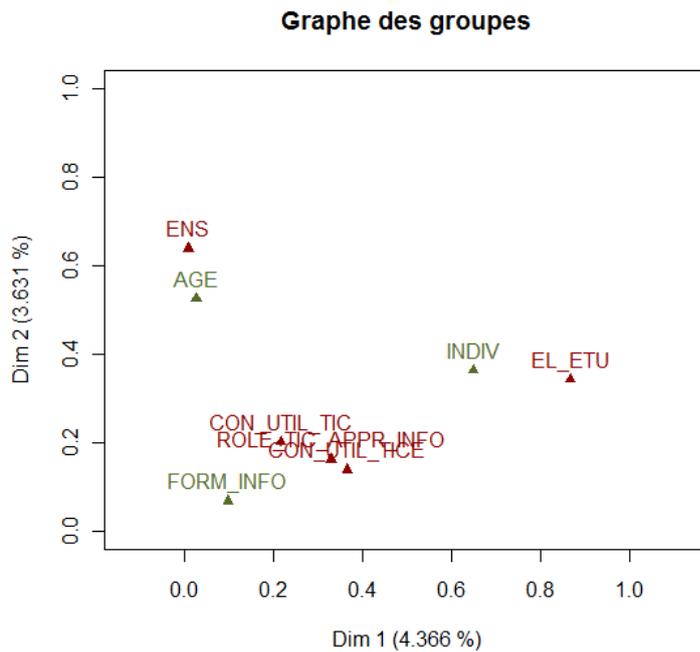


Figure 7.4 – Représentation de groupes de variables

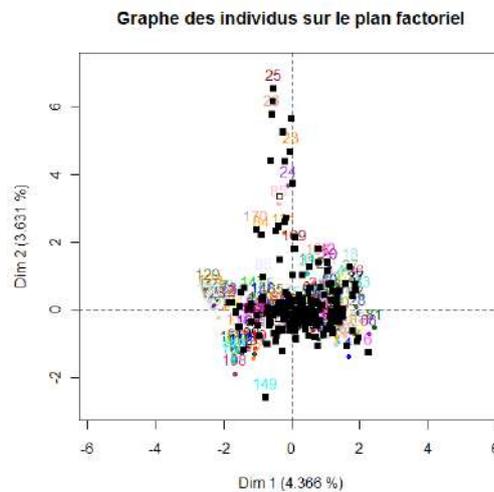
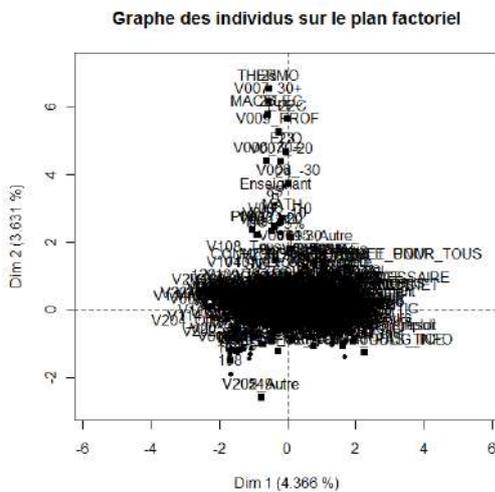


Figure 7.5 – Représentation des individus et variables sur le plan factoriel

Figure 7.6 – Représentation des individus sur le plan factoriel

Par ailleurs, on peut voir que 3 groupes (CON_UTIL_TIC, ROLE_TIC_APPR_INFO, CON_UTIL_TICE) se concentrent sur une inertie assez faible selon les 2 axes. Ces groupes de variables sont relatifs à la connaissance du TIC. Nous pouvons donc tirer une première conclusion que les individus enquêtés ont une connaissance insuffisante du TIC. Les figures 7.5 et 7.6 représentent respectivement les projections

sur les axes 1 et 2 des individus et variables, et des individus uniquement. L'axe 1 est commun à tous les groupes, par contre l'axe 2 principalement pour le groupe de variables ENS (figures 7.4 à 7.6). Le premier axe est corrélé à toutes les variables : il oppose les individus ayant une évaluation positive pour les critères satisfaisant les 3 aspects (CON_UTIL_TIC, ROLE_TIC_APPR_INFO, CON_UTIL_TICE) de leur connaissance personnelle, aux individus ayant émis les opinions opposées. Le deuxième axe met en évidence la spécificité de la situation de l'ENS vis à vis de leur formation FORM_INFO.

Les figures 7.7 et 7.8 représentent respectivement les groupes de variables et les variables quantitatives sur le plan factoriel. On peut observer sur la figure 7.6 la variable âge (V004) qui a une forte corrélation avec le groupe de variables AGE de la figure 7.5. Ces deux représentations montrent que les élèves et étudiants, vue leurs âges, connaissent peu le rôle et la place du TIC dans l'enseignement et plus particulièrement l'apprentissage de l'informatique malgré leur faible formation par rapport aux enseignants qui n'en reçoivent presque aucune. Les élèves, les étudiants, les enseignants connaissent mal le TIC et le TICE ainsi leur utilisation, mais les élèves et les étudiants utilisent le TIC et le TICE plus que les enseignants.

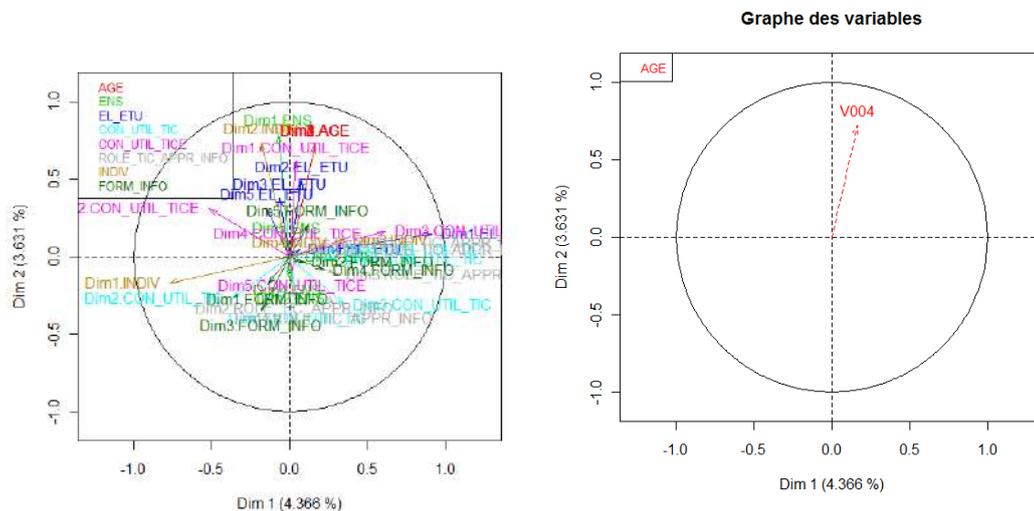


Figure 7.7 – Représentation de groupes de variables sur le plan factoriel

Figure 7.8 – Représentation des variables quantitatives sur le plan factoriel

7.4.2.3 Résultats et discussions de l'utilisation de la CAH

La Classification Automatique et analyse factorielle s'inscrivent dans une même perspective (l'analyse exploratoire d'un tableau rectangulaire) et diffèrent selon le mode de représentation (nuage euclidien, hiérarchie indicée ou partition). D'où l'idée de combiner les approches pour obtenir une méthodologie riche, qualité essentielle en statistique exploratoire car le fait de disposer de plusieurs points de vue ne peut que renforcer la solidité des conclusions. Dans ce cas, on utilise pour chaque méthode la même distance (euclidienne) entre individus. L'ensemble des axes factoriels est pris en compte pour conserver toute la variabilité de l'information initiale des données traitées, quitte à majorer le nombre de classes pour traiter des observations particulières et éviter d'occulter leur existence.

La classification d'individus décrits par des variables qualitatives est un vaste domaine d'étude et de nombreuses mesures de ressemblance existent. Cependant, ces méthodes sont difficiles à mettre en œuvre et en pratique il est courant d'utiliser l'Analyse Factorielle comme prétraitement : les variables sont transformées en composantes principales, c'est-à-dire en variables quantitatives. La classification est construite à partir des (premières) dimensions factorielles. Un arbre hiérarchique peut être considéré comme une séquence de partitions emboîtées, de la plus précise (un individu par classe) à la plus grossière (une seule classe). Ainsi, une hiérarchie est extrêmement utile pour déterminer le nombre de classes. Ce choix peut être fait à partir de l'allure générale de l'arbre, du niveau des nœuds (ces niveaux peuvent être représentées par un diagramme en barres), du nombre de classes (qui doit être ni trop faible ni trop grand) et de l'interprétabilité des classes.

Interprétation par les individus

Le tableau 7.11 suivant montre la répartition des individus selon leur classe respective :

	Classe 1	Classe 2	Classe 3
Individus	86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 135 136 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 154 156 157 158 163 164 165 167	22 23 24 25 26 84 85 131 169 170	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 132 133 134 137 153 155 159 160 161 162 166 168
TOTAL	70	10	90

Tableau 7.11 – Répartition des individus dans la classe

Dans la classe 1, les individus 100 et 129 sont les deux points les plus éloignés. 121, 145, 146 sont les plus proches du centre de gravité 150 de la classe dénommés « parangons », 115, 120, 130, 126 sont les plus éloignés du centre de gravité dénommés « extrêmes » (Figure 7.9). Dans la classe 2, les individus 170 et 25 sont les deux points les plus éloignés. 85, 23, 131 sont les plus proches du centre de gravité 24 de la classe, 23, 22, 26 sont les plus éloignés du centre de gravité. Dans la classe 3, les individus 1 et 81 sont les deux points les plus éloignés. 10, 38, 13 sont les plus proches du centre de gravité 82 de la classe, 36, 83, 68, 6 sont les plus éloignés du centre de gravité.

\$paramètre					
classe: 1					
150	121	145	146	100	
0.2190629	0.2757329	0.2798702	0.3123107	0.3483394	

classe: 2					
24	85	23	131	170	
0.2902734	0.6807461	0.9266740	1.5429907	1.5828670	

classe: 3					
82	10	38	13	1	
0.07919937	0.17656844	0.18160988	0.18280272	0.18420994	

\$distance					
classe: 1					
129	126	130	120	115	
3.643882	3.561288	3.544226	3.542378	3.525908	

classe: 2					
25	26	22	23	24	
6.868289	6.135678	5.879437	4.939545	3.995319	

classe: 3					
81	6	68	83	36	
3.786870	3.735145	3.654045	3.536925	3.43634	

Figure 7.9 – Classification des individus

Les figures 7.10 et 7.11 suivantes représentent respectivement la classification des individus sur le plan factoriel en 3D (dendrogramme) et les individus sur le plan factoriel selon CAH :

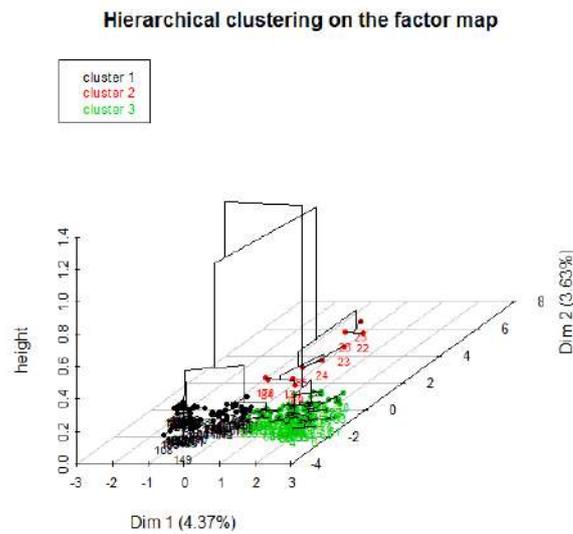


Figure 7.10 – Représentation de la classification des individus sur le plan factoriel en 3D

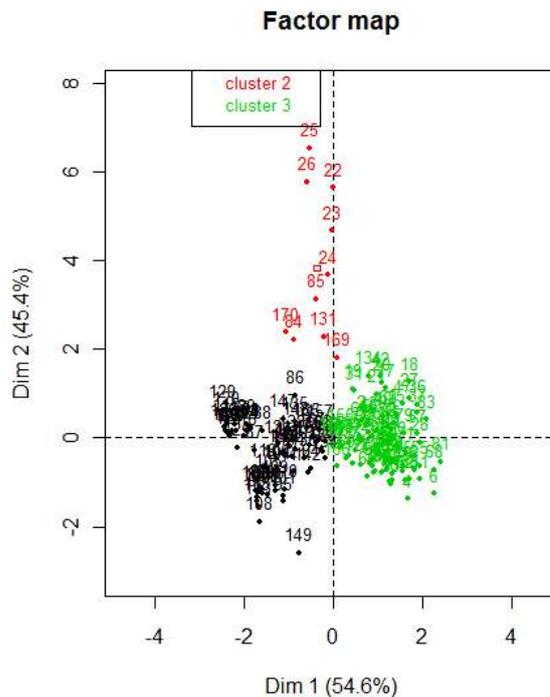


Figure 7.11 – Représentation de la classification des individus sur le plan factoriel en 2D

Les figures 7.10 et 7.11 mettent en évidence les trois classes d'individus. Sur la figure 7.11, les états des individus entre eux et, par rapport au centre de gravité 24 sont

bien perceptibles pour le cas de la classe 2. Les deux autres classes présentent des formes semblables mais opposées. Contrairement à la distribution de la classe 2 qui est de forme allongée, les états des individus des deux autres classes sont difficilement identifiables.

Interprétation par les variables En ce qui concerne les variables, des analyses ont été effectuées en se servant de la table de rapports Classe/Modalité et Valeur-test C.3. Trois tables [T-C.3, T-C.4, T-C.5] respectives à chaque classe ont donc été élaborées et utilisées pour les interprétations ci-dessous.

Classe 1+ : Les élèves en Terminale A et 1ère D de l'enseignement Général de sexe féminin connaissent l'existence de parc informatique dans leur Lycée, mais ils n'ont pas accès et travaillent rarement sur ordinateur. Ils se connectent à l'internet hors du lycée pour les envois de mails mais rarement. Ils sont conscients que l'ordinateur et l'internet peuvent servir à apprendre efficacement. La plupart disent qu'ils n'ont pas le temps d'accéder à la connexion mais ils proposent que le service devrait être gratuit pour améliorer l'accès à l'internet. Ils n'ont pas eu de formations en informatique malgré leur intérêt surtout à la formation en internet. Ils n'utilisent pas d'outils TICE comme le vidéoprojecteur mais ils sont presque unanimes sur l'efficacité d'utilisation de ces outils pour l'apprentissage. Les supports qu'ils utilisent pendant leurs études personnelles sont seulement des livres. Cependant, ils ont des supports électroniques de leurs cours à 50%. Ils connaissent l'existence du projet EDUCMAD, mais ne sont pas satisfaits de l'impact de ce projet dans leur vie collégiale. Ils parlent de l'efficacité de l'utilisation du TIC dans l'apprentissage de l'informatique car ceci leur apporterait plus de motivation, assiste les élèves faibles et leur asservit. Le TIC offre des ressources fiables et favorise l'apprentissage.

Classe 1- : Les étudiants en 1, 2, 3 et 4ème année GM, GMI, et GE du secteur technique et industriel et un enseignant (individu 86) de sexe masculin travaillent souvent avec l'ordinateur. Ils l'utilisent pour leur formation et leur passe temps. Ils rencontrent de problèmes dans l'utilisation de l'internet à cause des débits très lents et proposent pour qu'il y ait accès à tout moment. Ces individus utilisent des documents manuscrits pendant leurs études personnelles et la préparation de cours.

Seulement 25% de leur cours sont numériques car ils ont des problèmes de disponibilité. Ils ne connaissent pas l'existence du projet EDUCMAD et c'est ainsi qu'ils n'ont pas d'idées sur ce projet. D'après eux, le TIC aide un peu sur l'apprentissage de l'informatique à cause du coût de matériels élevé malgré l'apport de ceci pour l'amélioration du contenu des cours.

Classe 2+ : Les enseignants surtout les Professeurs de Mathématiques, qui ont effectué en moyenne 20 ans de services, utilisent l'ordinateur pour la saisie de leurs cours (75%). Ils ont accès à l'internet au Lycée/Université, mais ils n'ont pas de formation en informatique et n'utilisent pas d'outils TICE.

Classe 2- : Les étudiants surtout les élèves série A de l'enseignement général disent que la formation en informatique aide à l'apprentissage et ils s'intéressent à des formations. Le tableau noir sert toujours à apprendre efficacement. Ils n'ont pas de cours version numérique faute de moyen et n'ont pas entendu parler du TIC, TICE et le projet EDUCMAD.

Classe 3+ : Les étudiants de sexe masculin en 1, 2, 3, 4ème année GE, GMI et GM secteur technique et industriel utilisent souvent l'ordinateur pour se distraire. Ils utilisent rarement l'internet pour leur formation et pour leur passe temps. Ils rencontrent des problèmes de débits très lents. Ils proposent des accès à tout moment et des informations sur la disponibilité des matériels pour l'amélioration à venir. Ils ne s'intéressent pas à la formation de l'informatique. Ils utilisent le vidéoprojecteur pour leur apprentissage mais ne savent pas l'efficacité de ceci, de même pour les autres outils TICE dans l'enseignement. Ils utilisent des documents manuscrits pendant leurs études personnelles ; seulement 25% de leur cours sont numériques car ils ne sont pas disponibles et n'ont pas de moyens. Ils ont entendu parler du TIC mais par contre, n'ont aucune idée sur le TICE et le projet EDUCMAD. Ils disent que le TIC peut servir à apprendre un peu l'informatique car il offre des cours, des exercices mais ceci exige des moyens coûteux. Le rôle du TIC dans l'enseignement pour eux est pour améliorer le cours.

Classe 3- : Les enseignants, élèves en 1ère et Terminale A et D de l'enseignement

général savent que le Lycée/Université disposent d'ordinateurs et d'une connexion internet. Ils travaillent rarement sur ordinateur. Ils se connectent à l'internet chez eux seulement pour envoyer des mails car ils n'ont pas beaucoup de temps. Pour qu'il y ait accès pour tous, ils proposent que le service soit gratuit. Ils s'intéressent par une formation en informatique surtout dans le domaine de l'internet. Ils disent que les outils TICE servent efficacement à apprendre. Ils utilisent des livres pendant leurs études personnelles et/ou les préparations de leurs cours. Ils savent l'existence du projet EDUCMAD, mais ils ne sont pas satisfaits du tout de ce projet. Ils disent que le TIC sert à apprendre efficacement l'informatique car celui peut assister les élèves faibles, apporte des motivations et on y trouve des ressources intéressantes qui asservissent l'homme et favorise l'apprentissage et la recherche.

Dans cette étude, les résultats des deux approches complémentaires AFM et CAH nous montrent la méconnaissance des acteurs sur l'utilisation et le rôle du TIC dans l'enseignement malgré le slogan médiatisé sur « celui qui ne manipule pas l'ordinateur à notre ère est considéré comme un analphabète », le non accès des élèves, des étudiants et surtout des enseignants au TIC malgré les différents nombreux projets qui se sont mis en œuvre dans le pays depuis quelques années et la volonté des dirigeants de la mise en place de l'accessibilité à cette révolution technico-socio-éducatif mondiale. Le manque de formation sur cette haute technologie TIC malgré de nombreux projets d'appui institutionnel financés par les bailleurs de fonds étrangers.

7.4.3 Stratégie à mettre en œuvre

Les résultats des enquêtes sont alarmants en termes de méconnaissance aussi bien au niveau des élèves/étudiants que des enseignants. La première principale action qu'il faut mener est sans doute la revalorisation et relance de la politique de vulgarisation du TIC à tous les niveaux. Les individus les plus âgés ont en général du mal à s'adapter au changement, il s'avère ainsi de commencer cette action dès le plus jeune âge. De même, il faut reconduire la politique de détaxation des matériels TIC à usage collectif et créer des centres d'accès gratuits dans toutes les circonscriptions territoriales de la Région et dans tous les établissements publics et privés. A partir du moment où les individus sont sensibilisés en terme de TIC et y ont accès facilement, des campagnes de formations régulières sont primordiales et inévitables. Nous tenons

à noter aussi que des projets avaient déjà été entrepris par l'état mais n'ont pas été suffisamment efficaces. Ainsi, une pérennisation des projets s'impose. Enfin, le TIC va de pair avec la situation économique et énergétique du pays. Afin d'arriver à bon terme et de pérenniser les actions mises en place et futures, ces deux filières doivent être traitées en amont et prioritairement, et ceci de façon urgente.

7.4.4 Conclusion partielle

Des enquêtes sur les TIC ont été menées dans quelques établissements de la Région DIANA, en particulier le centre ville d'Antsiranana afin de proposer une stratégie de l'intégration du TIC dans l'enseignement. Un modèle de formulaire a été élaboré. Ceci a permis de recueillir 170 retours d'enquêtes. Les données ainsi recueillies ont été analysées en utilisant l'Analyse Factorielle Multiple et la Classification Ascendante Hiérarchique. Plusieurs indicateurs sont sortis de l'analyse des variables et individus considérés, en particulier la méconnaissance du TIC, le non accès au TIC et le manque de formation en TIC. La stratégie à mettre en place afin d'y remédier devrait passer par la revalorisation et relance des politiques qui avaient déjà été menées mais aussi le traitement prioritaire de nos filières économiques et énergétiques.

7.5 Analyse et diagnostic sur la connaissance en informatique des étudiants de l'enseignement supérieur à Madagascar

Une autre étude a été faite (Cf. article ([Rakotomalala, Ralahady, et Totohasina, 2018](#))) sur la connaissance en informatique des étudiants de l'enseignement supérieur. Ainsi, nous avons mené des enquêtes corporelles auprès de l'Université Privés et Publiques d'Antananarivo choisis comme sites de référence, en vue d'élaborer une étude diagnostique sur la connaissance et l'utilisation de l'informatique. Pour cela, nous avons élaboré un modèle de questionnaire qui est constitué de six grandes parties :

- Les caractéristiques des individus enquêtés (étudiants) ;
- Les connaissances des enquêtés et leur utilisation de l'ordinateur et de l'internet comme composantes TIC ;

- Les connaissances des enquêtés et leur utilisation du TICE ;
- L'opinion des enquêtés sur le rôle du TIC dans l'enseignement et apprentissage de l'informatique ;
- Leurs connaissances en programmation ;
- Les méthodologies utilisées par les enquêtés et les problèmes rencontrés pour l'enseignement/apprentissage en algorithmique.

En tout, le questionnaire composé de cinquante cinq questions formant cent cinquante deux variables ont été réparties dans les six parties susmentionnées.

7.5.1 Matériels et méthodes utilisés pour la fouille des données

Les questionnaires ont permis d'effectuer des enquêtes portées en février 2017 sur les étudiants de l'Université publics et privés d'Antananarivo concernant leur situation universitaire, et d'une part leur connaissance sur le TIC, TICE, la programmation et l'algorithme. 150 fiches ont été distribuées et nous avons eu en retour 80 questionnaires, soit 53%. Les versions papiers ont été numérisées en se servant de l'outil Excel pour constituer une base de données au format .csv dont l'extrait est présenté dans le tableau 7.12, pouvant nous permettre d'effectuer les différentes analyses qui sont exposées par la suite.

Par contre, nous adoptons cette fois-ci la méthode de classification hiérarchique implicite et cohésitive selon la mesure de qualité M_{GK} , dont les détails de notre théorie ont été exposés dans le chapitre 4 et chapitre 5 de cette thèse.

	V1	V2	V3	V150	V151	V152
i1	1	0	0	1	1	0
i2	1	0	0	0	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
i79	0	1	0	1	0	0
i80	1	0	0	1	0	0

Tableau 7.12 – Extrait du tableau binaire de l'enquête sur la connaissance en informatique des étudiants

En appliquant l'algorithme d'extraction des RA selon M_{GK} (Cf. 4.4), nous avons eu deux mille deux cent trente huit (2 238) règles valides et avons calculé les valeurs

de la normalisée du support basé sur M_{GK} de chacun de ces règles valides avec $0.034 \leq \text{supp}_{(n)M_{GK}}^f \leq 1$. Les valeurs de $\text{supp}_{(n)M_{GK}}^f$ sont des indicateurs de la force de la relation implicative entre chaque couple de variables formant des règles valides selon M_{GK} , ensuite des cohésions $\text{coh}_{\text{supp}_{(n)M_{GK}}}$ formées de six cent neuf couples de variables orientées ont été déduites en utilisant la définition 4.14 §4.4.5. celles-ci signent la consistance de chaque classe formée par chaque couple de variables des règles valides et nous constatons que $0.07 \leq \text{coh}_{\text{supp}_{(n)M_{GK}}} \leq 1$. Chaque couple de variables forme une paire orientée PO avec $\text{Card}(PO) = 609$:

$$PO = \{((v_2, v_{150})), ((v_2, v_{150})), ((v_3, v_7)), \dots, ((v_{151}, v_{143})), ((v_{152}, v_{148}))\}$$

Ces paires orientées ordonnées forment le préordre initial et global cohésitif Ω . Le préordre $\Omega : ((v_{19}, v_1)) \leq ((v_{130}, v_1)) \leq \dots \leq ((v_{143}, v_{101})) \leq ((v_{14}, v_{103}))$.

D'après les 3 figures 7.12, les nœuds significatifs sont les extremums de la courbe, i.e. les nœuds au niveau $H = \{1, 2, 3, 4, 7, 8, 9, 11, 12, 13, 15, 16, 17, 19, 20, 21, 23, 24, 26, 27, 28, 29, 30, 32, 33, 34, 36, 37, 38, 39, 41, 44, 45, 46, 47, 49, 50, 51, 54, 55, 56, 57, 58, 59, 61, 62, 63, 64, 66, 67, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78\}$ représentés en croix dans la figure 7.13. Soit 71.79% des nœuds sont significatifs. Mais parmi eux, les $H = \{1, 7, 8, 9, 11, 12, 13, 15, 16, 17, 19, 20, 21, 23, 24, 28, 29, 30, 36, 37, 38, 44, 45, 46, 54, 55, 56, 58, 59, 63, 64, 66, 67, 70, 71, 72, 76, 77, 78\}$ ont la plus de significativité. Car, pour les autres nœuds, la variation de la courbe qu'ils représentent est presque monotone.

La hiérarchie implicative entre classes montre une structuration plus large à un niveau supérieur épistémologique. La méthode adoptée nous permet de valider certaines hypothèses formulées a priori, mais aussi de les contrarier et mettre en évidence des formules imprévues ou des phénomènes complexes. Nous obtenons alors soixante dix huit (78) méta-règles (Cf. Tableau 7.13) et allons interpréter quelques unes.

Tableau 7.13 – Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau pour la connaissance en informatique des étudiants.

Niveau	Règle	Cohesion $coh_{supp(c),M,GK}(\underline{C})$	inter-classes (\underline{C})	Implication $\psi(\underline{C}, \underline{C}')$	inter-classes $s(\Omega, H)$	Niveau significatif	Noeud $v(\Omega, H)$	significatif
H(1) :	(V19 =>V1)	1,00000000		1,00000000	0,0000755		0,0000755	
H(2) :	(V37 =>V2)	1,00000000		1,00000000	0,00000000		-0,0000755	
H(3) :	(V36 =>V3)	1,00000000		1,00000000	-0,0000251		-0,0000252	
H(4) :	(V119 =>(V36 =>V3))	1,00000000		1,00000000	-0,0000239		0,0000012	
H(5) :	(V5 =>V4)	1,00000000		1,00000000	0,0001401		0,0001640	
H(6) :	(V116 =>V6)	1,00000000		1,00000000	0,0003266		0,0001865	
H(7) :	(V48 =>V7)	1,00000000		1,00000000	0,0029111		0,0025845	
H(8) :	(V44 =>V8)	1,00000000		1,00000000	-0,0006257		-0,0035368	
H(9) :	(V69 =>V9)	1,00000000		1,00000000	-0,0003274		0,0002983	
H(10) :	(V10 =>V46)	1,00000000		1,00000000	-0,0002388		0,0000886	
H(11) :	(V14 =>V11)	1,00000000		1,00000000	-0,0001963		0,0000425	
H(12) :	(V41 =>V12)	1,00000000		1,00000000	0,0002504		0,0004467	
H(13) :	(V13 =>V15)	1,00000000		1,00000000	0,0003911		0,0001407	
H(14) :	(V124 =>V16)	1,00000000		1,00000000	0,0007454		0,0003543	
H(15) :	(V133 =>V17)	1,00000000		1,00000000	0,0033353		0,0025900	
H(16) :	(V117 =>V21)	1,00000000		1,00000000	-0,0016581		-0,0049935	
H(17) :	(V28 =>V22)	1,00000000		1,00000000	-0,0007178		0,0009403	
H(18) :	(V23 =>V32)	1,00000000		1,00000000	-0,0004784		0,0002394	
H(19) :	(V24 =>V27)	1,00000000		1,00000000	-0,0003690		0,0001094	
H(20) :	(V26 =>V150)	1,00000000		1,00000000	0,0004119		0,0007809	
H(21) :	(V30 =>V49)	1,00000000		1,00000000	0,0006081		0,0001962	
H(22) :	(V59 =>V31)	1,00000000		1,00000000	0,0010688		0,0004607	
H(23) :	(V34 =>V43)	1,00000000		1,00000000	0,0034325		0,0023637	
H(24) :	(V35 =>V111)	1,00000000		1,00000000	-0,0033677		-0,00068002	
H(25) :	(V47 =>V39)	1,00000000		1,00000000	-0,0011959		0,0021718	
H(26) :	(V62 =>(V47 =>V39))	1,00000000		1,00000000	-0,0004725		0,0007234	
H(27) :	(V127 =>(V62 =>(V47 =>V39)))	1,00000000		1,00000000	0,0008658		0,0013383	
H(28) :	(V40 =>V67)	1,00000000		1,00000000	0,0034495		0,0025838	
H(29) :	(V42 =>V128)	1,00000000		1,00000000	-0,0068321		-0,0102816	
H(30) :	(V82 =>V45)	1,00000000		1,00000000	-0,0018006		0,0050315	
H(31) :	(V132 =>V50)	1,00000000		1,00000000	-0,0010643		0,0007363	

Suite page suivante

Niveau	Règle	Cohesion $coh_{suspp(n)} M_G K$	inter-classe \underline{C}	Implication $\psi(\underline{C}, \underline{C}')$	inter-classe	Niveau $s(\Omega, H)$	significatif	Noeud $v(\Omega, H)$	significatif
H(32) :	(V63 => V54)	1,0000000		1,0000000		-0,0007685		0,0002958	
H(33) :	(V135 => V55)	1,0000000		1,0000000		0,0006897		0,0014582	
H(34) :	(V141 => V56)	1,0000000		1,0000000		0,0009595		0,0002698	
H(35) :	(V64 => V125)	1,0000000		1,0000000		0,0015233		0,0005638	
H(36) :	(V65 => V83)	1,0000000		1,0000000		0,0034358		0,0019125	
H(37) :	(V68 => V66)	1,0000000		1,0000000		-0,0179760		-0,0214118	
H(38) :	(V70 => V74)	1,0000000		1,0000000		-0,0025976		0,0153784	
H(39) :	(V72 => V103)	1,0000000		1,0000000		-0,0014321		0,0011654	
H(40) :	(V113 => V75)	1,0000000		1,0000000		-0,0010084		0,0004287	
H(41) :	(V137 => V77)	1,0000000		1,0000000		0,0008087		0,0018121	
H(42) :	(V121 => V79)	1,0000000		1,0000000		0,0011014		0,0002927	
H(43) :	(V88 => V84)	1,0000000		1,0000000		0,0016839		0,0005825	
H(44) :	(V87 => V91)	1,0000000		1,0000000		0,0034077		0,0017238	
H(45) :	(V134 => V93)	1,0000000		1,0000000		0,1715326		0,1681249	
H(46) :	(V94 => V144)	1,0000000		1,0000000		-0,0037064		-0,1752390	
H(47) :	(V129 => V108)	1,0000000		1,0000000		0,0000000		0,0018344	
H(48) :	(V112 => V110)	1,0000000		1,0000000		-0,0012690		0,0006030	
H(49) :	(V122 => V130)	1,0000000		1,0000000		-0,0009691		0,0002999	
H(50) :	(V152 => V148)	1,0000000		1,0000000		0,0012249		0,0021940	
H(51) :	((V127 => (V62 => (V47 => (V39))) => V53)	0,9222231		1,0000000		-0,0036404		-0,0048653	
H(52) :	(V143 => (V69 => V9))	0,9081563		1,0000000		-0,0014645		0,0021760	
H(53) :	(V71 => (V14 => V11))	0,8879040		1,0000000		0,0012433		0,0027077	
H(54) :	((V143 => (V69 => V9)) => V151)	0,8557435		0,9529724		0,0098304		0,0085872	
H(55) :	((V132 => V50) => V126)	0,8051748		1,0000000		-0,0028984		-0,0127289	
H(56) :	(V147 => (V143 => (V69 => V9)) => V151))	0,7961172		0,9250641		0,0014776		0,0043760	
H(57) :	(V92 => V131)	0,7960000		0,8518519		0,0032910		0,0018134	
H(58) :	(V120 => V138)	0,6750000		0,7922078		0,0088091		0,0055182	
H(59) :	(V81 => (V68 => V66))	0,6036811		1,0000000		-0,0037572		-0,0125664	
H(60) :	(V98 => V60)	0,5930000		0,7538462		-0,0023439		0,0014134	
H(61) :	((V37 => V2) => V99)	0,5698019		1,0000000		0,0014276		0,0037715	
H(62) :	(V142 => V145)	0,5360000		0,7278912		0,0020689		0,0006412	
H(63) :	(V105 => (V72 => V103))	0,5241483		1,0000000		0,0069386		0,0048697	
H(64) :	(V149 => (V147 => ((V143 => (V69 => V9)) => V151)))	0,4983198		0,2150341		0,0010501		-0,0005885	
H(65) :	((V13 => V15) => V104)	0,4414381		0,4982699		0,0030556		0,0020054	
H(66) :	((V59 => V31) => V61)	0,4414005		1,0000000		0,1955327		0,1924771	
H(67) :	((V87 => V91) => V95)	0,4234894		0,4658604		-0,0033106		-0,1988433	

Suite page précédente

Suite page suivante

Niveau	Règle	Cohesion $coh_{SUP(n)MGK}(C)$	inter-classe (C)	Implication $\psi(C, C')$	inter-classe	Niveau $s(\Omega, H)$	significatif	Noeud $v(\Omega, H)$	significatif
H(68) :	(V136 => V106)	0,4080000		0,6721311		-0,0023230		0,0009876	
H(69) :	((V28 => V22) => V57)	0,3756286		1,0000000		0,0020731		0,0043961	
H(70) :	(V51 => V102)	0,3620000		0,6521739		0,0031622		0,0010891	
H(71) :	((V121 => V79) => V97)	0,3319021		0,4253308		0,0275905		0,0244283	
H(72) :	((V44 => V8) => V18)	0,3301927		1,0000000		-0,0040759		-0,0316663	
H(73) :	(V86 => V109)	0,2370000		0,5989975		-0,0027307		0,0013452	
H(74) :	(V76 => V146)	0,2160000		0,5897436		-0,0020222		0,0007085	
H(75) :	(V114 => (V30 => V49))	0,1875777		0,3786982		0,0029966		0,0050189	
H(76) :	(V89 => V38)	0,1700000		0,5714286		0,0052805		0,0022838	
H(77) :	(V80 => V101)	0,1080000		0,5454545		0,0164700		0,0111895	
H(78) :	(V90 => (V86 => V109))	0,0907837		0,1378684		-0,0050817		-0,0215518	

Suite page précédente

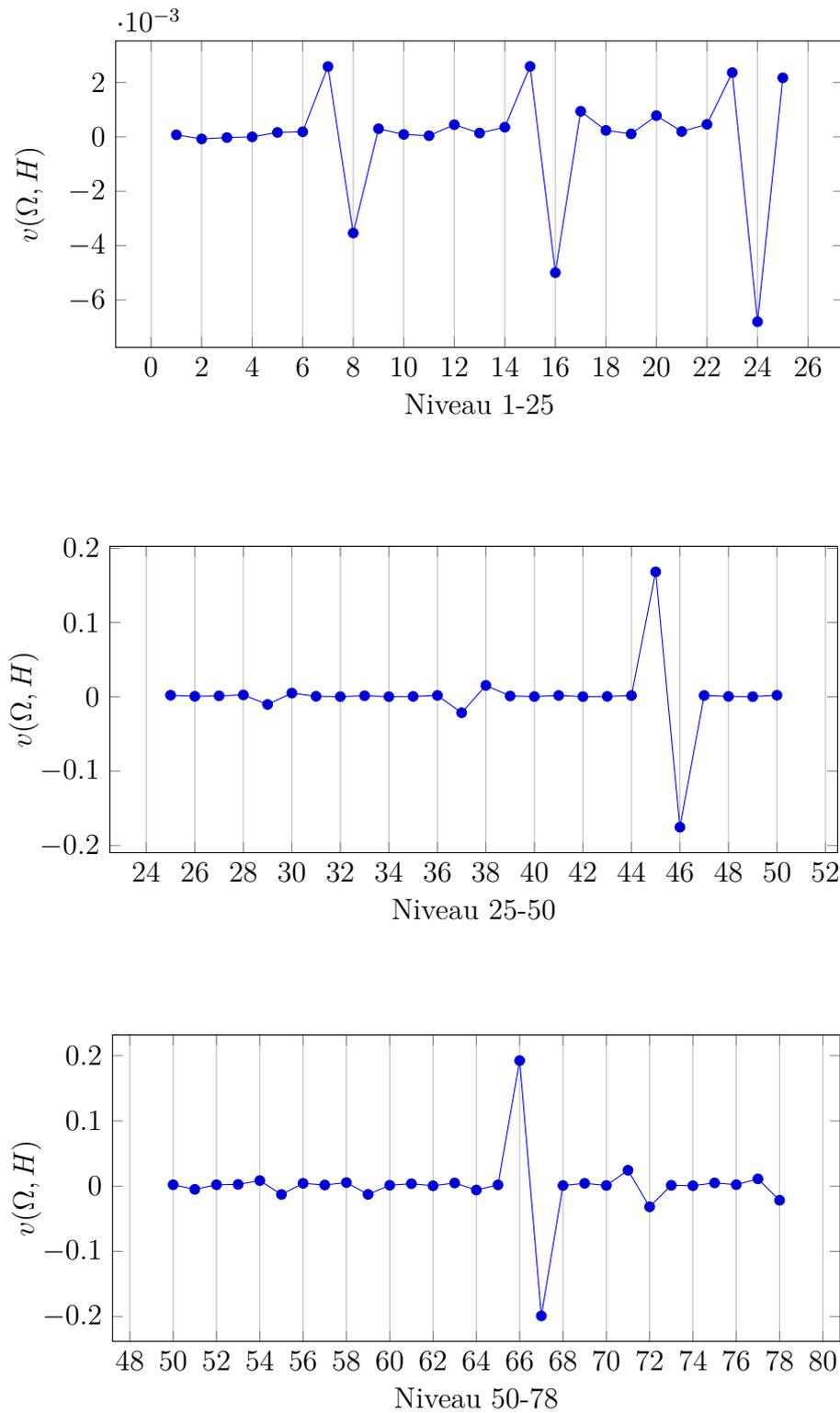


Figure 7.12 – 3 Graphes représentant les nœuds significatifs des 78 classes

Mais tout d'abord, nous allons montrer la signification de chaque variable participant à l'interprétation des règles dans le tableau 7.14 ci-dessous.

Tableau 7.14 – Information sur les variables

Codes	Variabiles	Significations
V1	Homme	Homme
V2	Femme	Femme
V3	Age_20	Agé(e) moins de 20 ans
V9	Bacc_Scien	Série Scientifique
V19	Trav_ordi_freq	Travailler fréquemment à l'ordinateur
V36	Trav_net_jamais	Utilise jamais l'internet
V37	Util_net_ens	Utilise principalement l'internet dans le cadre de formation
V39	Util_net_actu	Utilise principalement l'internet pour suivre l'actualité
V47	PC_net_appr_pasdutout	Ordinateur et/ou internet ne peuvent pas du tout apprendre efficacement
V53	Pb_abonn_cher	Problème due au frais d'abonnement trop cher
V62	Revoi_frais_abonn	Améliorer l'accès au service internet en rabaisant les frais d'abonnement
V69	Info_prog	Avoir suivi une Formation en informatique sur la programmation
V79	Form_net	Intéressé à une Formation en informatique sur l'internet
V86	Videoproj_appr_pasdutout	Le vidéo projecteur ne peut pas du tout servir à apprendre
V90	Retroproj_appr_pasdutout	Le vidéo rétroprojecteur ne peut pas du tout servir à apprendre
V97	Tabnoir_appr_peu	Le tableau noir peut servir à apprendre un peu
V109	Pb_moyen	Problème de moyen
V111	Parler_TIC_non	Pas de connaissance en TIC
V121	Role_PC_net_appr_info_offreincorrect	Ordinateur et/ou internet offrent des reponses dans l'apprentissage de l'informatique
V127	Net_enrich_courinfo_pasdutout	Internet n'enrichit pas du tout le cours informatique
V143	Log_algo_oui	Utiliserun logiciel libre d'apprentissage de l'algorithme
V147	Util_Prog_nation_algo_oui	Utilise un programme national et officiel pour l'algorithme
V149	Util_Prog_etab_algo_oui	Utilise un programme unique avec d'autres établissements
V151	Pb_etu_assim_algo_oui	Avoir rencontrer de problèmes pour l'assimilation de l'algorithme

7.5.2 Interprétation de quelques règles

- R(1) : $(V19 \Rightarrow V1)$ et R(2) : $(V37 \Rightarrow V2)$:

Les filles sont sélectives et se restreignent à l'utilisation de l'internet dans le cadre d'une formation tandis que les hommes travaillent fréquemment et sans restriction au préalable sur l'ordinateur. Les hommes apparaissent ainsi plus ouverts à tous travaux informatiques. Ce constat retrouvé est vécu à la faculté des lettres et des sciences humaines où sont présent beaucoup plus des filles que des garçons. Ce fait implique de fiabilité envers les résultats fournis par notre approche ;

- R(4) : $(V119 \Rightarrow (V36 \Rightarrow V3))$:

Ce sont les étudiants mâles sans expériences de l'internet qui apparaissent plus motivés à travailler sur ordinateur ;

- R(9) : $(V69 \Rightarrow V9)$:

Généralement, des étudiants qui ont leur baccalauréat scientifique qui ont déjà suivi une formation en informatique scientifique (programmation). Les futurs informaticiens potentiels seraient ainsi trouvés parmi les bacheliers scientifiques et non parmi les littéraires ;

- R(27) : $(V127 \Rightarrow (V62 \Rightarrow (V47 \Rightarrow V39)))$:

La classe des utilisateurs d'ordinateur qui ne sentent et ne voient pas l'utilité pédagogique de l'informatique ce sont essentiellement les lecteurs des journaux quotidiens qui souhaitent vivement le rabais de frais d'abonnement à l'internet ;

- R(56) : $(V147 \Rightarrow ((V143 \Rightarrow (V69 \Rightarrow V9)) \Rightarrow V151))$ et

R(64) : $(V149 \Rightarrow (V147 \Rightarrow ((V143 \Rightarrow (V69 \Rightarrow V9)) \Rightarrow V151)))$:

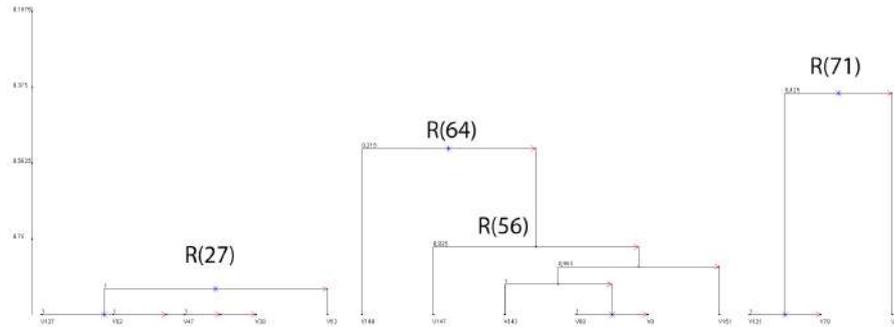
La difficulté des étudiants en informatique utilisant des logiciels d'algorithmique proviendraient du fait se conformer au programme national ou programme commun entre les établissements ce qui interrogerait au besoin de révision au programme de formation en informatique ;

- R(71) : $((V121 \Rightarrow V79) \Rightarrow V97)$:

La formation sur ordinateur (slide) est loin d'être suffisante et ne peuvent pas tout assurer, mais elle doit être compléter une formation par usage d'un tableau. L'usage d'un tableau s'avère être incontournable.

Nous montrons ci-dessous l'extrait du dendrogramme des règles R(27), R(56) et R(64).

Figure 7.13 – Extrait du dendrogramme représentant les méta-règles obtenues.



7.6 Conclusion

Dans ce chapitre, nous avons mené deux études différentes mais complémentaires. La première étude concerne la stratégie à mettre en œuvre pour l'intégration de TIC dans la pratique pédagogique dans la région DIANA, en utilisant la méthode de l'AFM et la CAH avec l'outil logiciel R. Le résultat de cette étude nous paraît alarmante en termes de méconnaissance des élèves/étudiants ainsi que les enseignants sur le TIC, le non accès au TIC et le manque de formation en TIC des acteurs (élève-étudiant-enseignant). La deuxième étude concerne l'analyse et diagnostique sur la connaissance en informatique des étudiants de l'enseignement supérieur à Madagascar, en appliquant notre méthode sur la Classification Hiérarchique Implicative et Cohésitive selon la mesure de qualité M_{GK} et notre outil logiciel CHIC- M_{GK} , plus précisément une étude sur l'alphabétisation informatique des étudiants. Des remarques ont été détectées sur la difficulté des étudiants en informatique utilisant des logiciels d'algorithmique proviendrait du fait se conformer au programme non commun, l'utilisation du tableau noir est toujours d'actualité et complémentaire malgré l'évolution technologique en utilisant la TIC dans la pratique d'enseignement. Nous remarquons que la première méthode adoptée est plus complexe pour l'interprétation ; tandis que la deuxième méthode (la nôtre) est plus adaptée en lisant facilement les règles extraites et surtout sur la présentation des connaissances (méta-règles) sous forme de dendrogramme. Par la suite, nous utilisons toujours notre outil

CHIC- M_{GK} pour faciliter les interprétations des variables liées entre-elles dans les bases de données fouillées.

Chapitre 8

Didactique de l'informatique/algorithmique - Formation en informatique

*“ L’amour, l’argent et
l’algorithme ne naissent pas du
hasard. ”*

Nanan-akassimandou

Sommaire

8.1	Introduction	236
8.2	Didactique de l’informatique	236
8.2.1	Qu’est ce que l’informatique ?	237
8.2.2	La pensée informatique	238
8.3	Formation en informatique dans le modèle anglo-saxon	249
8.3.1	Le modèle anglo-saxon	249
8.3.2	Analyse des thèmes dans CE, CS, IS, IT et SE	251
8.3.3	Analyse des thèmes en CS	256
8.3.4	Analyse des capacités d’aptitudes des étudiants en CE, CS, IS, IT et SE	263
8.3.5	Analyse des capacités d’aptitudes des étudiants en CS	273
8.3.6	Domaines de connaissances du modèle anglosaxon	286
8.3.7	Principes de base du développement logiciel (SDF)	290
8.4	Formation en informatique à l’ENSET	294
8.4.1	Les capacités d’aptitudes des jeunes en licence informa- tique de l’ENSET	294
8.4.2	Les parcours de licence en informatique	295

8.4.3	Analyse du curriculum informatique à l'ENSET	296
8.5	Sur la programmation	301
8.5.1	Qu'est-ce que la programmation?	301
8.5.2	Comparaison du domaine de connaissances SDF en CS à celle de l'ENSET	303
8.6	Sur l'algorithmie	304
8.6.1	Historique	304
8.6.2	Qu'est-ce que l'algorithmie?	305
8.6.3	La pensée algorithmique	308
8.6.4	Procédure de résolution d'un problème	309
8.6.5	Qu'est-ce que l'algorithmie?	310
8.7	Curriculum informatique en classe de Seconde du lycée à Madagascar	311
8.7.1	Le cours sur les généralités sur les systèmes informatiques	314
8.7.2	Le cours sur les applications logiciels	315
8.7.3	Le cours en algorithmie et programmation	316
8.8	Conclusion	320

8.1 Introduction

Comme nous le savons tous que l'école est un lieu d'apprentissages, là où devenu actuellement un lieu contestataire des jeunes face à la monotonie et l'empirisme traditionnel des méthodes de transmission et d'acquisition des connaissances. Comme l'apprentissage est un processus par lequel une personne acquiert des connaissances, maîtrise des habiletés i.e l'apprenant est capable d'une performance nouvelle dont il n'était pas capable au début de l'apprentissage. Apprendre est alors de trouver du sens dans une situation d'enseignement. Si la finalité de l'enseignement/apprentissage est doté l'apprenant d'une capacité à agir dans une nouvelle situation, une réflexion sur les contenus à enseigner s'impose, c'est le travail de la didactique.

8.2 Didactique de l'informatique

La didactique d'une discipline, selon P. Meirieu dans son ouvrage, est « *un ensemble de réflexions et propositions sur les méthodologies à mettre en œuvre pour permettre l'appropriation de contenus spécifiques* ». (Meirieu, 1988).

Selon Arnaud, « *la didactique est une théorie construit de l'exercice qui par un*

ensemble de situations instrumentales finalisées définit pour chaque matière d'enseignement un contenu structuré et hiérarchisé afin de guider les apprentissages scolaires des élèves »(Arnaud, 1989).

Vers la fin des années 70, pendant lesquels la programmation prenait son allure dans le monde universitaire, des questions se focalisaient surtout sur les moyens de bien enseigner cette nouvelle discipline. Ensuite, dans les années 80, des questions de didactique se posaient pendant l'apprentissage de l'option informatique. On constatait que l'apprentissage de la programmation était très difficile à l'époque. Cette difficulté est toujours valable aujourd'hui, malgré des recherches et évolutions technologiques menées par l'homme, face aux abondantes et multitudes informations qu'il entoure. Depuis, des revues informatiques, des articles, des bulletins de liaison sur la didactique de l'informatique, plus précisément sur l'apprentissage de la programmation ainsi que sur les activités permettant leur appropriation ont été publiés ; des associations savantes sur le thème ont été créées ; des colloques, des conférences et des ateliers de didactique de l'informatique ont été organisés dans plusieurs pays ; des thèses sur la didactique de l'informatique ont été soutenues, mais ce qui nous trouve important c'est de faire référence à la transposition didactique en introduisant une partie de l'environnement économique-socio-politico-culturel et de faire face à l'évolution de la TIC qui pourrait faciliter l'apprentissage de l'informatique en tant qu'outil et peut être rendu aussi compliqué l'apprentissage selon les situations de chaque pays. G. L. Baron et E. Bruillard dans leur article posaient la question :

« Informatique, TIC et enseignement, quelle place dans les curricula ? »(Baron et Bruillard, 2001),

que nous allons essayer de répondre par la suite par rapport à notre situation à Madagascar.

8.2.1 Qu'est ce que l'informatique ?

L'informatique est une science du traitement rationnel et automatique de l'information considérée comme le support des connaissances et des communications¹; c'est aussi l'ensemble des applications de cette science². L'informatique est un domaine d'activité scientifique, technique et industriel concernant le traitement automatique

1. Dictionnaire de français : Larousse

2. Informatique dans le dictionnaire de l'académie française, 9ème édition

de l'information par l'exécution de programmes informatiques par des machines. En 1957, l'ingénieur allemand Karl Steinbuch crée le terme « *Informatik* » pour son essai intitulé *Informatik : Automatische Informationsverarbeitung*, pouvant être rendu en français par « *Informatique : traitement automatique de l'information* ». Ce terme a été utilisé pour la première fois en France en 1962, par P. Dreyfus pour son entreprise Société d'informatique appliquée. En 1966, l'Académie française consacre l'usage officiel du mot pour désigner la « science du traitement de l'information ». La presse, l'industrie et le milieu universitaire l'adoptent dès cette époque. En 1968, le ministre fédéral de la Recherche scientifique d'Allemagne, G. Stoltenberg, prononce le mot « *Informatik* » lors d'un discours officiel sur la *nécessité d'enseigner cette nouvelle discipline* dans les universités de son pays ; on emploie ce même terme pour nommer certains cours dans les universités allemandes. Le mot *informatica* fait alors son apparition en Italie et en Espagne, de même qu'*informatics* au Royaume-Uni. A Madagascar, nous conservons cette appellation qui nous paraît difficilement à l'écriture : « in-fôrmatika ?, en-fôrmatika ?... ». Pour les didacticiens, l'informatique

« est considérée comme instrument universel de traitement rationnel du réel » (Linard, 1990)

« s'est constituée un corpus de savoir savant autour d'elle » (Arsac, 1987)

« est devenue progressivement une science socialement reconnue, enseignée à l'université » (Baron, 1981).

Cela évoque trois dimensions de l'informatique liées et autonomes : les *dispositifs qui en sont issus*, les *équipements qui la représente* et l'informatique en tant que *science socialement reconnue*.

8.2.2 La pensée informatique

Avant la vulgarisation des micro-ordinateurs vers les années 70, l'approche informatique est été marquée par une **démarche de pensée** et les chercheurs ont axé leur recherche en didactique de l'informatique « *recherche pédagogique* ». On attribut à l'informatique la capacité de favoriser, d'accélérer le développement de l'esprit d'analyse, de rigueur de la pensée, de créativité grâce aux possibilités de concrétisation des imaginaires, le développement de l'intelligence qui pourrait s'accélérer. Ainsi, on peut déjà mettre en œuvre une réflexion sur le fondement de l'informatique

et les aspects philosophiques et épistémologiques.

L'informatique – démarche de pensée consiste à

- classer selon les critères bien précis ;
- ordonner les différentes phases d'une action ;
- automatiser des actions qui se répètent et qui sont similaires.

La révolution numérique a pour racine le calcul numérique, i.e le calcul automatique sur des informations encodées numériquement. On décrira ici les différents principes du calcul automatique et leur réalisation dans des modèles de calculs, *objets à la fois intuitifs et mathématiques* au centre de la pensée informatique. Ces modèles mathématiques permettent de mieux maîtriser la transition entre notre pensée, créatrice mais lente et pas toujours rigoureuse, et les circuits électroniques, ultra-rapides et exacts mais totalement souples. **Penser, modéliser et maîtriser** le calcul informatique. Wing rappela dans son ouvrage que

« la pensée informatique s'est élaborée à partir de l'analyse de la puissance et de limites des processus informatiques, qu'ils soient exécutés par des machines ou par des humains. »(Wing, 2006)

La résolution des problèmes et la conception des systèmes qu'aucun être humain ne serait capable de voir tout seul se font à l'aide des méthodes et des modèles informatique. La pensée informatique n'oppose pas au secret de l'intelligence de la machine. *L'homme réalise mieux qu'une machine ?* ou vice versa.

Face à un problème donné, on peut se demander s'il est complexe ou non et comment le résoudre ?. L'informatique en s'appuyant sur des méthodes solides répondra à ces deux questions. Affirmer la difficulté d'un problème prend en compte la puissance de la machine (dispositif de calcul) qui produira la solution et doit prendre en compte l'ensemble des instructions de la machine, ses contraintes de ressources et son environnement d'exploitation. Pour résoudre un problème efficacement, on peut se demander si une solution approximative est assez bonne, si on peut utiliser l'aléatoire à son avantage et si des faux positifs ou des faux négatifs sont autorisés.

Dans Wing (2006), « *Penser informatiquement* :

- *c'est reformuler un problème apparemment difficile en un problème que nous savons résoudre, par exemple par réduction, par plongement, par transformation ou par simulation.*

- *c'est penser récursivement ; c'est penser parallélisme ; c'est interpréter des données comme du code et du code comme des données ; c'est vérifier des types comme une généralisation de l'analyse dimensionnelle ; c'est reconnaître les vertus et les dangers des alias qui consistent à donner plus d'un nom à quelqu'un ou à quelque chose ; c'est reconnaître à la fois le coût et la puissance de l'adressage indirect et des appels de procédure ; c'est apprécier un programme non seulement pour sa correction et son efficacité, mais pour son esthétique et c'est évaluer la qualité de la conception d'un système par sa simplicité et son élégance.*
- *c'est utiliser l'abstraction et la décomposition quand il s'agit d'affronter une tâche large et complexe ou quand il s'agit de concevoir un système large et complexe ; c'est séparer les différents aspects ; c'est choisir une représentation appropriée pour un problème ou modéliser les aspects pertinents d'un problème pour les rendre accessibles ; c'est utiliser des invariants pour décrire succinctement et déclarativement le comportement d'un système ; c'est avoir confiance que l'on peut sans danger utiliser, modifier et influencer un système large et complexe sans en comprendre tous les détails ; c'est modulariser en anticipant de nombreux utilisateurs, ou stocker des informations en prévoyant des utilisations à venir.*
- *c'est envisager le pire événement en pensant en terme de prévention, de protection et de récupération ; cela se fait par la redondance, le contrôle des dommages et la correction d'erreurs ; penser informatiquement c'est parler d'un interblocage quand il s'agit d'un embouteillage et parler d'interfaces quand il s'agit de contrats ; c'est éviter de faire jouer les coudes quand on peut synchroniser des rendez-vous.*
- *c'est utiliser des raisonnements heuristiques pour trouver une solution ; c'est faire des plans, apprendre et ordonnancer en présence d'incertitude ; c'est chercher, chercher et chercher encore pour produire une liste de pages web, une stratégie gagnante ou un contre-exemple ; c'est utiliser une masse de données pour accélérer*

un calcul ; c'est faire un compromis entre temps et espace et entre puissance de calcul et capacité de stockage(...)»

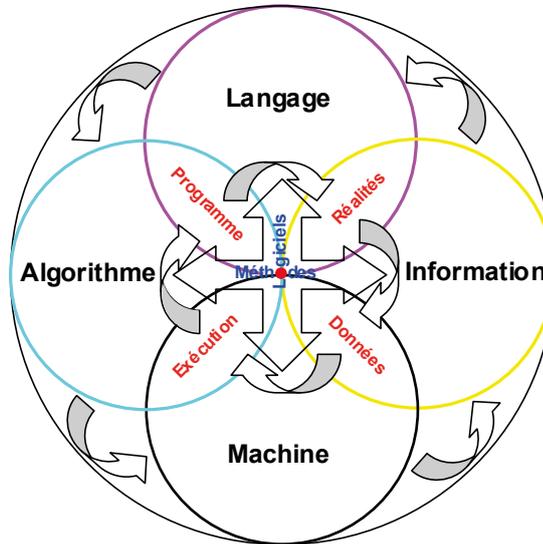


Figure 8.1 – La pensée informatique

En avril 2007, Paul Mathias a évoqué dans l'introduction de sa communication intitulée - « *Y a-t-il une pensée informatique ?* »

« (...) qu'il n'est effectivement pas possible de restreindre l'analyse de la pensée informatique aux circonstances technico-épistémiques de son développement. Car son développement n'est pas simplement scientifique, il est dans le sens le plus large et indéterminé du terme humain. »

L'informatique — et par conséquent ses modes de pensée — concerne notre « *vie* », car elle touche nos gestes quotidiens, nos activités sociales et nos activités intellectuelles. La pensée informatique est alors « *la vie humaine au quotidien* ».

Par rapport à cela, on peut dire que la pensée informatique c'est utile au quotidien et allons nous apprendre cette pensée philosophique (pensée informatique) à nos enfants, à nos jeunes ainsi qu'à nous même sans distinction que ce soit informaticien ou pas. La pensée informatique est une doctrine qui embrasse tout dans la vie humaine et son environnement.

« Cette façon de pensée fera partie du bagage non seulement des autres scientifiques, mais aussi de tout un chacun. L'informatique ubiquitaire

qui était le rêve d'hier devient la réalité d'aujourd'hui et la pensée informatique sera la réalité de demain », dixit Jeannette Wing.³

Plusieurs problèmes de la didactique de l'informatique auraient pu être résolus grâce au concept fondamental de « la pensée informatique ». Ça c'est du côté humain. Par contre, du côté machine, Claud Pair dans son discours de clôture d'un colloque international⁴ disait :

« (...) Il est vrai qu'un système informatique - matériel + logiciel - est un objet technique assez caractéristique de la civilisation moderne ; on peut dire interactif, entrant comme composant dans d'autres objets techniques, et utilisant une représentation symbolique des réalités sous formes d'information. Il est donc légitime de demander à l'école d'introduire les élèves à l'usage d'un tel objet technique, ne serait-ce que pour assurer une certaine égalité entre les jeunes, quels que soient leur milieu social et leurs sexes⁵. C'est en particulier l'une des justifications de la présence de systèmes informatiques dans l'enseignement de la technologie au collège. Mais ce seul aspect ne justifierait pas la présence de l'informatique pendant l'ensemble des études.

L'ordinateur est aussi un outil d'enseignement, au service des diverses disciplines. Par exemple, en mathématiques, employer des logiciels qui évitent des manipulations formelles devrait permettre de mettre davantage l'accent sur la signification des notions plutôt que sur leur traitement syntaxique, ou sur l'activité de poser des problèmes plutôt que sur la résolution d'équations en dehors de tout contexte. Mais une telle révolution implique des changements considérables dans les contenus, dans l'importance relative des notions, et notamment dans les didactiques. L'emploi de l'outil informatique dans diverses disciplines peut inciter à de tels changements, de toute manière indispensables ; il renforce aussi le contact avec l'objet technique-ordinateur. Mais il ne s'agit pas réellement d'informatique, et les évolutions didactiques dont nous parlons

3. Voir Newsletter Interstices.info - Algorithmes - publié le 29/05/2009 - <https://interstices.info/la-pensee-informatique/>, consulté le 07/01/2019

4. EPI (1989). Colloque francophone sur la didactique de l'informatique, Université René Descartes, Paris les 1er, 2 et 3 septembre 1988, p.302, https://edutice.archives-ouvertes.fr/edutice-00374950/file/Allocution_de_cloture.pdf, consulté le 15/06/2018

5. Informatique Pour Tous

demanderaient des recherches : informatique et didactique de telle ou telle discipline(...) »

Nous devons analyser et répondre d'abord sur les (aux) questions de réflexions de tous les jours des chercheurs que « *l'informatique est-il un outil ou objet d'enseignement ? Une discipline ou non ? Un langage de demain ? Une culture spécifique ?* » Le fait que l'informatique soit une science signifie autre chose. Une science se caractérise par des faits, des concepts, des réflexes, des méthodes, et la question que l'enseignement devrait poser à chaque science porte sur ce qu'elle peut apporter à la formation générale de l'esprit des élèves. Faits, concepts, réflexes, méthodes : Il est plus facile d'enseigner des faits que des concepts ; mais, de plus en plus, et la technique informatique y est pour quelque chose, on peut retrouver les faits pourvu qu'on possède concepts et méthodes. Concepts, réflexes, méthodes s'enseignent moins - par le maître - qu'ils ne s'apprennent par l'élève. Mais si on parvient en général assez bien à faire acquérir des concepts et des réflexes, fait-on vraiment, dans les disciplines traditionnelles, acquérir des méthodes ? Or les méthodes, comme nous les savons par expérience que cela ne s'enseigne pas à proprement parler ; il s'agit de savoir plutôt comment favoriser leur apprentissage. On se plaint volontiers tout au long de la scolarité que les élèves n'ont pas de méthodes, mais toujours comme si elles avaient dû être acquises dans les années précédentes, et l'être par simple imprégnation.

Voilà une cause de l'échec scolaire, notamment pour les populations chez qui, traditionnellement, on ne faisait pas d'études longues.

Mais cette insistance sur les méthodes est sans doute aussi ce que la discipline informatique peut apporter de plus riche à la formation des jeunes. Il faut cependant être conscient que ce n'est pas simple, parce que justement nous manquons de traditions en ce domaine. Comment peut-on faire acquérir des méthodes, notamment en informatique ? Des travaux de recherche apportent des éléments pour construire des réponses. Promouvoir l'informatique dans l'enseignement fondamental est indispensable, mais reste à vérifier.

Romainville cita les leçons et difficultés sur cette alphabétisation informatique ainsi que de définir les contenus et les méthodes liées à cette initiative (Romainville, 1988). Certains auteurs ont écrit dans leur exposé de motifs en insistant sur l'éveil au phénomène socio-culturel et technologique que constitue l'informatique. Il vaut mieux

d'apprendre aux jeunes enfants d'abord le rôle, les effets positifs et négatifs de l'informatique avant de les familiariser avec la « MACHINE »⁶ les composantes et sa structure. Cela concerne la structure de base, les composantes de base et la technologie de base de la machine afin d'ignorer la vitesse hâtive de l'évolution technologique. Des documents analysés insistent sur la « *programmation-prétexte* ».

« Car partout, il est bien rappelé que ce n'est pas l'apprentissage d'un langage qui est visé, mais bien le développement des capacités cognitives logiques supposées être la base de tout travail de programmation structurée. Ces « compétences » de résolution de problèmes sont, entre autres, les capacités à poser un problème, à le décomposer en sous-problèmes, à planifier des actions, à anticiper des résultats (formuler des hypothèses), (...) »(Romainville, 1988)

Un des relevés des plus complets des effets cognitifs attendus de l'activité de programmation a été réalisé par Feurzeig

«...la rigueur de la pensée et la précision de l'expression ; la compréhension de concepts généraux tels que variable, fonction ; une plus grande facilité dans l'utilisation d'heuristiques, c'est-à-dire d'approches générales de résolution de problèmes telles que la planification, la décomposition en sous-problèmes ; la conception du « debugging » (recherche des erreurs) comme une étape constructive et planifiée ; l'attitude générale de chercher une solution à un problème important en construisant petit à petit des solutions partielles ; la réflexion et la prise de conscience des processus de résolution de problèmes ; enfin, la prise de conscience du fait qu'il n'y a que rarement une seule « bonne » solution mais plus souvent différents chemins avec chacun leurs avantages et inconvénients. »(cité par (Pea et Kurland, 1984) p.143)

La rigueur, résultant de la démarche algorithmique, fondement de l'informatique constitue un apport indispensable à la formation des élèves. « ALGORITHME »⁷. La question du rôle du contexte dans l'apprentissage de la programmation est complexe parce qu'il ne s'agit pas d'une compétence unique. Tout comme la lecture, elle englobe un grand nombre d'habiletés qui interagissent avec les connaissances

6. 1^{ère} composante de la pensée informatique

7. 2^{ème} composante de la pensée informatique

antérieures de l'élève, sa mémoire, ses capacités de traitement de l'information, ses stratégies générales de résolution de problèmes telles que l'inférence, la génération d'hypothèses, etc.. L'entraînement à la lecture suppose une gamme étendue d'expériences avec différents genres (narrations, essais, poésie, débat) et avec différents buts. Tout comme la lecture est souvent assimilée à l'habileté de décodage, l'apprentissage de la programmation est souvent équivalent à l'apprentissage d'une syntaxe et d'un vocabulaire d'un « LANGAGE »⁸ ; alors que l'entraînement à la programmation, comme la lecture, est complexe et fortement dépendante du contexte. « *Nous devons donc commencer par préciser le contexte dans lequel la programmation a été mise en pratique et apprise* » (Pea et Kurland, 1984). C'est bien démontré l'existence d'une richesse des interactions entre des disciplines comme : la programmation, la technologie de l'éducation et la psychologie de l'apprentissage (Hauglustaine-Charlier, 1989). Au travers des interactions entre les spécialistes de ces différents domaines qu'on puisse faire progresser nos connaissances à propos de l'apprentissage et de la didactique de l'informatique. Donnay montra l'utilisation des images pour apprendre à programmer et évalua les connaissances des élèves à partir des questionnaires. Ensuite il parla la relation entre les métaphores et apprentissage et nota la définition de ce dernier comme un changement durable dans les structures cognitives de l'individu (Donnay, 1988). Les informations (verbales, visuelles, auditives, motrices) seraient stockées en mémoire sous forme de réseaux non linéaires. Le réseau élaboré à propos d'un concept serait propre à la personne, il dépendrait de son histoire individuelle. Il serait transformable ce qui permettrait l'apprentissage. Un changement à ce niveau impliquerait un investissement affectif indispensable. Ce changement nécessiterait en outre, une activité de l'individu, activité de recherche d'informations, d'analogies, de discriminations, de formulation d'hypothèses ... Un apprentissage serait donc une construction active, individuelle, nécessitant un investissement affectif. La durabilité du changement réalisé pourrait dépendre de répétitions, de la variété des canaux utilisés pour transmettre une information et de la profondeur du traitement réalisé par l'apprenant. L'apprentissage particulièrement chez l'adulte consiste à intégrer de nouvelles informations dans une structure mnémorique existante (déjà plus structurée que chez l'enfant). Les images visuelles font partie du réseau sémantique. Dans le domaine de connaissances qui

8. 3^{ème} composante de la pensée informatique

nous occupe, il ne s'agit vraisemblablement pas de reproductions fidèles de la réalité mais bien de représentations possibles de celle-ci le plus souvent simplifiées ou mieux modélisées. Dans le processus dynamique de traitement de l'information, la création d'images mentales, la recherche d'analogies, exerce certainement un rôle essentiel. Lorsqu'un apprenant découvre une description de l'ordinateur en ces termes « l'ordinateur est une machine à traiter l'information... », il peut imaginer une machine sous différentes formes, douée de plus ou moins de compétences. La création d'images mentales est peut-être une des formes initiales de traitement de l'information chez certains sujets, l'analogie, la transposition constituent une première marque de compréhension. La formation d'images mentales s'avère un moyen d'apprentissage très efficace ... susceptible de favoriser la rétention d'informations (Norman et Lindsay, 1977) « INFORMATION »⁹.

Apprendre la programmation diffère de l'apprentissage d'un langage. - Programmer c'est faire-faire une tâche par un exécutant. Ce faire-faire prend la forme d'une marche à suivre destinée à cet exécutant. - Il s'agit de faire-faire une tâche et non de résoudre un problème. - Les étapes obligées du traitement informatique d'une tâche sont : Quoi faire ? (spécifications), Comment faire ? (stratégies), Comment faire-faire ? (rédaction des algorithmes), Comment dire ? - Caractéristiques d'une marche à suivre et organisation de celle-ci : instructions d'action, structures de contrôle, commentaires. - L'ordinateur est un exécutant gestionnaire de casiers (variables), communicant avec l'extérieur, capable d'effectuer un nombre limité d'instructions et ayant quelques outils à sa disposition. - Démarche descendante et structurée et formalisation de celle-ci dans la phase d'analyse. Le public concerné étant surtout un public d'adultes et en grande partie d'enseignants disposant d'un temps limité à consacrer à la formation, l'accent est placé sur la découverte de concepts fondamentaux, sur l'acquisition d'une méthode de travail et enfin sur le développement d'un mode de pensée nouveau. Ces compétences étant (peut-être) celles qui sont et seront transférables quels que soient le langage et la machine utilisés.

La didactique de l'informatique a un statut à part dans la mesure où elle est concernée à deux titres par les Nouvelles Technologies de l'Information NTI. D'abord, en tant que discipline qui réfléchit aux modes de transmission des savoirs et savoir-faire de la science informatique (enseignement de la programmation), ensuite en

9. 4^{ème} composante de la pensée informatique

tant que concepteur de formalismes informatiques et de logiciels compatibles avec cet enseignement (développement et génie logiciel). Si l'informatique y poursuit une carrière de discipline d'enseignement supérieur, au niveau scolaire, l'intérêt institutionnel s'est focalisé sur l'intégration des logiciels dans les disciplines existantes. L'essentiel des recherches de ce domaine porte sur les méthodes et la pertinence de l'enseignement de la programmation, ainsi que sur l'évolution des langages de programmation et des logiciels professionnels. L'enseignement de l'informatique pose à nouveau un vieux problème : à côté de sa spécificité didactique ou - si l'on préfère - de la méthodologie d'enseignement qui lui est propre ou est propre au groupe de disciplines auquel il s'apparente, quelle place fait-il réellement à une méthodologie générale nourrie des acquis des sciences de l'éducation ?

Question préliminaire : existe-t-il dès maintenant un corpus articulé et jugé suffisant de règles en matière de didactique de l'informatique ? Reposent-elles sur des bases expérimentales éprouvées ou sur l'expérimentation, c'est-à-dire sur des règles empiriques, voire des recettes tirées d'essais sur le terrain, sans souci de contrôle rigoureux ou de représentativité de sous-populations ? (Gilbert De Landsheere, 1988)¹⁰.

Le fait que l'informatique soit une science signifie autre chose. Une science se caractérise par des faits, des concepts, des réflexes, des méthodes, et la question que l'enseignement devrait poser à chaque science porte sur ce qu'elle peut apporter à la formation générale de l'esprit des élèves. Faits, concepts, réflexes, méthodes : Il est plus facile d'enseigner des faits que des concepts ; mais, de plus en plus, et la technique informatique y est pour quelque chose, on peut retrouver les faits pourvu qu'on possède concepts et méthodes. Concepts, réflexes, méthodes s'enseignent moins - par le maître - qu'ils ne s'apprennent par l'élève. Mais si on parvient en général assez bien à faire acquérir des concepts et des réflexes, fait-on vraiment, dans les disciplines traditionnelles, acquérir des méthodes ? On se plaint volontiers tout au long de la scolarité que les élèves n'ont pas de méthodes, mais toujours comme si elles avaient dû être acquises dans les années précédentes, et l'être par simple imprégnation. Voilà sans doute une autre cause de l'échec scolaire, notamment pour les populations chez qui, traditionnellement, on ne faisait pas d'études longues. Peut-être l'enseignement

10. 1988. Gilbert De Landsheere reçoit le Prix José Vasconcelos, considéré comme un prix Nobel dans le domaine de l'éducation. Les prix du Conseil Culturel Mondial jouissent d'une très grande réputation et sont considérés parmi les distinctions les plus prestigieuses qu'une personnalité puisse recevoir.

technique, par sa nature même et aussi parce qu'il accueille prioritairement ces populations, peut-il relever en priorité ce défi de l'apprentissage méthodologique. C'est une des idées de fond que Robert Chapuis avait apporté. Mais cette insistance sur les méthodes est sans doute aussi ce que la discipline informatique peut apporter de plus riche à la formation des jeunes. Il faut cependant être conscient que ce n'est pas simple, parce que justement nous manquons de traditions en ce domaine. Comment peut-on faire acquérir des méthodes, notamment en informatique ? Quelques travaux apportent des éléments pour construire des réponses, même si le titre pouvait être quelque peu trompeur : "didactique" en effet attire l'attention plus sur le maître qui enseigne que sur l'élève qui apprend. Or les méthodes, ne s'enseignent pas à proprement parler ; il s'agit de savoir plutôt comment favoriser leur apprentissage. On distingue habituellement deux domaines de recherche dans l'informatique éducative :

- l'informatique « **objet** » considérée comme une discipline.
- l'informatique « **outil** » intervenant dans d'autres disciplines.

Nous nous intéressons ici essentiellement à l'informatique « **objet** », parce que nous pensons que ces deux domaines sont liés :

1. d'une part, nous faisons l'hypothèse que « *l'utilisation de l'informatique dans l'enseignement d'une autre discipline, comme les mathématiques, nécessite une alphabétisation réelle en informatique* » (Rogalski, 1985). Certaines opérations de pensée, qui ne sont pourtant pas propres à l'informatique, peuvent avoir dans cette discipline un caractère nécessaire ou explicite : la structuration des données, l'explicitation des procédures de résolution en sont des exemples. Ce caractère explicite devrait être producteur d'acquisitions chez les élèves. Or les travaux des psychologues Pea et Kurland (1984) mettent en évidence que l'apprentissage de la programmation n'a d'influence sur le développement général de la connaissance que s'il est d'assez haut niveau ;
2. d'autre part, le réinvestissement dans d'autres domaines des acquisitions résultant d'apprentissages en informatique dépend des activités rencontrées dans cet apprentissage.

Un des résultats essentiels de la psychologie cognitive consiste à avoir mis en évidence le rôle éminent de la résolution de problème dans le processus d'acquisition

des connaissances. Dans cette perspective, la connaissance s'acquiert et se construit par résolution de problèmes. C'est dans cet esprit que nous avons conçu nos séquences d'enseignement. L'étude de l'acquisition des concepts élémentaires propres à l'informatique et l'analyse de leur enseignement présentent donc une importance non seulement du point de vue de la connaissance didactique ; mais du point de vue de la pratique pédagogique. Cela nous amène à faire des analyses sur les contenus de la formation en informatique dans les modèle anglo-saxon et comparés avec le modèle de l'ENSET.

8.3 Formation en informatique dans le modèle anglo-saxon

Aujourd'hui, l'informatique est presque toujours présente dans de nombreux domaines liés aux activités humaines. A Madagascar, la plus grande partie de la population n'utilise pas l'informatique dans un cadre privé comme dans des pays développés. Cela a un impact dans la formation en informatique et exige surtout des analyses approfondies et des innovations adaptées à la situation du pays. Alors, il nous est opportun de voir le modèle de formations en informatique anglosaxon. Une partie de l'étude a été présentée dans ([Rakotomalala et Totohasina, 2019](#)).

8.3.1 Le modèle anglo-saxon

Aux Etat-Unis, quatre organisations appuyées par l'Etat établissent des normes et décrivent les formations en informatique dans l'enseignement supérieur dans le modèle anglo-saxon, telles que : l'ACM (Association for Computing Machinery), l' AIS (Association for Information Systems), l'AITP (Association of Information Technology Professionals) et l'IEEE-CS (Computer Society of the Institute for Electrical and Electronic Engineers).

L'informatique est aussi une discipline universitaire. Malgré le mot « science », l'informatique n'est pas typiquement une science. En effet, le mot « *science* » est plus souvent associé à une discipline basée sur l'étude d'un phénomène réel, l'observation du phénomène et la construction de modèles l'expliquant le plus fidèlement possible. Comme la mathématique, l'informatique n'étudie pas les phénomènes réels. Ces

deux disciplines ont le privilège de pouvoir construire leur propre monde sous la forme d'objets abstraits. En mathématique, il s'agit de nombres, de relations, de fonctions, de transformations, etc. En informatique, on manipule des algorithmes, des programmes, des arbres, des preuves, des systèmes de réécriture, des images numériques et des graphes.

8.3.1.1 Les disciplines informatiques dans le modèle anglo-saxon

Le rapport technique de l'ACM/AIS/IEEE-CS ([Shackelford et al., 2005](#)) a décrit cinq (5) branches de disciplines de l'informatique à l'université :

1. *Computer Engineering* (CE) dont l'application dominante concerne les systèmes embarqués (développement de dispositifs tant sur le plan logiciel que matériel) ;
2. *Computer Science* (CS) qui fournit des bases complètes pour permettre aux diplômés de s'adapter aux nouvelles technologies et aux nouvelles idées ;
3. *Information Systems* (IS) pour lesquels les jeunes diplômés doivent être performants en spécification, conception, et implémentation afin de déterminer les besoins d'une organisation ;
4. *Information Technology* (IT) qui a pour objectif de comprendre les systèmes informatiques et leurs logiciels et s'engage à résoudre tous les problèmes en rapport avec ces systèmes ;
5. *Software Engineering* (SE) dont l'objectif est de former un personnel capable de gérer des projets de logiciels importants, complexes et fiables.

Chaque discipline a ses propres objectifs de formation, mais de nombreuses connaissances et aptitudes sont transversales.

Zone de connaissances	CE		CS		IS		IT		SE	
	min	max								
Fondamentaux de programmation	4	4	4	5	2	4	2	4	5	5
Programmation intégrative	0	2	1	3	2	4	3	5	1	3
Algorithmes et complexité	2	4	4	5	1	2	1	2	3	4
Architecture informatique et organisation	5	5	2	4	1	2	1	2	2	4
Principes et conception des systèmes d'exploitation	2	5	3	5	1	1	1	2	3	4
Configuration et utilisation des systèmes d'exploitation	2	3	2	4	2	3	3	5	2	4
Net Centric Principes et Design	1	3	2	4	1	3	3	4	2	4
Net Centric Utilisation et configuration	1	2	2	3	2	4	4	5	2	3
Technologies de plateforme	0	1	0	2	1	3	2	4	0	3
Théorie des langages de programmation	1	2	3	5	0	1	0	1	2	4
Interaction homme machine	2	5	2	4	2	5	4	5	3	5
Graphisme et visualisation	1	3	1	5	1	1	0	1	1	3
Systèmes intelligents (AI)	1	3	2	5	1	1	0	0	0	0
Théorie de la gestion de l'information (DB)	1	3	2	5	1	3	1	1	2	5
Pratique de la gestion de l'information (DB)	1	2	1	4	4	5	3	4	1	4
Calcul scientifique (Méthodes numériques)	0	2	0	5	0	0	0	0	0	0
Légal / Professionnel / Éthique / Société	2	5	2	4	2	5	2	4	2	5
Développement de systèmes d'information	0	2	0	2	5	5	1	3	2	4
Analyse des besoins de l'entreprise	0	1	0	1	5	5	1	2	1	3
E-business	0	0	0	0	4	5	1	2	0	3
Analyse des exigences techniques	2	5	2	4	2	4	3	5	3	5
Fondations d'ingénierie pour SW	1	2	1	2	1	1	0	0	2	5
Economie d'ingénierie pour SW	1	3	0	1	1	2	0	1	2	3
Modélisation et analyse de logiciels	1	3	2	3	3	3	1	3	4	5
Conception de logiciels	2	4	3	5	1	3	1	2	5	5
Vérification et validation de logiciels	1	3	1	2	1	2	1	2	4	5
Evolution logicielle (maintenance)	1	3	1	1	1	2	1	2	2	4
Processus logiciel	1	1	1	2	1	2	1	1	2	5
Qualité du logiciel	1	2	1	2	1	2	1	2	2	4
Ingénierie des systèmes comparés	5	5	1	2	0	0	0	0	2	3
Logique numérique	5	5	2	3	1	1	1	1	0	3
Systèmes embarqués	2	5	0	3	0	0	0	1	0	4
Systèmes distribués	3	5	1	3	2	4	1	3	2	4
Sécurité : problèmes et principes	2	3	1	4	2	3	1	3	1	3
Sécurité : implémentation et management	1	2	1	3	1	3	3	5	1	3
Administration des systèmes	1	2	1	1	1	3	3	5	1	2
Gestion des systèmes d'information Org.	0	0	0	0	3	5	0	0	0	0
Intégration de systèmes	1	4	1	2	1	4	4	5	1	4
Développement de médias numériques	0	2	0	1	1	2	3	5	0	1
Support technique	0	1	0	1	1	3	5	5	0	1

Tableau 8.1 – Tableau comparatif des poids des thèmes informatiques dans les cinq types de programmes d'études, tiré du (Shackelford et al., 2005, p.24)

8.3.2 Analyse des thèmes dans CE, CS, IS, IT et SE

Le rapport ACM/AIS/IEEE (Shackelford et al., 2005) définit 40 **thèmes** et indique leur poids dans chacune des cinq **disciplines** informatiques. Le poids est caractérisé par deux valeurs comprises entre 0 (faible) et 5 (élevée) qui représentent l'importance d'un thème pour le **domaine**. Ces deux valeurs indiquent respectivement l'exigence minimale et maximale attendue par la discipline relativement aux autres.

8.3.2.1 Liens implicatifs entre les thèmes informatiques dans les cinq types de disciplines d'études

En considérant du tableau 8.1, nous pouvons construire un tableau 8.2 binaire des relations entre les poids minimum et maximum de chaque thème dans les cinq types de programmes d'études : CE, CS, IS, IT et SE ci-dessous.

	Principes de programmation	Programmation ... intégrative	Intégration de systèmes	Développement de médias numériques	Support technique
CEmin0	0	1	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮
CEmin5	0	0	0	0	0
CEmax0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮
CEmax5	0	0	0	0	0
CSmin0	0	0	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮
CSmax5	1	0	0	0	0
ISmin0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮
ISmax5	0	0	0	0	0
ITmin0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮
ITmax5	0	1	1	1	1
SEmin0	0	0	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮
SEmax5	1	0	0	0	0

Tableau 8.2 – Tableau binaire extrait du tableau 8.1 pour les disciplines CE,CS, IS, IT et SE

Nous avons 780 règles d'association valides selon M_{GK} , avec un seuil de risque de 10% fixé par nous-même, dont parmi elles, 508 sont des règles positives et 272 sont des règles négatives, avec $0.04 \leq \text{supp}_{(n)M_{GK}}^f \leq 0.76$ qui forment 32 Paires Orientées PO, i.e. $\text{card}(\text{coh}_{\text{supp}_{(n)M_{GK}}}(\underline{C})) = 32$ avec $0.048 \leq \text{coh}_{\text{supp}_{(n)M_{GK}}}(\underline{C}) \leq 0.607$. Le tableau 8.3 nous montre les valeurs des cohésions inter-classes $\text{coh}_{\text{supp}_{(n)M_{GK}}}(\underline{C})$, des implications inter-classes $\psi(\underline{C}, \underline{C}')$, des niveaux $s(\Omega, H)$ et nœuds significatifs $v(\Omega, H)$ de chaque niveau de la hiérarchie H . Nous présentons aussi dans le tableau 8.4, les 15 méta-règles avec un risque de se tromper à $\alpha = 10\%$, pour les programmes d'études CE,CS, IS, IT et SE.

Niveau	Cohesion inter-classe $\text{coh}_{\text{supp}_{(n)M_{GK}}}(\underline{C})$	Implication inter-classe $\psi(\underline{C}, \underline{C}')$	Niveau significatif $s(\Omega, H)$	Noeud significatif $v(\Omega, H)$
1	0,607	0,76000000	0,00111029	0,00111029
2	0,607	0,76000000	0,00000071	-0,00110958
3	0,607	0,76000000	-0,00036867	-0,00036938
4	0,40759346	0,31912020	0,00022463	0,00059330
5	0,334	0,64000000	-0,00073853	-0,00096316
6	0,334	0,64000000	-0,00079123	-0,00005270
7	0,334	0,64000000	-0,00083093	-0,00003970
8	0,26703568	0,16736072	0,00010680	0,00093773
9	0,21349717	0,31912020	-0,00059377	-0,00070056
10	0,09163815	0,15627155	-0,00066196	-0,00006820
11	0,048	0,52000000	-0,00096953	-0,00030757
12	0,048	0,52000000	-0,00097764	-0,00000811
13	0,048	0,52000000	-0,00098493	-0,00000729
14	0,048	0,52000000	-0,00099138	-0,00000645
15	0,048	0,52000000	-0,00099725	-0,00000587

Tableau 8.3 – Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau.

Regle	
R(1)=	(Analyse des exigences techniques \Rightarrow Interaction homme machine)
R(2)=	(Qualité du logiciel \Rightarrow Evolution logicielle (maintenance))
R(3)=	(Administration des systèmes \Rightarrow Sécurité : implémentation et mgt)
R(4)=	((Qualité du logiciel \Rightarrow Evolution logicielle (maintenance)) \Rightarrow Vérification et validation de logiciels)
R(5)=	(Principes et conception des systèmes d'exploitation \Rightarrow Algorithmes et complexité)
R(6)=	(Net Centric Principes et Design \Rightarrow Configuration et utilisation des systèmes d'exploitation)
R(7)=	(Support technique \Rightarrow Développement de médias numériques)
R(8)=	(Processus logiciel \Rightarrow ((Qualité du logiciel \Rightarrow Evolution logicielle (maintenance)) \Rightarrow Vérification et validation de logiciels))
R(9)=	(Légal / Professionnel / Éthique / Société \Rightarrow (Analyse des exigences techniques \Rightarrow Interaction homme machine))
R(10)=	(Conception de logiciels \Rightarrow (Principes et conception des systèmes d'exploitation \Rightarrow Algorithmes et complexité))
R(11)=	(Net Centric Utilisation et configuration \Rightarrow Programmation intégrative)
R(12)=	(Systèmes intelligents (AI) \Rightarrow Graphisme et visualisation)
R(13)=	(Modélisation et analyse de logiciels \Rightarrow Théorie de la gestion de l'information (DB))
R(14)=	(Gestion des systèmes d'information Org. \Rightarrow Calcul scientifique (Mthds numériques))
R(15)=	(E-business \Rightarrow Analyse des besoins de l'entreprise)

Tableau 8.4 – Les 15 méta-règles concernant les 40 thèmes par rapport aux cinq programmes d'études CE, CS, IS, IT et SE

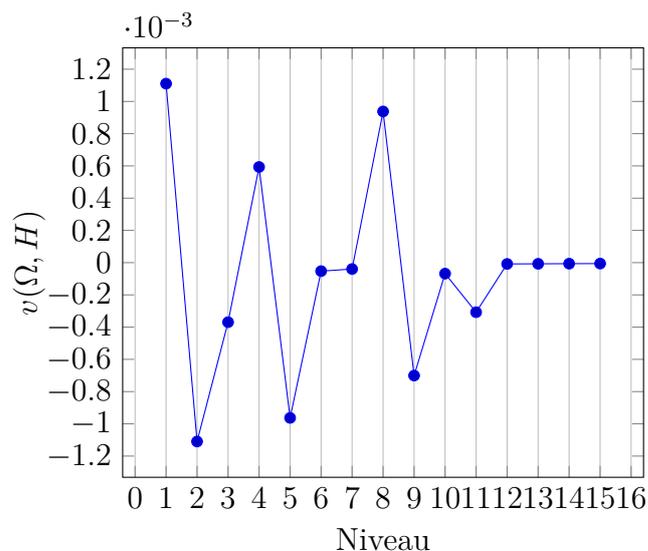


Figure 8.2 – Graphe représentant les nœuds significatifs des 15 classes

D'après la figure 8.2, les nœuds significatifs sont les extremums de la courbe, i.e. les nœuds au niveau $H = \{1, 2, 4, 5, 8, 9, 10, 11, 12, 15\}$, représentés en croix dans la figure 8.3, les cohésions inter-classes $coh_{supp(n)_{MGK}}(\underline{C})$ à l'axe des ordonnées et les implications inter-classe $\psi(\underline{C}, \underline{C}')$ au dessus des rectangles. 66.66% des nœuds sont significatifs.

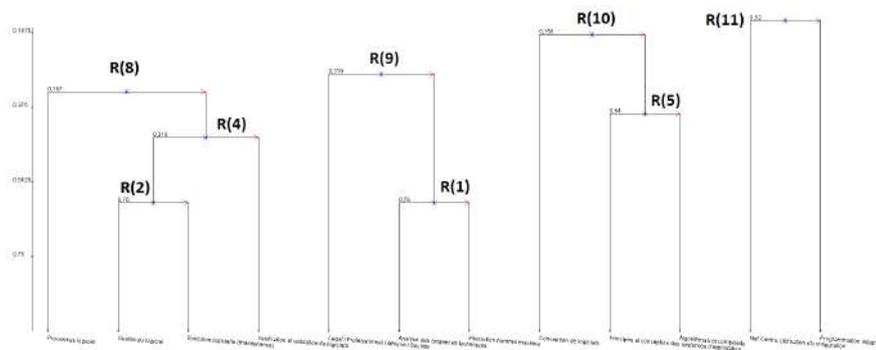


Figure 8.3 – Extrait du dendrogramme représentant des 15 classes

8.3.2.2 Quelques interprétations

- **R(1) = (Analyse des exigences techniques \Rightarrow Interaction homme machine)**: Le thème : « *Interaction homme machine* » est le prérequis du thème « *Analyse des exigences techniques* ».

Poids	CEmin1	CEmax3	CSmin1	CSmax1	ISmin1	ISmax2	ITmin1	ITmax2	SEmin2	SEmax4
Φ_1	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)

- **R(2)= (Qualité du logiciel \Rightarrow Evolution logicielle (maintenance)):**
Ceux qui maîtrisent le thème « *Qualité du logiciel* » doivent maîtriser le thème « *Evolution logicielle (maintenance)* »

Poids	CEmin2	CEmax5	CSmin2	CSmax4	ISmin2	ISmax5	ITmin4	ITmax5	SEmin3	SEmax5
Φ_2	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)

- **R(4)= ((Qualité du logiciel \Rightarrow Evolution logicielle (maintenance)) \Rightarrow Vérification et validation de logiciels):** le prérequis du **R(3)** est le thème « *Vérification et validation de logiciels* ».

Poids	CEmin1	CEmax3	CSmin1	CSmax1	CSmax2	ISmin1	ISmax2	ITmin1	ITmax2
Φ_4	(1,1,1)	(1,1,1)	(1,1,1)	(1)	(1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)

Poids	SEmin2	SEmin4	SEmax4	SEmax5
Φ_4	(1)	(1,1)	(1)	(1,1)

- **R(5)= (Principes et conception des systèmes d'exploitation \Rightarrow Algorithmes et complexité):** Le thème « *Algorithmes et complexité* » est l'étape primordial avant d'aborder le thème « *Principes et conception des systèmes d'exploitation* ».

Poids	CEmin2	CEmax4	CSmin4	CSmax5	ISmin1	ISmax2	ITmin1	ITmax2	SEmin3	SEmax4
Φ_5	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)

- **R(8)= (Processus logiciel \Rightarrow ((Qualité du logiciel \Rightarrow Evolution logicielle (maintenance)) \Rightarrow Vérification et validation de logiciels)):** la base du thème « *Processus logiciel* » est la règle **R(4)**.

Poids	CEmin1	CEmax1	CEmax3	CSmin1	CSmax1	CSmax2	ISmin1	ISmax2
Φ_8	(1,1,1,1,1,1)	(1)	(1,1,1,1,1)	(1,1,1,1,1,1)	(1,1)	(1,1,1,1)	(1,1,1,1,1,1)	(1,1,1,1,1,1)
Poids	ITmin1	ITmax1	ITmax2	SEmin2	SEmin4	SEmax4	SEmax5	
Φ_8	(1,1,1,1,1,1)	(1)	(1,1,1,1,1)	(1,1,1)	(1,1,1)	(1,1)	(1,1,1,1)	

- **R(9)= (Légal / Professionnel / Éthique / Société \Rightarrow (Analyse des exigences techniques \Rightarrow Interaction homme machine)):** le prérequis du thème « *Légal / Professionnel / Éthique / Société* » est le respect de la règle **R(1)**.

Poids	CEmin2	CEmax5	CSmin2	CSmax4	ISmin2	ISmax5	
Φ_9	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	
Poids	ITmin2	ITmin4	ITmax4	ITmax5	SEmin2	SEmin3	SEmax5
Φ_9	(1)	(1,1)	(1)	(1,1)	(1)	(1,1)	(1,1,1)

- **R(10)**= (**Conception de logiciels** \Rightarrow (**Principes et conception des systèmes d'exploitation** \Rightarrow **Algorithmes et complexité**)): le thème « *Conception de logiciels* » dépend de la règle **R(5)**.

Poids	CEmin2	CEmax4	CEmax5	CSmin3	CSmin4	CSmax5	ISmin1	ISmax1	ISmax2
Φ_{10}	(1,1,1)	(1,1)	(1)	(1)	(1,1)	(1,1,1)	(1,1,1)	(1)	(1,1)
Poids	ITmin1	ITmax2	SEmin3	SEmax4					
Φ_{10}	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)					

- **R(11)**= (**Net Centric Utilisation et configuration** \Rightarrow **Programmation intégrative**): le prérequis du thème « *Net Centric Utilisation et configuration* » est le thème « *Programmation intégrative* ».

Poids	CEmin0	CEmax2	CSmin1	CSmax3	ISmin2	ISmax4	ITmin3	ITmax5	SEmin1	SEmax3
Φ_{11}	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)

- **R(12)**= (**Systèmes intelligents (AI)** \Rightarrow **Graphisme et visualisation**): Le thème « *Graphisme et visualisation* » a une relation avec le thème « *Systèmes intelligents (AI)* ».

Poids	CEmin1	CEmax3	CSmin1	CSmax5	ISmin1	ISmax1	ITmin0	ITmax1	SEmin1	SEmax3
Φ_{12}	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)

- **R(15)**= (**E-business** \Rightarrow **Analyse des besoins de l'entreprise**): Pour faire le « *E-business* », on a besoin de savoir généralement sur le thème « *Analyse des besoins de l'entreprise* ».

Poids	CEmin0	CEmax1	CSmin0	CSmax1	ISmin5	ISmax5	ITmin1	ITmax2	SEmin1	SEmax3
Φ_{15}	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)

8.3.3 Analyse des thèmes en CS

8.3.3.1 Liens implicatifs entre les thèmes informatiques dans les types de programmes d'études CS

En partant du tableau 8.1, nous pouvons construire un tableau 8.5 binaire des relations entre les poids minimum et maximum de chaque thème dans les cinq types de programmes d'études : CE, CS, IS, IT et SE. Nous obtenons 25 méta-règles avec un risque de se tromper à $\alpha = 10\%$ présentées dans le tableau 8.7 ci-dessous pour le programme d'études CS.

	Principes de programmation	Programmation ... intégrative	Intégration de systèmes	Développement de médias numériques	Support technique
CSmin0	0	0	...	0	1
CSmin1	0	1	...	1	0
CSmin2	0	0	...	0	0
CSmin3	0	0	...	0	0
CSmin4	1	0	...	0	0
CSmin5	0	0	...	0	0
CSmax0	0	0	...	0	0
CSmax1	0	0	...	0	1
CSmax2	0	0	...	1	0
CSmax3	0	1	...	0	0
CSmax4	0	0	...	0	0
CSmax5	1	0	...	0	0

Tableau 8.5 – Tableau binaire extrait du tableau 8.1 pour les disciplines CS

Nous avons 780 règles d'association valides selon M_{GK} , avec un seuil de risque de 10% fixé par nous-même, dont parmi elles, 273 sont des règles positives et 507 sont des règles négatives, avec $0.0399 \leq \text{supp}_{(n)M_{GK}}^f \leq 1$ qui forment 51 Paires Orientées PO. Le tableau 8.6 nous les valeurs de $\text{coh}_{\text{supp}_{(n)M_{GK}}}(\underline{C})$, $\psi(\underline{C}, \underline{C}')$, $s(\Omega, H)$ et $v(\Omega, H)$ de H . Le tableau 8.7, montre les 25 méta-règles à $\alpha = 10\%$ de risque en CS.

Niveau	Cohesion inter-classe $\text{coh}_{\text{supp}_{(n)M_{GK}}}(\underline{C})$	Implication inter-classe $\psi(\underline{C}, \underline{C}')$	Niveau significatif $s(\Omega, H)$	Noeud significatif $v(\Omega, H)$
1	1	1	0.001110289	0.001110289
2	1	1	7.121808160	-0.00110957
3	1	1	0.000556928	0.000556215
4	1	1	-0.00066446	-0.00122139
5	1	1	-0.00015534	0.000509115
6	1	1	0.000227499	0.000382846
7	1	1	0.000484248	0.000256749
8	1	1	0.000654930	0.000170682
9	1	1	-0.00099725	-0.00165218
10	1	1	-0.00080303	0.000194220
11	1	1	-0.00101168	-0.00020864
12	1	1	-0.00083926	0.000172420
13	1	1	-0.00102283	-0.00018357
14	1	1	-0.00102596	-3.13141337
15	1	1	-0.00102892	-2.95937203
16	1	1	-0.00103163	-2.70526155
17	1	1	-0.00089072	0.000140906
18	1	1	-0.00071013	0.000180589
19	1	1	-0.00104473	-0.00033460
20	1	1	-0.00104651	-1.77993957
21	1	1	-0.00093023	0.000116281
22	1	1	-0.00077678	0.000153449
23	1	1	-0.00060355	0.000173229
24	1	1	-0.00042531	0.000178244
25	1	1	-0.00106460	-0.00063929

Tableau 8.6 – Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau pour CS.

Regle	
R(1)=	(Algorithmes_et_complexité \Rightarrow Principes_de_programmation)
R(2)=	(Systèmes_distribués \Rightarrow Programmation_intégrative)

R(3)= (Sécurité :_implémentation_et_mgt \Rightarrow (Systèmes_distribués \Rightarrow Programmation_intégrative))

R(4)= (Configuration_et_utilisation_des_systèmes_d'exploitation \Rightarrow Architecture_informatique_et_organisation)

R(5)= (Net_Centric_Principes_et_Design \Rightarrow (Configuration_et_utilisation_des_systèmes_d'exploitation \Rightarrow Architecture_informatique_et_organisation))

R(6)= (Interaction_homme_machine \Rightarrow (Net_Centric_Principes_et_Design \Rightarrow (Configuration_et_utilisation_des_systèmes_d'exploitation \Rightarrow Architecture_informatique_et_organisation))))

R(7)= (Légal_/_Professionnel_/_Éthique_/_Société \Rightarrow (Interaction_homme_machine \Rightarrow (Net_Centric_Principes_et_Design \Rightarrow (Configuration_et_utilisation_des_systèmes_d'exploitation \Rightarrow Architecture_informatique_et_organisation))))))

R(8)= (Analyse_des_exigences_techniques \Rightarrow (Légal_/_Professionnel_/_Éthique_/_Société \Rightarrow (Interaction_homme_machine \Rightarrow (Net_Centric_Principes_et_Design \Rightarrow (Configuration_et_utilisation_des_systèmes_d'exploitation \Rightarrow Architecture_informatique_et_organisation))))))

R(9)= (Théorie_des_langages_de_programmation \Rightarrow Principes_et_conception_des_systèmes_d'exploitation)

R(10)= (Conception_de_logiciels \Rightarrow (Théorie_des_langages_de_programmation \Rightarrow Principes_et_conception_des_systèmes_d'exploitation))

R(11)= (Modélisation_et_analyse_de_logiciels \Rightarrow Net_Centric_Utilisation_et_configuration)

R(12)= (Logique_numérique \Rightarrow (Modélisation_et_analyse_de_logiciels \Rightarrow Net_Centric_Utilisation_et_configuration))

R(13)= (Développement_de_systèmes_d'information \Rightarrow Technologies_de_plateforme)

R(14)= (Théorie_de_la_gestion_de_l'information_(DB) \Rightarrow Systèmes_intelligents_(AI))

R(15)=	(Sécurité :_problèmes_et_principes ⇒ Pratique_de_la_gestion_de_l'information_(DB))
R(16)=	(Economie_d'ingénierie_pour_SW ⇒ Analyse_des_besoins_de_l'entreprise)
R(17)=	(Développement_de_médias_numériques ⇒ (Economie_d'ingénierie_pour_SW ⇒ Analyse_des_besoins_de_l'entreprise))
R(18)=	(Support_technique ⇒ (Développement_de_médias_numériques ⇒ (Economie_d'ingénierie_pour_SW ⇒ Analyse_des_besoins_de_l'entreprise)))
R(19)=	(Gestion_des_systèmes_d'information_Org. ⇒ E-business)
R(20)=	(Vérification_et_validation_de_logiciels ⇒ Fondations_d'ingénierie_pour_SW)
R(21)=	(Processus_logiciel ⇒ (Vérification_et_validation_de_logiciels ⇒ Fondations_d'ingénierie_pour_SW))
R(22)=	(Qualité_du_logiciel ⇒ (Processus_logiciel ⇒ (Vérification_et_validation_de_logiciels ⇒ Fondations_d'ingénierie_pour_SW)))
R(23)=	(Ingénierie_des_systèmes_comp ⇒ (Qualité_du_logiciel ⇒ (Proces- sus_logiciel ⇒ (Vérification_et_validation_de_logiciels ⇒ Fonda- tions_d'ingénierie_pour_SW))))
R(24)=	(Intégration_de_systèmes ⇒ (Ingénierie_des_systèmes_comp ⇒ (Qualité_du_logiciel ⇒ (Processus_logiciel ⇒ (Vérifica- tion_et_validation_de_logiciels ⇒ Fondations_d'ingénierie_pour_SW))))))
R(25)=	(Administration_des_systèmes ⇒ Evolution_logicielle_(maintenance))

Tableau 8.7 – Les 25 méta-règles concernant les 40 thèmes par rapport au programme d'études CS

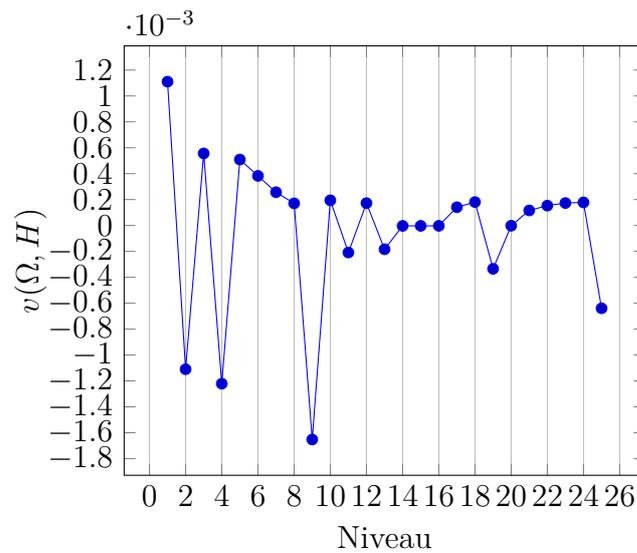


Figure 8.4 – Graphe représentant les nœuds significatifs des 25 classes

D'après la figure 8.4, les nœuds significatifs sont les extremums de la courbe, i.e. les nœuds au niveau $H = \{1, 2, 3, 4, 5, 9, 10, 11, 12, 13, 14, 16, 18, 19, 24, 25\}$, représentés en croix dans la figure 8.5. 64% des nœuds sont significatifs.

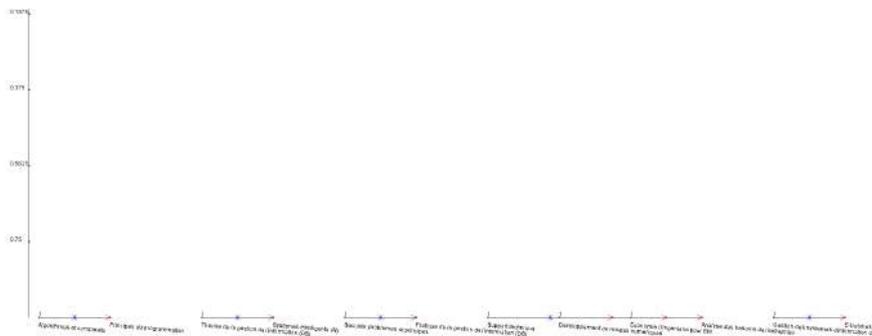


Figure 8.5 – Extrait du dendrogramme représentant des 25 classes

8.3.3.2 Quelques interprétations

— **R(1) = (Algorithmes_et_complexité \Rightarrow Principes_de_programmation):**

Le thème « *Principes de programmation* » est le pré-requis du thème « *Algorithmes et complexité* ». CSmin=4 avec $\Phi_1 = (1)$, CSmax=5 avec $\Phi_1 = (1)$,

— **R(2) = (Systèmes_distribués \Rightarrow Programmation_intégrative):** Avant d'entâmer le thème « *Systèmes distribués* », il est obligatoirement de passer

le thème « *Programmation intégrative* ». CSmin=1 avec $\Phi_2 = (1)$, CSmax=3 avec $\Phi_2 = (1)$,

- **R(3)**= (**Sécurité :_implémentation_et_mgt** \Rightarrow (**Systèmes_distribués** \Rightarrow **Programmation_intégrative**)): Pour pouvoir suivre le thème « *Sécurité :implémentation et mgt* », il fallait respecter la règle **R(2)**. CSmin=1 avec $\Phi_3 = (1, 1, 1)$, CSmax=3 avec $\Phi_3 = (1, 1, 1)$,
- **R(4)**= (**Configuration_et_utilisation_des_systèmes_d'exploitation** \Rightarrow **Architecture_informatique_et_organisation**): il faut étudier d'abord le thème « *Architecture informatique et organisation* » pour pouvoir poursuivre l'étude sur le thème « *Configuration et utilisation des systèmes d'exploitation* ». CSmin=2 avec $\Phi_4 = (1)$, CSmax=4 avec $\Phi_4 = (1)$,
- **R(5)**= (**Net_Centric_Principes_et_Design** \Rightarrow (**Configuration_et_utilisation_des_systèmes_d'exploitation** \Rightarrow **Architecture_informatique_et_organisation**)): Le respect de la règle **R(4)** nous mène à assimiler le thème « *Net Centric Principes et Design* ». CSmin=2 avec $\Phi_5 = (1, 1, 1)$, CSmax=4 avec $\Phi_5 = (1, 1, 1)$,
- **R(10)**= (**Conception_de_logiciels** \Rightarrow **R(9)**): **R(9)** est le prérequis du thème « *Conception de logiciels* ». CSmin=3 avec $\Phi_{10}=(1,1,1)$, CSmax=5 avec $\Phi_{10}=(1,1,1)$,
- **R(11)**= (**Modélisation_et_analyse_de_logiciels** \Rightarrow **Net_Centric_Utilisation_et_configuration**): Si on étudie le thème « *Modélisation et analyse de logiciels* », on étudie généralement le thème « *Net Centric Utilisation et configuration* ». CSmin=2 avec $\Phi_{11}=(1)$, CSmax=3 avec $\Phi_{11}=(1)$,
- **R(12)**= (**Logique_numérique** \Rightarrow **R(11)**): Respecter d'abord la règle **R(11)** avant d'entâmer le thème « *Logique numérique* ». CSmin=2 avec $\Phi_{12}=(1,1,1)$, CSmax=3 avec $\Phi_{12}=(1,1,1)$,

- **R(13)= (Développement_de_systèmes_d'information ⇒ Technologies_de_plateforme)**: Le thème « *Technologies de plateforme* » assure généralement l'étude du thème « *Développement de systèmes d'information* ». CSmin=0 avec $\Phi_{13}=(1)$, CSmax=2 avec $\Phi_{13}=(1)$,

- **R(14)= (Théorie_de_la_gestion_de_l'information_(DB) ⇒ Systèmes_intelligents_(AI))**: Si on étudie le thème « *Théorie de la gestion de l'information (DB)* », on apprend généralement le thème « *Systèmes intelligents (AI)* ». CSmin=2 avec $\Phi_{14}=(1)$, CSmax=5 avec $\Phi_{14}=(1)$,

- **R(16)= (Economie_d'ingénierie_pour_SW ⇒ Analyse_des_besoins_de_l'entreprise)**: le thème « *Economie d'ingénierie pour SW* » doit être généralement précédé du thème « *Analyse des besoins de l'entreprise* ». CSmin=0 avec $\Phi_{16}=(1)$, CSmax=1 avec $\Phi_{16}=(1)$,

- **R(18)= (Support_technique ⇒ (Développement_de_médias_numériques ⇒ R(16)))**: « *Développement de médias numériques* » est la base du thème « *Support technique* ». CSmin=0 avec $\Phi_{18}=(1,1,1,1,1,1)$, CSmax=1 avec $\Phi_{18}=(1,1,1,1,1,1)$,

- **R(19)= (Gestion_des_systèmes_d'information_Org. ⇒ E-business)**: Si on étudie le thème « *Gestion des systèmes d'information Org.* », on doit connaître généralement sur le thème « *E-business* ». CSmin=0 avec $\Phi_{19}=(1)$, CSmax=0 avec $\Phi_{19}=(1)$, une remarque que le thème *Gestion des systèmes d'information Org.* est pris en charge seulement en IS et le thème E-business n'est pas abordé dans CE et CS. Ces deux thèmes ne sont pas obligatoires dans quelques programmes d'études,

- **R(24)= (Intégration_de_systèmes ⇒ R(23))**: le thème « *Intégration de systèmes* » dépend de la règle **R(23)**. CSmin=1 avec $\Phi_{24}=(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)$, CSmax=2 avec $\Phi_{24}=(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)$,

- **R(25)= (Administration_des_systèmes ⇒**

Evolution_logicielle_(maintenance)): le prérequis du thème « *Administration des systèmes* » est le thème « *Evolution logicielle (maintenance)* ». CSmin=1 avec $\Phi_{25}=(1)$, CSmax=1 avec $\Phi_{25}=(1)$

8.3.4 Analyse des capacités d'aptitudes des étudiants en CE, CS, IS, IT et SE

Le rapport (Shackelford et al., 2005) définit également 60 **capacités d'aptitudes** parmi les 11 **catégories** suivantes (CF. Tableau 8.8) :

- Algorithmes (*Algorithms*)
- Programmes d'application (*Application programs*)
- Programmation informatique (*Computer programming*)
- Matériel et appareils (*Hardware and devices*)
- Interface Homme-Machine (*Human-Computer Interface*)
- Systèmes d'information (*Information systems*)
- Gestion de l'information (*Information management*)
- Planification des ressources informatiques (*Information management*)
- Systèmes intelligents (*Intelligent systems*)
- Réseaux et communications (*Networking and communications*)
- Développement de systèmes via l'intégration (*Systems Development Through Integration*)

et évalue leur poids pour chaque discipline informatique. L'analyse de ce rapport a permis d'identifier les deux composantes essentielles et communes aux cinq disciplines informatiques : « **acquérir les fondamentaux de la programmation** » et être capable de « **réaliser des programmes simples** ». Les premières formations aux Etats-Unis sont : le Computer Science (CS) pour les étudiants qui voulaient suivre le cursus d'experts en software, l'Electrical Engineering (EE) en hardware et l'Information Systems (IS), ceux qui souhaitaient utiliser le matériel et le logiciel pour résoudre des problèmes liés aux entreprises. En Europe fait référence au CS. C'est au cours des années 90 sont apparues aux États-Unis les nouvelles disciplines comme le Computer Engineering (CE) le Software Engineering (SE) et l'Information Technology (IT). Cette décomposition n'ayant pas été réalisée dans le modèle

européen, l'informatique par exemple, en France regroupe bien entendu le CS mais aussi tout ou partie du CE, du SE, de l'IT et de L'IS. Le CS est donc la discipline du modèle anglo-saxon la plus proche historiquement de l'informatique et mérite donc d'être présentée en détail.

La discipline CS est présentée dans l'ACM/-IEEE ([Chang et al., 2001](#)). Ce rapport montre l'architecture des enseignements, le détail des différentes approches envisageables et le positionnement de l'apprentissage de la programmation. Nous allons maintenant analyser les capacités d'aptitudes des étudiants ayant suivis les parcours informatiques dans les cinq disciplines existants à partir du tableau 8.8 dans le modèle anglosaxon, et pouvons construire un tableau 8.9 binaire des relations entre les poids minimum et maximum de chaque capacité d'aptitude des étudiants ayant poursuivi des études en informatique dans les cinq types de programmes d'études : CE, CS, IS, IT et SE. Nous obtenons 24 méta-règles avec un risque de se tromper à $\alpha = 1\%$ présentées dans le tableau 8.8.

Catégorie	Capacité d'aptitude	CE	CS	IS	IT	SE	
Algorithmes	Prouver des résultats théoriques	3	5	1	0	3	
	Développer des solutions aux problèmes de programmation	3	5	1	1	3	
	Développer des programmes de preuve de concept	3	5	3	1	3	
Programmes d'application	Déterminer si des solutions plus rapides sont possibles	3	5	1	1	3	
	Concevoir un programme de traitement de texte	3	4	1	0	4	
	Utilisez bien les fonctionnalités de traitement de texte	3	3	5	5	3	
	Former et soutenir les utilisateurs de traitement de texte	2	2	4	5	2	
	Concevoir un tableau (par exemple : Excel)	3	4	1	0	4	
	Utilisez bien les fonctionnalités de la feuille de calcul	2	2	5	5	3	
	Former et soutenir les utilisateurs de tableurs	2	2	4	5	2	
Programmation informatique	Faire de la programmation à petite échelle	5	5	3	3	5	
	Faire de la programmation à grande échelle	3	4	2	2	5	
	Faire la programmation des systèmes	4	4	1	1	4	
	Développer de nouveaux systèmes logiciels	3	4	3	1	5	
	Créer des systèmes critiques pour la sécurité	4	3	0	0	5	
Matériel et appareils	Gérer des projets critiques pour la sécurité	3	2	0	0	5	
	Concevoir des systèmes embarqués	5	1	0	0	1	
	Mettre en œuvre des systèmes intégrés	5	2	1	1	3	
	Concevoir des périphériques d'ordinateur	5	1	0	0	1	
	Concevoir des systèmes de capteurs complexes	5	1	0	0	1	
	Concevoir une puce	5	1	0	0	1	
	Programmer une puce	5	1	0	0	1	
	Concevoir un ordinateur	5	1	0	0	1	
	Interface homme-machine	Créer une interface utilisateur de logiciel	3	4	4	5	4
		Produire des graphiques ou des logiciels de jeux	2	5	0	0	5
Concevoir un appareil convivial pour l'homme		4	2	0	1	3	
Systèmes d'information	Définir les exigences du système d'information	2	2	5	3	4	
	Concevoir des systèmes d'information	2	3	5	3	3	
	Mettre en place des systèmes d'information	3	3	4	3	5	
	Former les utilisateurs à utiliser les systèmes d'information	1	1	4	5	1	
	Maintenir et modifier les systèmes d'information	3	3	5	4	3	
Gestion de l'information (base de données)	Concevoir un système de gestion de base de données (par exemple, Oracle)	2	5	1	0	4	
	Modéliser et concevoir une base de données	2	2	5	5	2	
	Implémenter un logiciel de recherche d'information	1	5	3	3	4	
	Sélectionner les produits de base	1	3	5	5	3	
	Configurer les produits de base	1	2	5	5	2	
	Gérer les bases	1	2	5	5	2	
	Former et soutenir les utilisateurs de bases de données	2	2	5	5	2	
Planification des ressources informatiques	Développer un plan d'information d'entreprise	0	0	5	3	0	
	Développer un plan de ressources informatiques	2	2	5	5	2	
	Mettre à niveau des ressources de calendrier / budget	2	2	5	5	2	
	Installer / mettre à niveau les ordinateurs	4	3	3	5	3	
	Installer / mettre à niveau un logiciel informatique	3	3	3	5	3	
Systèmes intelligents	Concevoir des systèmes de raisonnement automatique	2	4	0	0	2	
	Mettre en place des systèmes intelligents	2	4	0	0	4	
Réseaux et communications	Configuration du réseau de conception	3	3	3	4	2	
	Sélectionnez les composants réseau	2	2	4	5	2	
	Installer le réseau informatique	2	1	3	5	2	
	Gérer les réseaux informatiques	3	3	3	5	3	
	Implémenter un logiciel de communication	5	4	1	1	4	
	Gérer les ressources de communication	1	0	3	5	0	
	Mettre en œuvre un système informatique mobile	5	3	0	1	3	
Développement de systèmes via l'intégration	Gérer les ressources informatiques mobiles	3	2	2	4	2	
	Gérer la présence sur le Web d'une organisation	2	2	4	5	2	
	Configurer et intégrer les logiciels de commerce électronique	2	3	4	5	4	
	Développer des solutions multimédia	2	3	4	5	3	
	Configurer et intégrer des systèmes d'apprentissage en ligne	1	2	5	5	3	
	Développer des solutions d'affaires	1	2	5	3	2	
	Évaluer de nouvelles formes de moteur de recherche	2	4	4	4	4	

Tableau 8.8 – Capacités d'aptitude des diplômés en informatique par discipline, tiré du (Shackelford et al., 2005, p.28)

	Prouver des résultats théoriques	Developper des solutions aux problemes de programmation	...	Configurer et integrer des systemes d'apprentissage en ligne	Developper des solutions d'affaires	Evaluer de nouvelles formes de moteur de recherche
CE0	0	0	...	0	0	0
...
CE5	0	0	...	0	0	0
CS0	0	0	...	0	0	0
...
CS5	1	1	...	0	0	0
IS0	0	0	...	0	0	0
...
IS5	0	0	...	1	1	0
IT0	1	0	...	0	0	0
...
IT5	0	0	...	1	0	0
SE0	0	0	...	0	0	0
...
SE5	0	0	...	0	0	0

Tableau 8.9 – Tableau binaire extrait du tableau 8.8 pour les capacités d'aptitudes des étudiants poursuivant les parcours CE,CS, IS, IT et SE

8.3.4.1 Liens implicatifs des capacités d'aptitudes dans CE, CS, IS, IT et SE

Nous avons 1711 règles d'association valides selon M_{GK} , avec un seuil de risque de 1% fixé par nous-même, dont parmi elles, 1009 sont des règles positives et 702 sont des règles négatives, avec $0.04 \leq \text{supp}_{(n)M_{GK}}^f \leq 1$ qui forment 78 Paires Orientées PO, i.e. $\text{card}(\text{coh}_{\text{supp}_{(n)M_{GK}}}) = 78$ avec $0.607 \leq \text{coh}_{\text{supp}_{(n)M_{GK}}} \leq 1$. Le tableau 8.10 nous montre les valeurs des cohésions inter-classes $\text{coh}_{\text{supp}_{(n)M_{GK}}}(\underline{C})$, des implications inter-classes $\psi(\underline{C}, \underline{C}')$, des niveaux $s(\Omega, H)$ et nœuds significatifs $v(\Omega, H)$ de chaque niveau de la hiérarchie H . Nous présentons aussi dans le tableau 8.11, les 24 méta-règles avec un risque de se tromper à $\alpha = 1\%$, pour les capacités d'aptitudes des étudiants poursuivant les parcours en CE, CS, IS, IT et SE.

Niveau	Cohesion inter-classe $coh_{supp(n)M_{GK}}(\mathcal{C})$	Implication inter-classe $\psi(\mathcal{C}, \mathcal{C}')$	Niveau significatif $s(\Omega, H)$	Noeud significatif $v(\Omega, H)$
1	1	1	0,00050615	0,00050615
2	1	1	0,00000015	-0,00050600
3	1	1	-0,00016842	-0,00016857
4	1	1	0,00010176	0,00027018
5	1	1	0,00025400	0,00015224
6	1	1	-0,00039328	-0,00064728
7	1	1	-0,00022926	0,00016402
8	1	1	-0,00007095	0,00015831
9	1	1	0,00005829	0,00012924
10	1	1	0,00015692	0,00009863
11	1	1	-0,00046355	-0,00062047
12	1	1	-0,00038841	0,00007514
13	1	1	-0,00029504	0,00009336
14	1	1	-0,00047198	-0,00017694
15	1	1	-0,00047306	-0,00000108
16	0,752	0,012	0,00010270	0,00057577
17	0,717	0,578	-0,00044318	-0,00054588
18	0,717	0,578	-0,00044561	-0,00000243
19	0,717	0,578	-0,00044785	-0,00000224
20	0,607	0,760	-0,00048697	-0,00003912
21	0,607	0,760	-0,00048731	-0,00000034
22	0,607	0,760	-0,00048764	-0,00000033
23	0,607	0,760	-0,00048795	-0,00000031
24	0,607	0,760	-0,00048826	-0,00000031

Tableau 8.10 – Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau pour les capacités d'aptitudes.

Regle	
R(1)=	(Déterminer si des solutions plus rapides sont possibles \Rightarrow Développer des solutions aux problèmes de programmation)
R(2)=	(Concevoir un tableur (par exemple : Excel) \Rightarrow Concevoir un programme de traitement de texte)
R(3)=	(Former et soutenir les utilisateurs de tableurs \Rightarrow Former et soutenir les utilisateurs de traitement de texte)
R(4)=	(Sélectionner les composants réseau \Rightarrow (Former et soutenir les utilisateurs de tableurs \Rightarrow Former et soutenir les utilisateurs de traitement de texte))
R(5)=	(Gérer la présence sur le Web d'une organisation \Rightarrow (Sélectionner les composants réseau \Rightarrow (Former et soutenir les utilisateurs de tableurs \Rightarrow Former et soutenir les utilisateurs de traitement de texte)))
R(6)=	(Concevoir des périphériques d'ordinateur \Rightarrow Concevoir des systèmes embarqués)
R(7)=	(Concevoir des systèmes de capteurs complexes \Rightarrow (Concevoir des périphériques d'ordinateur \Rightarrow Concevoir des systèmes embarqués))

R(8)= (Concevoir une puce \Rightarrow (Concevoir des systèmes de capteurs complexes \Rightarrow (Concevoir des périphériques d'ordinateur \Rightarrow Concevoir des systèmes embarqués)))

R(9)= (Programmer une puce \Rightarrow (Concevoir une puce \Rightarrow (Concevoir des systèmes de capteurs complexes \Rightarrow (Concevoir des périphériques d'ordinateur \Rightarrow Concevoir des systèmes embarqués))))

R(10)= (Concevoir un ordinateur \Rightarrow (Programmer une puce \Rightarrow (Concevoir une puce \Rightarrow (Concevoir des systèmes de capteurs complexes \Rightarrow (Concevoir des périphériques d'ordinateur \Rightarrow Concevoir des systèmes embarqués))))))

R(11)= (Former et soutenir les utilisateurs de bases de données \Rightarrow Modéliser et concevoir une base de données)

R(12)= (Développer un plan de ressources informatiques \Rightarrow (Former et soutenir les utilisateurs de bases de données \Rightarrow Modéliser et concevoir une base de données))

R(13)= (Mettre à niveau des ressources de calendrier / budget \Rightarrow (Développer un plan de ressources informatiques \Rightarrow (Former et soutenir les utilisateurs de bases de données \Rightarrow Modéliser et concevoir une base de données)))

R(14)= (Gérer les bases \Rightarrow Configurer les produits de base)

R(15)= (Gérer les réseaux informatiques \Rightarrow Installer / mettre à niveau un logiciel informatique)

R(16)= ((Mettre à niveau des ressources de calendrier / budget \Rightarrow (Développer un plan de ressources informatiques \Rightarrow (Former et soutenir les utilisateurs de bases de données \Rightarrow Modéliser et concevoir une base de données))) \Rightarrow (Gérer la présence sur le Web d'une organisation \Rightarrow (Sélectionner les composants réseau \Rightarrow (Former et soutenir les utilisateurs de tableurs \Rightarrow Former et soutenir les utilisateurs de traitement de texte))))))

R(17)= ((Déterminer si des solutions plus rapides sont possibles \Rightarrow Développer des solutions aux problèmes de programmation) \Rightarrow Prouver des résultats théoriques)

R(18)=	(Développer des solutions d'affaires \Rightarrow (Gérer les bases \Rightarrow Configurer les produits de base))
R(19)=	((Gérer les réseaux informatiques \Rightarrow Installer / mettre à niveau un logiciel informatique) \Rightarrow Installer / mettre à niveau les ordinateurs)
R(20)=	(Maintenir et modifier les systèmes d'information \Rightarrow Utiliser bien les fonctionnalités de traitement de texte)
R(21)=	(Configurer et intégrer des systèmes d'apprentissage en ligne \Rightarrow Utiliser bien les fonctionnalités de la feuille de calcul)
R(22)=	(Implémenter un logiciel de communication \Rightarrow Faire la programmation des systèmes)
R(23)=	(Mettre en place des systèmes intelligents \Rightarrow Concevoir des systèmes de raisonnement automatique)
R(24)=	(Développer des solutions multimédia \Rightarrow Configurer et intégrer les logiciels de commerce électronique)

Tableau 8.11 – Les 24 méta-règles par rapport aux capacités d'aptitudes des diplômés en informatique

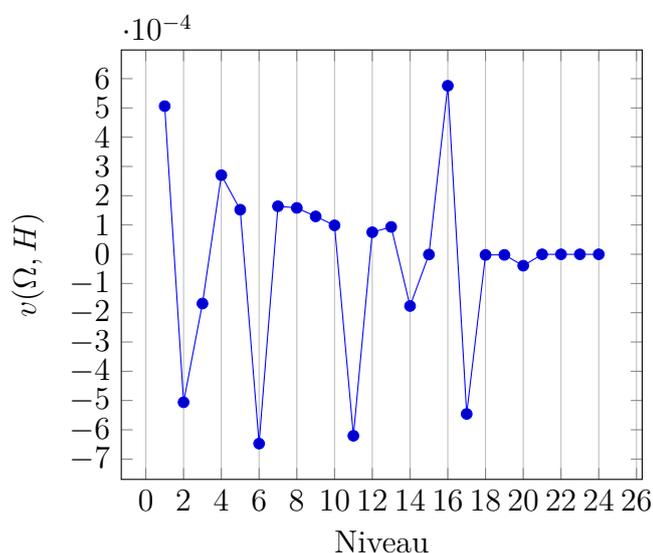


Figure 8.6 – Graphe représentant les nœuds significatifs des 24 classes

D'après la figure 8.6, les nœuds significatifs sont les extremums de la courbe, i.e. les nœuds au niveau $H = \{1, 2, 4, 6, 7, 11, 13, 14, 16, 17, 19, 20, 21\}$, représentés en croix dans la figure 8.7. 54.16% des nœuds sont significatifs.

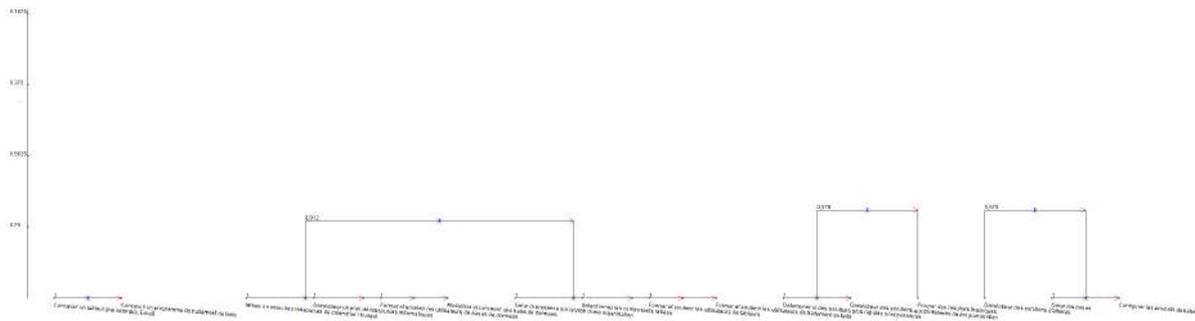


Figure 8.7 – Extrait du dendrogramme représentant des 24 classes

8.3.4.2 Quelques interprétations

- **R(1) = (Développer des solutions aux problèmes de programmation ⇒ Prouver des résultats théoriques)**: tous ceux qui sont capables de déterminer si des solutions plus rapides sont possibles, sont capables de développer des solutions aux problèmes de programmation.

Poids	CE3	CS5	IS1	IT1	SE3
Φ_1	(1)	(1)	(1)	(1)	(1)

- **R(2) = (Concevoir un tableur (par exemple : Excel) ⇒ Concevoir un programme de traitement de texte)**: les étudiants qui arrivent à concevoir un tableur paraissent généralement être capables de concevoir un programme de traitement de texte.

Poids	CE3	CS4	IS1	IT0	SE4
Φ_2	(1)	(1)	(1)	(1)	(1)

- **R(5) = (Gérer la présence sur le Web d'une organisation ⇒ (Sélectionner les composants réseau ⇒ (Former et soutenir les utilisateurs de tableurs ⇒ Former et soutenir les utilisateurs de traitement de texte)))**, car nous avons $R(3) \subset R(4) \subset R(5)$: Les étudiants capables de gérer la présence sur le Web d'une organisation arrivent généralement à sélectionner des composants réseau et pourraient former et soutenir les utilisateurs de tableurs et de traitement de texte.

Poids	CE2	CS2	IS4	IT5	SE2
Φ_5	(1,1,1,1,1,1)	(1,1,1,1,1,1)	(1,1,1,1,1,1)	(1,1,1,1,1,1)	(1,1,1,1,1,1)

- **R(10)**= (Concevoir un ordinateur \Rightarrow (Programmer une puce \Rightarrow (Concevoir une puce \Rightarrow (Concevoir des systèmes de capteurs complexes \Rightarrow (Concevoir des périphériques d'ordinateur \Rightarrow Concevoir des systèmes embarqués))))), car nous avons $R(6) \subset R(7) \subset R(8) \subset R(9) \subset R(10)$: Ceux qui sont capables de concevoir un ordinateur arrivent généralement à programmer et concevoir une puce, concevoir des systèmes de capteurs complexes, des périphériques d'ordinateurs et des systèmes embarqués.

Poids	CE5	CS1	IS0
Φ_{10}	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)
Poids	IT0	SE1	
Φ_{10}	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)	

- **R(13)**= (Mettre à niveau des ressources de calendrier / budget \Rightarrow (Développer un plan de ressources informatiques \Rightarrow (Former et soutenir les utilisateurs de bases de données \Rightarrow Modéliser et concevoir une base de données))), car nous avons $R(11) \subset R(12) \subset R(13)$: Ceux qui sont capables de modéliser et concevoir une base de données arrivent généralement à former et soutenir les utilisateurs de bases de données et mettre à niveau des ressources de calendrier/budget.

Poids	CE2	CS2	IS5	IT5	SE2
Φ_{13}	(1,1,1,1,1,1)	(1,1,1,1,1,1)	(1,1,1,1,1,1)	(1,1,1,1,1,1)	(1,1,1,1,1,1)

- **R(14)**= (Gérer les bases \Rightarrow Configurer les produits de base): Ceux qui arrivent gérer les bases sont généralement capables de configurer les produits de base.

Poids	CE1	CS2	IS5	IT5	SE2
Φ_{14}	(1)	(1)	(1)	(1)	(1)

- **R(15)**= (Gérer les réseaux informatiques \Rightarrow Installer / mettre à niveau un logiciel informatique): Ceux qui sont capables de gérer des réseaux informatiques sont généralement capables d'installer et mettre à niveau un logiciel informatique.

Poids	CE3	CS3	IS3	IT5	SE3
Φ_{15}	(1)	(1)	(1)	(1)	(1)

- **R(16)** \Leftrightarrow (**R(13)** \Rightarrow **R(5)**): ceux qui ont la compétence selon la règle **R(13)** sont généralement capables d'assurer la règle **R(5)**.

- **R(22)= (Implémenter un logiciel de communication \Rightarrow Faire la programmation des systèmes):** Ceux qui sont capables d'implémenter un logiciel de communication n'ont généralement pas de problèmes à faire la programmation des systèmes.

Poids	CE4	CS4	IS1	IT1	SE4
Φ_{22}	(1)	(1)	(1)	(1)	(1)

- **R(23)= (Mettre en place des systèmes intelligents \Rightarrow Concevoir des systèmes de raisonnement automatique):** Ceux qui sont capables de mettre en place des systèmes intelligents sont généralement capables de concevoir des systèmes de raisonnement automatique.

Poids	CE2	CS4	IS0	IT0	SE2
Φ_{23}	(1)	(1)	(1)	(1)	(1)

- **R(24)= (Développer des solutions multimédia \Rightarrow Configurer et intégrer les logiciels de commerce électronique):** Ceux qui arrivent à développer des solutions multimédia arrivent généralement à configurer et intégrer les logiciels de commerce électronique.

Poids	CE2	CS3	IS4	IT5	SE4
Φ_{23}	(1)	(1)	(1)	(1)	(1)

8.3.5 Analyse des capacités d'aptitudes des étudiants en CS

A partir du tableau 8.11, nous pouvons faire l'analyse des capacités d'aptitudes des étudiants diplômés qui suivent la branche CS.

8.3.5.1 Liens implicatifs des capacités d'aptitudes en CS

Nous obtenons 53 méta-règles concernant les capacités d'aptitudes des diplômés en CS en utilisant l'outil d'analyse CHIC- M_{GK} .

Regle	
R(1)=	(Développer des solutions aux problèmes de programmation \Rightarrow Prouver des résultats théoriques)
R(2)=	(Développer des programmes de preuve de concept \Rightarrow (Développer des solutions aux problèmes de programmation \Rightarrow Prouver des résultats théoriques))

R(3)= (Déterminer si des solutions plus rapides sont possibles \Rightarrow (Développer des programmes de preuve de concept \Rightarrow (Développer des solutions aux problèmes de programmation \Rightarrow Prouver des résultats théoriques)))

R(4)= (Faire de la programmation à petite échelle \Rightarrow (Déterminer si des solutions plus rapides sont possibles \Rightarrow (Développer des programmes de preuve de concept \Rightarrow (Développer des solutions aux problèmes de programmation \Rightarrow Prouver des résultats théoriques))))

R(5)= (Produire des graphiques ou des logiciels de jeux \Rightarrow (Faire de la programmation à petite échelle \Rightarrow (Déterminer si des solutions plus rapides sont possibles \Rightarrow (Développer des programmes de preuve de concept \Rightarrow (Développer des solutions aux problèmes de programmation \Rightarrow Prouver des résultats théoriques))))))

R(6)= (Concevoir un système de gestion de base de données (par exemple : Oracle) \Rightarrow (Produire des graphiques ou des logiciels de jeux \Rightarrow (Faire de la programmation à petite échelle \Rightarrow (Déterminer si des solutions plus rapides sont possibles \Rightarrow (Développer des programmes de preuve de concept \Rightarrow (Développer des solutions aux problèmes de programmation \Rightarrow Prouver des résultats théoriques))))))

R(7)= (Implémenter un logiciel de recherche d'information \Rightarrow (Concevoir un système de gestion de base de données (par exemple : Oracle) \Rightarrow (Produire des graphiques ou des logiciels de jeux \Rightarrow (Faire de la programmation à petite échelle \Rightarrow (Déterminer si des solutions plus rapides sont possibles \Rightarrow (Développer des programmes de preuve de concept \Rightarrow (Développer des solutions aux problèmes de programmation \Rightarrow Prouver des résultats théoriques))))))

R(8)= (Concevoir un tableur (par exemple : Excel) \Rightarrow Concevoir un programme de traitement de texte)

R(9)= (Faire de la programmation à grande échelle \Rightarrow (Concevoir un tableur (par exemple : Excel) \Rightarrow Concevoir un programme de traitement de texte))

R(10)= (Faire la programmation des systèmes \Rightarrow (Faire de la programmation à grande échelle \Rightarrow (Concevoir un tableur (par exemple : Excel) \Rightarrow Concevoir un programme de traitement de texte)))

R(11)= (Développer de nouveaux systèmes logiciels \Rightarrow (Faire la programmation des systèmes \Rightarrow (Faire de la programmation à grande échelle \Rightarrow (Concevoir un tableur (par exemple : Excel) \Rightarrow Concevoir un programme de traitement de texte))))

R(12)= (Créer une interface utilisateur de logiciel \Rightarrow (Développer de nouveaux systèmes logiciels \Rightarrow (Faire la programmation des systèmes \Rightarrow (Faire de la programmation à grande échelle \Rightarrow (Concevoir un tableur (par exemple : Excel) \Rightarrow Concevoir un programme de traitement de texte))))))

R(13)= (Concevoir des systèmes de raisonnement automatique \Rightarrow (Créer une interface utilisateur de logiciel \Rightarrow (Développer de nouveaux systèmes logiciels \Rightarrow (Faire la programmation des systèmes \Rightarrow (Faire de la programmation à grande échelle \Rightarrow (Concevoir un tableur (par exemple : Excel) \Rightarrow Concevoir un programme de traitement de texte))))))

R(14)= (Mettre en place des systèmes intelligents \Rightarrow (Concevoir des systèmes de raisonnement automatique \Rightarrow (Créer une interface utilisateur de logiciel \Rightarrow (Développer de nouveaux systèmes logiciels \Rightarrow (Faire la programmation des systèmes \Rightarrow (Faire de la programmation à grande échelle \Rightarrow (Concevoir un tableur (par exemple : Excel) \Rightarrow Concevoir un programme de traitement de texte))))))

R(15)= (Implémenter un logiciel de communication \Rightarrow (Mettre en place des systèmes intelligents \Rightarrow (Concevoir des systèmes de raisonnement automatique \Rightarrow (Créer une interface utilisateur de logiciel \Rightarrow (Développer de nouveaux systèmes logiciels \Rightarrow (Faire la programmation des systèmes \Rightarrow (Faire de la programmation à grande échelle \Rightarrow (Concevoir un tableur (par exemple : Excel) \Rightarrow Concevoir un programme de traitement de texte))))))

R(16)= (Évaluer de nouvelles formes de moteur de recherche \Rightarrow (Implémenter un logiciel de communication \Rightarrow (Mettre en place des systèmes intelligents \Rightarrow (Concevoir des systèmes de raisonnement automatique \Rightarrow (Créer une interface utilisateur de logiciel \Rightarrow (Développer de nouveaux systèmes logiciels \Rightarrow (Faire la programmation des systèmes \Rightarrow (Faire de la programmation à grande échelle \Rightarrow (Concevoir un tableur (par exemple : Excel) \Rightarrow Concevoir un programme de traitement de texte))))))))))

R(17)= (Créer des systèmes critiques pour la sécurité \Rightarrow Utiliser bien les fonctionnalités de traitement de texte)

R(18)= (Concevoir des systèmes d'information \Rightarrow (Créer des systèmes critiques pour la sécurité \Rightarrow Utiliser bien les fonctionnalités de traitement de texte))

R(19)= (Mettre en place des systèmes d'information \Rightarrow (Concevoir des systèmes d'information \Rightarrow (Créer des systèmes critiques pour la sécurité \Rightarrow Utiliser bien les fonctionnalités de traitement de texte)))

R(20)= (Maintenir et modifier les systèmes d'information \Rightarrow (Mettre en place des systèmes d'information \Rightarrow (Concevoir des systèmes d'information \Rightarrow (Créer des systèmes critiques pour la sécurité \Rightarrow Utiliser bien les fonctionnalités de traitement de texte))))

R(21)= (Sélectionner les produits de base \Rightarrow (Maintenir et modifier les systèmes d'information \Rightarrow (Mettre en place des systèmes d'information \Rightarrow (Concevoir des systèmes d'information \Rightarrow (Créer des systèmes critiques pour la sécurité \Rightarrow Utiliser bien les fonctionnalités de traitement de texte))))))

R(22)= (Installer / mettre à niveau les ordinateurs \Rightarrow (Sélectionner les produits de base \Rightarrow (Maintenir et modifier les systèmes d'information \Rightarrow (Mettre en place des systèmes d'information \Rightarrow (Concevoir des systèmes d'information \Rightarrow (Créer des systèmes critiques pour la sécurité \Rightarrow Utiliser bien les fonctionnalités de traitement de texte))))))

R(23)= (Installer / mettre à niveau un logiciel informatique \Rightarrow (Installer / mettre à niveau les ordinateurs \Rightarrow (Sélectionner les produits de base \Rightarrow (Maintenir et modifier les systèmes d'information \Rightarrow (Mettre en place des systèmes d'information \Rightarrow (Concevoir des systèmes d'information \Rightarrow (Créer des systèmes critiques pour la sécurité \Rightarrow Utiliser bien les fonctionnalités de traitement de texte))))))

R(24)= (Configuration du réseau de conception \Rightarrow (Installer / mettre à niveau un logiciel informatique \Rightarrow (Installer / mettre à niveau les ordinateurs \Rightarrow (Sélectionner les produits de base \Rightarrow (Maintenir et modifier les systèmes d'information \Rightarrow (Mettre en place des systèmes d'information \Rightarrow (Concevoir des systèmes d'information \Rightarrow (Créer des systèmes critiques pour la sécurité \Rightarrow Utiliser bien les fonctionnalités de traitement de texte))))))

R(25)= (Gérer les réseaux informatiques \Rightarrow (Configuration du réseau de conception \Rightarrow (Installer / mettre à niveau un logiciel informatique \Rightarrow (Installer / mettre à niveau les ordinateurs \Rightarrow (Sélectionner les produits de base \Rightarrow (Maintenir et modifier les systèmes d'information \Rightarrow (Mettre en place des systèmes d'information \Rightarrow (Concevoir des systèmes d'information \Rightarrow (Créer des systèmes critiques pour la sécurité \Rightarrow Utiliser bien les fonctionnalités de traitement de texte))))))

R(26)= (Mettre en œuvre un système informatique mobile \Rightarrow (Gérer les réseaux informatiques \Rightarrow (Configuration du réseau de conception \Rightarrow (Installer / mettre à niveau un logiciel informatique \Rightarrow (Installer / mettre à niveau les ordinateurs \Rightarrow (Sélectionner les produits de base \Rightarrow (Maintenir et modifier les systèmes d'information \Rightarrow (Mettre en place des systèmes d'information \Rightarrow (Concevoir des systèmes d'information \Rightarrow (Créer des systèmes critiques pour la sécurité \Rightarrow Utiliser bien les fonctionnalités de traitement de texte))))))

R(27)= (Configurer et intégrer les logiciels de commerce électronique ⇒ (Mettre en œuvre un système informatique mobile ⇒ (Gérer les réseaux informatiques ⇒ (Configuration du réseau de conception ⇒ (Installer / mettre à niveau un logiciel informatique ⇒ (Installer / mettre à niveau les ordinateurs ⇒ (Sélectionner les produits de base ⇒ (Maintenir et modifier les systèmes d'information ⇒ (Mettre en place des systèmes d'information ⇒ (Concevoir des systèmes d'information ⇒ (Créer des systèmes critiques pour la sécurité ⇒ Utiliser bien les fonctionnalités de traitement de texte))))))))))

R(28)= (Développer des solutions multimédia ⇒ (Configurer et intégrer les logiciels de commerce électronique ⇒ (Mettre en œuvre un système informatique mobile ⇒ (Gérer les réseaux informatiques ⇒ (Configuration du réseau de conception ⇒ (Installer / mettre à niveau un logiciel informatique ⇒ (Installer / mettre à niveau les ordinateurs ⇒ (Sélectionner les produits de base ⇒ (Maintenir et modifier les systèmes d'information ⇒ (Mettre en place des systèmes d'information ⇒ (Concevoir des systèmes d'information ⇒ (Créer des systèmes critiques pour la sécurité ⇒ Utiliser bien les fonctionnalités de traitement de texte))))))))))

R(29)= (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte)

R(30)= (Former et soutenir les utilisateurs de tableurs ⇒ (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte))

R(31)= (Gérer des projets critiques pour la sécurité ⇒ (Former et soutenir les utilisateurs de tableurs ⇒ (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte)))

R(32)= (Mettre en œuvre des systèmes intégrés ⇒ (Gérer des projets critiques pour la sécurité ⇒ (Former et soutenir les utilisateurs de tableurs ⇒ (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte))))

R(33)= (Concevoir un appareil convivial pour l'homme \Rightarrow (Mettre en œuvre des systèmes intégrés \Rightarrow (Gérer des projets critiques pour la sécurité \Rightarrow (Former et soutenir les utilisateurs de tableurs \Rightarrow (Utiliser bien les fonctionnalités de la feuille de calcul \Rightarrow Former et soutenir les utilisateurs de traitement de texte))))))

R(34)= (Définir les exigences du système d'information \Rightarrow (Concevoir un appareil convivial pour l'homme \Rightarrow (Mettre en œuvre des systèmes intégrés \Rightarrow (Gérer des projets critiques pour la sécurité \Rightarrow (Former et soutenir les utilisateurs de tableurs \Rightarrow (Utiliser bien les fonctionnalités de la feuille de calcul \Rightarrow Former et soutenir les utilisateurs de traitement de texte))))))

R(35)= (Modéliser et concevoir une base de données \Rightarrow (Définir les exigences du système d'information \Rightarrow (Concevoir un appareil convivial pour l'homme \Rightarrow (Mettre en œuvre des systèmes intégrés \Rightarrow (Gérer des projets critiques pour la sécurité \Rightarrow (Former et soutenir les utilisateurs de tableurs \Rightarrow (Utiliser bien les fonctionnalités de la feuille de calcul \Rightarrow Former et soutenir les utilisateurs de traitement de texte))))))

R(36)= (Configurer les produits de base \Rightarrow (Modéliser et concevoir une base de données \Rightarrow (Définir les exigences du système d'information \Rightarrow (Concevoir un appareil convivial pour l'homme \Rightarrow (Mettre en œuvre des systèmes intégrés \Rightarrow (Gérer des projets critiques pour la sécurité \Rightarrow (Former et soutenir les utilisateurs de tableurs \Rightarrow (Utiliser bien les fonctionnalités de la feuille de calcul \Rightarrow Former et soutenir les utilisateurs de traitement de texte))))))

R(37)= (Gérer les bases \Rightarrow (Configurer les produits de base \Rightarrow (Modéliser et concevoir une base de données \Rightarrow (Définir les exigences du système d'information \Rightarrow (Concevoir un appareil convivial pour l'homme \Rightarrow (Mettre en œuvre des systèmes intégrés \Rightarrow (Gérer des projets critiques pour la sécurité \Rightarrow (Former et soutenir les utilisateurs de tableurs \Rightarrow (Utiliser bien les fonctionnalités de la feuille de calcul \Rightarrow Former et soutenir les utilisateurs de traitement de texte))))))

R(38)= (Former et soutenir les utilisateurs de bases de données ⇒ (Gérer les bases ⇒ (Configurer les produits de base ⇒ (Modéliser et concevoir une base de données ⇒ (Définir les exigences du système d'information ⇒ (Concevoir un appareil convivial pour l'homme ⇒ (Mettre en œuvre des systèmes intégrés ⇒ (Gérer des projets critiques pour la sécurité ⇒ (Former et soutenir les utilisateurs de tableurs ⇒ (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte))))))))))

R(39)= (Développer un plan de ressources informatiques ⇒ (Former et soutenir les utilisateurs de bases de données ⇒ (Gérer les bases ⇒ (Configurer les produits de base ⇒ (Modéliser et concevoir une base de données ⇒ (Définir les exigences du système d'information ⇒ (Concevoir un appareil convivial pour l'homme ⇒ (Mettre en œuvre des systèmes intégrés ⇒ (Gérer des projets critiques pour la sécurité ⇒ (Former et soutenir les utilisateurs de tableurs ⇒ (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte))))))))))

R(40)= (Mettre à niveau des ressources de calendrier / budget ⇒ (Développer un plan de ressources informatiques ⇒ (Former et soutenir les utilisateurs de bases de données ⇒ (Gérer les bases ⇒ (Configurer les produits de base ⇒ (Modéliser et concevoir une base de données ⇒ (Définir les exigences du système d'information ⇒ (Concevoir un appareil convivial pour l'homme ⇒ (Mettre en œuvre des systèmes intégrés ⇒ (Gérer des projets critiques pour la sécurité ⇒ (Former et soutenir les utilisateurs de tableurs ⇒ (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte))))))))))

R(41)= (Sélectionner les composants réseau ⇒ (Mettre à niveau des ressources de calendrier / budget ⇒ (Développer un plan de ressources informatiques ⇒ (Former et soutenir les utilisateurs de bases de données ⇒ (Gérer les bases ⇒ (Configurer les produits de base ⇒ (Modéliser et concevoir une base de données ⇒ (Définir les exigences du système d'information ⇒ (Concevoir un appareil convivial pour l'homme ⇒ (Mettre en œuvre des systèmes intégrés ⇒ (Gérer des projets critiques pour la sécurité ⇒ (Former et soutenir les utilisateurs de tableurs ⇒ (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte))))))))))

R(42)= (Gérer les ressources informatiques mobiles ⇒ (Sélectionner les composants réseau ⇒ (Mettre à niveau des ressources de calendrier / budget ⇒ (Développer un plan de ressources informatiques ⇒ (Former et soutenir les utilisateurs de bases de données ⇒ (Gérer les bases ⇒ (Configurer les produits de base ⇒ (Modéliser et concevoir une base de données ⇒ (Définir les exigences du système d'information ⇒ (Concevoir un appareil convivial pour l'homme ⇒ (Mettre en œuvre des systèmes intégrés ⇒ (Gérer des projets critiques pour la sécurité ⇒ (Former et soutenir les utilisateurs de tableurs ⇒ (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte))))))))))

R(43)= (Gérer la présence sur le Web d'une organisation ⇒ (Gérer les ressources informatiques mobiles ⇒ (Sélectionner les composants réseau ⇒ (Mettre à niveau des ressources de calendrier / budget ⇒ (Développer un plan de ressources informatiques ⇒ (Former et soutenir les utilisateurs de bases de données ⇒ (Gérer les bases ⇒ (Configurer les produits de base ⇒ (Modéliser et concevoir une base de données ⇒ (Définir les exigences du système d'information ⇒ (Concevoir un appareil convivial pour l'homme ⇒ (Mettre en œuvre des systèmes intégrés ⇒ (Gérer des projets critiques pour la sécurité ⇒ (Former et soutenir les utilisateurs de tableurs ⇒ (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte))))))))))

R(44)= (Configurer et intégrer des systèmes d'apprentissage en ligne ⇒ (Gérer la présence sur le Web d'une organisation ⇒ (Gérer les ressources informatiques mobiles ⇒ (Sélectionner les composants réseau ⇒ (Mettre à niveau des ressources de calendrier / budget ⇒ (Développer un plan de ressources informatiques ⇒ (Former et soutenir les utilisateurs de bases de données ⇒ (Gérer les bases ⇒ (Configurer les produits de base ⇒ (Modéliser et concevoir une base de données ⇒ (Définir les exigences du système d'information ⇒ (Concevoir un appareil convivial pour l'homme ⇒ (Mettre en œuvre des systèmes intégrés ⇒ (Gérer des projets critiques pour la sécurité ⇒ (Former et soutenir les utilisateurs de tableurs ⇒ (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte))))))))))))))

R(45)= (Développer des solutions d'affaires ⇒ (Configurer et intégrer des systèmes d'apprentissage en ligne ⇒ (Gérer la présence sur le Web d'une organisation ⇒ (Gérer les ressources informatiques mobiles ⇒ (Sélectionner les composants réseau ⇒ (Mettre à niveau des ressources de calendrier / budget ⇒ (Développer un plan de ressources informatiques ⇒ (Former et soutenir les utilisateurs de bases de données ⇒ (Gérer les bases ⇒ (Configurer les produits de base ⇒ (Modéliser et concevoir une base de données ⇒ (Définir les exigences du système d'information ⇒ (Concevoir un appareil convivial pour l'homme ⇒ (Mettre en œuvre des systèmes intégrés ⇒ (Gérer des projets critiques pour la sécurité ⇒ (Former et soutenir les utilisateurs de tableurs ⇒ (Utiliser bien les fonctionnalités de la feuille de calcul ⇒ Former et soutenir les utilisateurs de traitement de texte))))))))))))))

R(46)= (Concevoir des périphériques d'ordinateur ⇒ Concevoir des systèmes embarqués)

R(47)= (Concevoir des systèmes de capteurs complexes ⇒ (Concevoir des périphériques d'ordinateur ⇒ Concevoir des systèmes embarqués))

R(48)= (Concevoir une puce ⇒ (Concevoir des systèmes de capteurs complexes ⇒ (Concevoir des périphériques d'ordinateur ⇒ Concevoir des systèmes embarqués)))

R(49)= (Programmer une puce \Rightarrow (Concevoir une puce \Rightarrow (Concevoir des systèmes de capteurs complexes \Rightarrow (Concevoir des périphériques d'ordinateur \Rightarrow Concevoir des systèmes embarqués))))

R(50)= (Concevoir un ordinateur \Rightarrow (Programmer une puce \Rightarrow (Concevoir une puce \Rightarrow (Concevoir des systèmes de capteurs complexes \Rightarrow (Concevoir des périphériques d'ordinateur \Rightarrow Concevoir des systèmes embarqués))))))

R(51)= (Former les utilisateurs à utiliser les systèmes d'information \Rightarrow (Concevoir un ordinateur \Rightarrow (Programmer une puce \Rightarrow (Concevoir une puce \Rightarrow (Concevoir des systèmes de capteurs complexes \Rightarrow (Concevoir des périphériques d'ordinateur \Rightarrow Concevoir des systèmes embarqués))))))

R(52)= (Installer le réseau informatique \Rightarrow (Former les utilisateurs à utiliser les systèmes d'information \Rightarrow (Concevoir un ordinateur \Rightarrow (Programmer une puce \Rightarrow (Concevoir une puce \Rightarrow (Concevoir des systèmes de capteurs complexes \Rightarrow (Concevoir des périphériques d'ordinateur \Rightarrow Concevoir des systèmes embarqués))))))

R(53)= (Gérer les ressources de communication \Rightarrow Développer un plan d'information d'entreprise)

Tableau 8.12 – Les 53 méta-règles par rapport aux capacités d'aptitudes des diplômés en CS.

8.3.5.2 Quelques interprétations

- **R(7)= (Implémenter un logiciel de recherche d'information \Rightarrow (Concevoir un système de gestion de base de données (par exemple : Oracle) \Rightarrow (Produire des graphiques ou des logiciels de jeux \Rightarrow (Faire de la programmation à petite échelle \Rightarrow (Déterminer si des solutions plus rapides sont possibles \Rightarrow (Développer des programmes de preuve de concept \Rightarrow (Développer des solutions aux problèmes de programmation \Rightarrow Prouver des résultats théoriques))))))**,

R(1) \subset R(2) \subset R(3) \subset R(4) \subset R(5) \subset R(6) \subset R(7) : Afin d'implémenter un logiciel de recherche d'information, il faut au moins être capable de concevoir un système de gestion de base de données, de produire des graphiques ou des logiciels de jeux, de faire de la programmation à petite échelle, de déterminer si des

domaines à la fois à l'intérieur et à l'extérieur des frontières CS traditionnelles. Le matériel électif couvre des sujets qui préparent les étudiants à contribuer à des efforts tels que la biologie computationnelle, la bioinformatique, l'éco-informatique, la finance computationnelle et la chimie computationnelle.

- **DS** - Structures discrètes (*Discrete Structures*): Les concepts couverts dans le noyau ne sont pas nouveaux, mais un certain temps de couverture est passé de la logique à la probabilité discrète, reflétant l'utilisation croissante de la probabilité en tant qu'outil mathématique en informatique.
- **GV** - Graphisme et visualisation (*Graphics and Visualization*): Le stockage de signaux analogiques sous forme numérique est une idée informatique générale, tout comme le stockage d'informations par rapport au recalcul. (Ce résultat apparaît dans SF, également)
- **HCI** - Interaction Homme-Machine (*Human-Computer Interaction*): Bien que les heures de base n'aient pas augmenté, il y a un changement d'orientation dans ce domaine de connaissances pour reconnaître l'importance accrue des méthodes de conception et des approches interdisciplinaires au sein de la spécialité.
- **IAS** - Assurance de l'information et sécurité (*Information Assurance and Security*): Ceci est un nouveau domaine de connaissances. Tous ces résultats reflètent l'importance croissante de la profession pour la sécurité. Le domaine de connaissances du IAS contient des unités de connaissance de sécurité et d'assurance spécifiques; Cependant, il est également fortement intégré avec de nombreux autres domaines de connaissances.
- **IM** - Gestion de l'information (*Information Management*): Les principaux résultats de ce domaine de connaissances reflètent des sujets plus larges qu'un cours de base de données classique. Ils peuvent facilement être couverts dans un cours de base de données traditionnel, mais ils doivent être explicitement abordés.
- **IS** - Systèmes intelligents (*Intelligent Systems*): L'accent a été mis davantage sur l'apprentissage automatique que par le passé. Des conseils supplémentaires ont été fournis sur ce que l'on attend des étudiants en ce qui concerne la compréhension des défis de la mise en œuvre et de l'utilisation de systèmes intelligents.

- **NC** - Réseaux et communications (*Networking and Communications*): L'accent est davantage mis sur la comparaison des réseaux IP et Ethernet et une attention accrue aux réseaux sans fil. Un sujet connexe est la livraison fiable. Ici, on a aussi mis l'accent sur la mise en œuvre des protocoles et des applications.
- **OS** - Systèmes d'exploitation (*Operating Systems*): Ce domaine de connaissances est structuré de manière à être complémentaire des principes fondamentaux des systèmes, de la mise en réseau et de la communication, de l'assurance et de la sécurité de l'information et des domaines de connaissances parallèles et distribués. Alors que certains soutiennent que l'administration du système est le domaine de l'informatique et non du CS, le groupe de travail estime que chaque étudiant devrait avoir la capacité de mener des activités administratives de base, en particulier celles qui ont un impact sur le contrôle d'accès. La sécurité et la protection étaient facultatives dans CC2001, alors qu'elles étaient incluses dans le noyau dans CS2008 (Cassel et al., 2008). Ils apparaissent dans le noyau ici aussi. Réalisation de la mémoire virtuelle en utilisant le matériel et les logiciels a été déplacé pour être un résultat d'apprentissage électif (OS / machines virtuelles).
- **PBD** - Développement basé sur une plate-forme (Platform-based Development) concerne la conception et le développement d'applications logicielles qui résident sur des plates-formes logicielles spécifiques. Contrairement à la programmation générale, le développement basé sur une plateforme prend en compte les contraintes spécifiques à la plate-forme. Par exemple, la programmation Web, le développement multimédia, l'informatique mobile, le développement d'applications et la robotique sont des exemples de plates-formes pertinentes qui fournissent des services / API / matériel spécifiques qui limitent le développement. De telles plates-formes sont caractérisées par l'utilisation d'API spécialisées, de mécanismes distincts de livraison / mise à jour et d'abstraction au niveau de la machine. Le développement basé sur une plateforme peut être appliqué sur un large éventail d'écosystèmes.
- **PD** - Calcul parallèle et distribué (*Parallel and Distributed Computing*): Il s'agit d'un nouveau domaine de connaissances qui démontre la nécessité pour les étudiants de travailler dans des environnements parallèles et distribués.

Cette tendance a été initialement identifiée, mais non incluse, dans le corpus de connaissances CS2008 (Cassel et al., 2008). Il est explicite ici pour refléter le fait qu'une certaine familiarité avec ce sujet est devenue essentielle pour tous les étudiants de premier cycle en informatique.

- **PL** - Langages de programmation (*Programming Languages*): Pour le matériel de base, les résultats ont été rendus plus uniformes et généraux en refactorisant du matériel sur la programmation orientée objet, la programmation fonctionnelle et la programmation orientée événements qui se trouvait dans plusieurs domaines de connaissances dans CC2001 (Chang et al., 2001). Programmation avec moins d'état mutable et avec plus d'utilisation des fonctions d'ordre supérieur (comme la carte et réduire) ont une plus grande importance. Pour le matériel facultatif, il y a plus de profondeur sur les constructions de langage avancées, les systèmes de types, l'analyse statique à des fins autres que l'optimisation du compilateur, et les systèmes d'exécution, en particulier la récupération de place.
- **SDF** - Principes de base du développement logiciel (*Software Development Fundamentals*): Ce nouveau domaine de connaissances rassemble les concepts fondamentaux et les compétences nécessaires au développement de logiciels. Il est dérivé du domaine de connaissances Fondamentaux de programmation de CC2001, mais tire également des matériaux d'analyse de base des algorithmes et de la complexité, des processus de développement du génie logiciel, des structures de données fondamentales des structures discrètes et des langages de programmation. Le matériel spécifique à des paradigmes de programmation particuliers (par exemple, orienté objet, fonctionnel) a été déplacé vers des langages de programmation pour permettre un traitement plus uniforme avec un matériau complémentaire.
- **SE** - Génie logiciel (*Software Engineering*): Les changements dans ce domaine de connaissances introduisent ou requièrent des sujets tels que le refactoring, la programmation sécurisée, la modélisation de code, les révisions de code, les contrats, la participation d'équipe et l'amélioration des processus. Ces sujets, qui reflètent la prise de conscience croissante des processus logiciels dans l'industrie, sont essentiels à tout niveau de développement de logiciels modernes, et devraient être utilisés pour les projets de développement de logiciels tout

au long du programme. Des modèles de processus agiles ont été ajoutés.

- **SF** - Principes de base des systèmes (*Systems Fundamentals*): Ceci est un nouveau domaine de connaissances. Ses résultats reflètent la refactorisation des domaines de connaissances pour identifier des thèmes communs à travers les domaines de connaissances liés aux systèmes existants (en particulier, les systèmes d'exploitation, les réseaux et l'architecture informatique). Les nouveaux domaines thématiques transversaux comprennent le parallélisme, les communications, la performance, la proximité, la virtualisation / isolation et la fiabilité.
- **SP** - Questions sociales et pratique professionnelle (*Social Issues and Professional Practice*): Ces résultats dans ce domaine de connaissances reflètent un changement dans la dernière décennie vers la compréhension de la propriété intellectuelle liée à la propriété intellectuelle numérique et à la gestion des droits numériques, la nécessité d'une prise de conscience mondiale et une préoccupation croissante pour la vie privée à l'ère numérique. Ils reconnaissent en outre l'impact énorme que l'informatique a eu sur la société dans son ensemble en mettant l'accent sur un avenir durable et en plaçant des responsabilités supplémentaires sur les professionnels de l'informatique. Les résultats identifient également les besoins vitaux pour l'éthique professionnelle, le développement professionnel, la communication professionnelle, et la capacité à collaborer en personne et à distance à travers les fuseaux horaires.

Dans notre étude, nous nous intéressons le domaine de connaissance SDF (*Software Development Fundamentals*): contenant le thème algorithmique.

8.3.7 Principes de base du développement logiciel (SDF)

(Sahami et al., 2013) La maîtrise du processus de développement logiciel est une condition préalable à l'étude de la plupart des sciences informatiques. Afin d'utiliser les ordinateurs pour résoudre les problèmes efficacement, les étudiants doivent être compétents pour lire et écrire des programmes dans plusieurs langages de programmation. Au-delà des compétences en programmation, cependant, ils doivent être capables de concevoir et d'analyser des algorithmes, de sélectionner des paradigmes appropriés et d'utiliser des outils modernes de développement et de test. Ce

domaine de connaissances rassemble les concepts fondamentaux et les compétences liées au processus de développement logiciel. En tant que tel, il fournit une base pour d'autres domaines de connaissances axés sur le logiciel, notamment les langages de programmation, les algorithmes et la complexité, et l'ingénierie logicielle.

Il est important de noter que cette zone de connaissance est distincte de l'ancienne zone de connaissance des principes de programmation de **PP** CC2001 (Chang et al., 2001) et **PF** de CC2005 (Shackelford et al., 2005). Alors que ce domaine de connaissances se concentrait exclusivement sur les compétences en programmation requises dans un cours d'introduction en informatique, ce nouveau domaine de connaissances vise à remplir un objectif beaucoup plus large. Il se concentre sur l'ensemble du processus de développement de logiciels, en identifiant les concepts et les compétences qui devraient être maîtrisés dans la première année d'un programme d'informatique. Cela comprend la conception et l'analyse simple d'algorithmes (*Algorithms and Design*) SDF1, de concepts de programmation fondamentaux (*Fundamental Programming Concepts*) SDF2 et de structures de données (*Fundamental Data Structures*) SDF3, ainsi que de méthodes et d'outils de développement de logiciels de base (*Development Methods*) SDF4. En raison de son objectif plus large, le domaine de connaissances Développement logiciel comprend des concepts et compétences fondamentaux qui pourraient naturellement être répertoriés dans d'autres domaines de connaissance orientés logiciels (Ex., Programmation de langages de programmation, analyse algorithmique simple d'algorithmes et complexité, développement simple, méthodologies de génie logiciel). De même, chacun de ces domaines de connaissances contiendra du matériel plus avancé qui s'appuie sur les concepts fondamentaux et les compétences devant être acquis.

Bien que plus large que les anciens principes fondamentaux de la programmation, ce domaine de connaissances permet toujours une grande flexibilité dans la conception des programmes de première année. Par exemple, l'unité *Fundamental Programming Concepts* identifie uniquement les concepts communs à tous les paradigmes de programmation. On s'attend à ce qu'un instructeur choisisse un ou plusieurs paradigmes de programmation (Ex. Programmation orientée objet, programmation fonctionnelle, écriture de scripts) pour illustrer ces concepts de programmation et extraire du contenu spécifique au paradigme du domaine de connaissance des langages de programmation. . De même, un instructeur pourrait choisir de mettre l'accent sur

l'analyse formelle (Ex., Big-Oh, calculabilité) ou les méthodologies de conception (p. Ex. Projets d'équipe, cycle de vie logiciel), intégrant ainsi les heures de programmation et/ou domaines de connaissances en génie logiciel.

Lorsque l'on considère les heures allouées à chaque unité de connaissances, il convient de noter que ces heures reflètent la quantité minimale de couverture de classe requise pour introduire le matériel. De nombreux sujets de développement de logiciel réapparaîtront et seront renforcés par des sujets ultérieurs (par exemple, l'application de constructions d'itération lors du traitement des listes). En outre, la maîtrise des concepts et des compétences de ce domaine de connaissances nécessite une expérience significative en développement de logiciels en dehors de la classe.

Pour les autres modèles, comme le modèle français, l'informatique en tant que discipline scientifique regroupe le CS et tout ou partie des autres disciplines du modèle anglo-saxon. L'acquisition des fondamentaux de la programmation et la capacité à réaliser des programmes simples sont toujours deux composantes essentielles dans l'enseignement de l'informatique.

Architecture des enseignements

Le rapport de l'ACM/IEEE ([Sahami et al., 2013](#)) structure les connaissances en trois niveaux :

- **les domaines**: représentent les différents champs disciplinaires ;
- **les unités**: constituent les modules thématiques de chaque domaine ;
- **les thèmes** qui composent les cours de chaque unité.

Détails des six approches des enseignements introductifs

- *Approche impérative* : Cette approche est la plus répandue, elle aborde l'introduction à la programmation via un style de programmation impératif. Les enseignements se concentrent sur les aspects impératifs du langage : expressions, structures de contrôle, procédures et fonctions, etc.
- *Approche orientée objet*: se concentre également sur la programmation, elle met l'accent sur les principes de la conception OO dès le début des enseignements. Les cours portent sur les notions d'objets et d'héritage afin de familiariser très tôt les étudiants à ces concepts. Après avoir expérimenté ces notions dans un contexte de programmes interactifs simples, les enseignements s'orientent vers l'introduction des traditionnelles structures de contrôle mais toujours dans un contexte lié à la conception OO.

- *Approche fonctionnelle*: se caractérise par l'utilisation d'un langage fonctionnel qui présente les avantages suivants : ce paradigme de programmation est peu connu des étudiants ce qui rend les promotions plus homogènes ; la syntaxe minimale des langages fonctionnels permet de concentrer le cours sur les notions fondamentales ; plusieurs concepts importants, comme la récursivité, les structures de données liées et les fonctions sont exprimées très naturellement et peuvent être abordées plus tôt dans la formation.
- *Approche étendue*: d'après cette approche, aborder l'enseignement de l'informatique par la programmation donne une vision limitée de la discipline aux étudiants. En effet, l'informatique est un domaine en constante évolution qui inclut de nombreuses activités au delà de la simple programmation.
- *Approche algorithmique*: dans cette approche, les concepts basiques de l'informatique sont introduits à l'aide de pseudo-codes au lieu de langages exécutables. En introduisant les concepts basiques d'algorithmique sans langage particulier, cette approche minimise les préoccupations liées aux détails syntaxiques de la programmation.
- *Approche matérielle*: Cette approche enseigne les bases de l'informatique au niveau de la machine puis intègre des concepts de plus en plus abstraits.

L'enseignement par rapport à l'approche matérielle et l'approche algorithmique peut être déjà entamé au lycée, et considérées comme prérequis, pour les nouveaux étudiants qui suivent le cursus informatique à l'université. Cela les aide à apprendre sans difficultés le SDF afin d'atteindre les objectifs déjà sus-mentionnés : « **acquérir les fondamentaux de la programmation** » et être capable de « **réaliser des programmes simples** ».

« *Un programme est un texte, l'histoire qu'il raconte un algorithme, la programmation son écriture* » ([Ganascia, 1998](#)).

Tandis qu'un *langage de programmation* est un ensemble d'*instructions* et de *règles* syntaxiques compréhensible par l'ordinateur et permettant de créer des algorithmes. Un *programme* est la *traduction d'un algorithme* dans le *langage de programmation* utilisé. On peut dire que l'algorithme est la base fondamentale de programme. Il faut maîtriser l'algorithmique avant de se focaliser à la programmation. Mais on peut faire tous les deux en parallèle.

8.4 Formation en informatique à l'ENSET

Les écoles normales supérieures (ENS) dont l'ENSET (Ecole Normale Supérieure pour l'Enseignement Technique) sont chargées d'assurer suivant le décret n° 93-394 du 20 juillet 1993 les formations dans le domaine de la Science de l'Education (SeD) suivantes :

- La formation initiale des enseignants du secondaire (collège et lycée) ;
- La formation des encadreurs de l'éducation ;
- Le perfectionnement et le recyclage des enseignants et encadreurs du secondaire ;
- La recherche en matière d'enseignement et d'éducation ;
- La formation est non payante et est offerte pour les étudiants ayant leur diplôme de baccalauréat série (C, D ou Technique), pour les étudiants étrangers et pour les fonctionnaires ayant remplis certaines conditions.

La formation des enseignants est un facteur déterminant de la réussite éducative. Toutes les études montrent en effet que les progrès d'un élève dépendent de manière significative de la qualité de ses enseignants et donc de la qualité de la formation reçue par le professeur auquel il est confié. C'est ce que l'on appelle l'« *effet maître* ». Il est mondialement connu que les programmes de formation initiale les plus efficaces sont ceux qui garantissent un équilibre approprié entre la théorie et la pratique et la collaboration entre les enseignants : les futurs enseignants découvrent, en semestre 6 pour le grade Licence et en semestre 10 pour le grade Master, le contact avec les élèves, passent durant un temps relativement élevé (respectivement 9 et 12 semaines en L et M) dans les classes et bénéficient d'un soutien de proximité et de qualité, notamment grâce à un système de double tutorat par des enseignants qualifiés : le Tuteur de Métier et une équipe des Enseignants chercheurs nommés par l'ENSET.

8.4.1 Les capacités d'aptitudes des jeunes en licence informatique de l'ENSET

A l'issue de la Licence, l'étudiant diplômé sera :

- pré professionnalisé en milieu scolaire (Collège et niveau seconde) : Capacité d'enseigner les disciplines de sa pré-spécialisation ; être initié en psychopéda-

- gogie et didactique spécifiques de la matière adaptée aux adolescents, et aux techniques de la communication (oral, écrit, TICE) ; s'adapter aux contextes évolutifs ; connaître le système éducatif de base de Madagascar ; avoir un esprit ouvert pour transférer les connaissances acquises durant son parcours ;
- pré professionnalisé dans les secteurs relevant de son parcours ;
 - pré professionnalisé dans le secteur de la gestion et du management des disciplines de son parcours ;
 - détenteur d'une compétence préprofessionnelle autorisant un choix d'orientation éclairé ;
 - préparé à une proche spécialisation, voire insertion professionnelle, ou à une poursuite d'étude dans un Master prolongeant son parcours ;
 - initié à la conception et à l'étude d'une formation ou d'un programme d'étude ;
 - capable de partager les connaissances acquises aux non-initiés et de mettre à niveau ces derniers ;
 - capable de Travailler et de s'intégrer dans une équipe ;
 - capable d'intervenir en entreprise (société de services, bureau d'études, etc.) en qualité de technicien spécialisé ;
 - détenteur d'un sens civique et d'une fierté nationale (aimer sa patrie, apprécier sa culture, etc.) ;
 - capable de s'auto-former.

8.4.2 Les parcours de licence en informatique

Le grade de Licence de la mention Education-Apprentissage-Didactique et Ingénierie en Mathématique et Informatique (EADIMI) du parcours Professorat d'Enseignement en Génie Informatique (PEGINFO) que l'Université d'Antsirananana propose les parcours intermédiaires suivants :

- Bases Pédagogiques en S1 (semestre 1) et S2, pour tous les étudiants nouvellement recrutés, permettent aux étudiants de bien s'informer et s'initier avant la prise de décision finale pour intégrer S3. Les UE de ce parcours sont toutes

obligatoires, car elles sont jugées comme porteuses de connaissances de base exigées par la poursuite des études supérieures.

- Professorat d'Enseignement en Génie Mathématiques et Informatique de Base (PEGMIB) en S3 apprentissage des techniques opératoires de base en mathématiques et en informatique donne à ceux qui ont l'intention de poursuivre le parcours « Professorat d'Enseignement en Génie Informatique » (PEGINFO).
- Professorat d'Enseignement en Génie Informatique (PEGINFO) de S4-S6, la possibilité d'aborder S4 et d'y prendre la décision qui leur convient afin d'obtenir la licence en Education-Apprentissage-Didactique et Ingénierie en Mathématique et Informatique (EADIMI) - Spécialité : Informatique

8.4.3 Analyse du curriculum informatique à l'ENSET

DOMAINES / GRADES	SeD					
	EADIMI					
PARCOURS	S1	S2	S3	S4	S5	S6
Bases Pédagogiques	X	X				
PEGMIB – PEGINFO Informatique			X	X	X	X

Tableau 8.13 – Le parcours licence en informatique EADIMI

Unité d'enseignement	Crédit
Semestre 6	
STAEDU06 : STAGE PEDAGOGIQUE	18
STAIN06 : STAGE INDUSTRIEL	12
TOTAL S6	30
Semestre 5	
MROINF05 : METHODES NUMERIQUES ET INITIATION A LA RECHERCHE OPERATIONNELLE	8
MGLINF05 : METHODES ET OUTILS DE GENIE LOGICIEL	4
PSCLNG05 : PRATIQUE SCIENTIFIQUE DE LANGUES ETRANGERES	3
PENEDU05 : PREPARATION A LA PRATIQUE DE L'ENSEIGNEMENT	4
PRAINP05 : PROGRAMMATION AVANCEE	6
SGBINF05 : SYSTÈME DE GESTION DE BASE DE DONNEES (SGBD)	5
TOTAL S5	30
Semestre 4	
STAMTH04 : STATISTIQUE ET COMPLEMENTES D'ANALYSE	5
PRNSINF04 : PROGRAMMATION STRUCTUREE	6
BDDINF04 : BASE DE DONNEES	5
ALAINF04 : ALGORITHMIQUE AVANCEE	5
ADRINF04 : ADMINISTRATION DE RESEAUX	6
TSIINF04 : TECHNOLOGIES ET SOCIETE DE L'INFORMATION	3
TOTAL S4	30
Semestre 3	
SIEEDU03 : SITUATIONS EDUCATIVES	4
TEXLNG03 : TECHNIQUES D'EXPRESSION DE LANGUES ETRANGERES	4
ALIMTH03 : ALGEBRE LINEAIRE	4
EAVMTH03 : ESPACES AFFINES ET VECTORIELS	5
PCAMTH03 : PROBABILITE ET COMPLEMENTES D'ANALYSE	6
ARSINF03 : ARCHITECTURE DES SYSTÈMES INFORMATIQUES	7
TOTAL S3	30
Semestre 2	
ALGMTH02 : ALGEBRE ET GEOMETRIE	6
MCPMEC02 : MECANIQUE DU POINT	4
ELMELC02 : ELECTROSTATIQUE ET MAGNETOSTATIQUE	4
STPMEC02 : STATIQUE PLANE	4
CSTMTH02 : CALCUL NUMERIQUE ET STATISTIQUE	6
PCHPHY02 : PHYSIQUE ET CHIMIE	6
TOTAL S2	30
Semestre 1	
LAELNG01 : LANGUES ETRANGERES	3
DOAINF01 : DEMARCHES SCIENTIFIQUES ET OUTILS D'APPRENTISSAGE	5
ANMMTH01 : ANALYSE MATHEMATIQUE	5
ECTEDU01 : ECOLE ET TRANSMISSION	4
DSCDES01 : DESSINS ET SCHEMA	8
EDEMTH01 : ÉQUATIONS DIFFERENTIELLES ORDINAIRES ET ESPACE VECTORIEL	5
TOTAL S1	30

Tableau 8.14 – Unités d'Enseignements de licence en informatique EADIMI

8.4.3.1 Liens implicatifs entre les UE du parcours PEGINFO

En traitant le tableau 8.14 à l'aide de l'outil CHIC- M_{GK} , nous obtenons les 25 méta-règles représentées dans le tableau 8.15 ci-dessous :

Regle	
R(1)=	(TECHNIQUES D'EXPRESSION DE LANGUES ETRANGERES \Rightarrow SITUATIONS EDUCATIVES)
R(2)=	(ALGEBRE LINEAIRE \Rightarrow (TECHNIQUES D'EXPRESSION DE LANGUES ETRANGERES \Rightarrow SITUATIONS EDUCATIVES))
R(3)=	(MECANIQUE DU POINT \Rightarrow (ALGEBRE LINEAIRE \Rightarrow (TECHNIQUES D'EXPRESSION DE LANGUES ETRANGERES \Rightarrow SITUATIONS EDUCATIVES)))

R(4)= (ELECTROSTATIQUE ET MAGNETOSTATIQUE \Rightarrow (MECANIQUE DU POINT \Rightarrow (ALGEBRE LINEAIRE \Rightarrow (TECHNIQUES D'EXPRESSION DE LANGUES ETRANGERES \Rightarrow SITUATIONS EDUCATIVES))))

R(5)= (STATIQUE PLANE \Rightarrow (ELECTROSTATIQUE ET MAGNETOSTATIQUE \Rightarrow (MECANIQUE DU POINT \Rightarrow (ALGEBRE LINEAIRE \Rightarrow (TECHNIQUES D'EXPRESSION DE LANGUES ETRANGERES \Rightarrow SITUATIONS EDUCATIVES))))

R(6)= (ECOLE ET TRANSMISSION \Rightarrow (STATIQUE PLANE \Rightarrow (ELECTROSTATIQUE ET MAGNETOSTATIQUE \Rightarrow (MECANIQUE DU POINT \Rightarrow (ALGEBRE LINEAIRE \Rightarrow (TECHNIQUES D'EXPRESSION DE LANGUES ETRANGERES \Rightarrow SITUATIONS EDUCATIVES))))))

R(7)= (METHODES ET OUTILS DE GENIE LOGICIEL \Rightarrow (ECOLE ET TRANSMISSION \Rightarrow (STATIQUE PLANE \Rightarrow (ELECTROSTATIQUE ET MAGNETOSTATIQUE \Rightarrow (MECANIQUE DU POINT \Rightarrow (ALGEBRE LINEAIRE \Rightarrow (TECHNIQUES D'EXPRESSION DE LANGUES ETRANGERES \Rightarrow SITUATIONS EDUCATIVES))))))

R(8)= (PREPARATION A LA PRATIQUE DE L'ENSEIGNEMENT \Rightarrow (METHODES ET OUTILS DE GENIE LOGICIEL \Rightarrow (ECOLE ET TRANSMISSION \Rightarrow (STATIQUE PLANE \Rightarrow (ELECTROSTATIQUE ET MAGNETOSTATIQUE \Rightarrow (MECANIQUE DU POINT \Rightarrow (ALGEBRE LINEAIRE \Rightarrow (TECHNIQUES D'EXPRESSION DE LANGUES ETRANGERES \Rightarrow SITUATIONS EDUCATIVES))))))

R(9)= (DEMARCHES SCIENTIFIQUES ET OUTILS D'APPRENTISSAGE \Rightarrow ESPACES AFFINES ET VECTORIELS)

R(10)= (ANALYSE MATHEMATIQUE \Rightarrow (DEMARCHES SCIENTIFIQUES ET OUTILS D'APPRENTISSAGE \Rightarrow ESPACES AFFINES ET VECTORIELS))

R(11)= (ÉQUATIONS DIFFERENTIELLES ORDINAIRES ET ESPACE VECTORIEL \Rightarrow (ANALYSE MATHEMATIQUE \Rightarrow (DEMARCHES SCIENTIFIQUES ET OUTILS D'APPRENTISSAGE \Rightarrow ESPACES AFFINES ET VECTORIELS)))

R(12)= (SYSTÈME DE GESTION DE BASE DE DONNEES (SGBD) ⇒ (ÉQUATIONS DIFFERENTIELLES ORDINAIRES ET ESPACE VECTORIEL ⇒ (ANALYSE MATHEMATIQUE ⇒ (DEMARCHES SCIENTIFIQUES ET OUTILS D'APPRENTISSAGE ⇒ ESPACES AFFINES ET VECTORIELS))))

R(13)= (STATISTIQUE ET COMPLEMENTS D'ANALYSE ⇒ (SYSTÈME DE GESTION DE BASE DE DONNEES (SGBD) ⇒ (ÉQUATIONS DIFFERENTIELLES ORDINAIRES ET ESPACE VECTORIEL ⇒ (ANALYSE MATHEMATIQUE ⇒ (DEMARCHES SCIENTIFIQUES ET OUTILS D'APPRENTISSAGE ⇒ ESPACES AFFINES ET VECTORIELS))))

R(14)= (BASE DE DONNEES ⇒ (STATISTIQUE ET COMPLEMENTS D'ANALYSE ⇒ (SYSTÈME DE GESTION DE BASE DE DONNEES (SGBD) ⇒ (ÉQUATIONS DIFFERENTIELLES ORDINAIRES ET ESPACE VECTORIEL ⇒ (ANALYSE MATHEMATIQUE ⇒ (DEMARCHES SCIENTIFIQUES ET OUTILS D'APPRENTISSAGE ⇒ ESPACES AFFINES ET VECTORIELS))))))

R(15)= (ALGORITHMIQUE AVANCEE ⇒ (BASE DE DONNEES ⇒ (STATISTIQUE ET COMPLEMENTS D'ANALYSE ⇒ (SYSTÈME DE GESTION DE BASE DE DONNEES (SGBD) ⇒ (ÉQUATIONS DIFFERENTIELLES ORDINAIRES ET ESPACE VECTORIEL ⇒ (ANALYSE MATHEMATIQUE ⇒ (DEMARCHES SCIENTIFIQUES ET OUTILS D'APPRENTISSAGE ⇒ ESPACES AFFINES ET VECTORIELS))))))

R(16)= (ALGEBRE ET GEOMETRIE ⇒ PROBABILITE ET COMPLEMENTS D'ANALYSE)

R(17)= (CALCUL NUMERIQUE ET STATISTIQUE ⇒ (ALGEBRE ET GEOMETRIE ⇒ PROBABILITE ET COMPLEMENTS D'ANALYSE))

R(18)= (PHYSIQUE ET CHIMIE ⇒ (CALCUL NUMERIQUE ET STATISTIQUE ⇒ (ALGEBRE ET GEOMETRIE ⇒ PROBABILITE ET COMPLEMENTS D'ANALYSE)))

R(19)=	(PROGRAMMATION AVANCEE \Rightarrow (PHYSIQUE ET CHIMIE \Rightarrow (CALCUL NUMERIQUE ET STATISTIQUE \Rightarrow (ALGEBRE ET GEOMETRIE \Rightarrow PROBABILITE ET COMPLEMENTS D'ANALYSE))))
R(20)=	(PROGRAMMATION STRUCTUREE \Rightarrow (PROGRAMMATION AVANCEE \Rightarrow (PHYSIQUE ET CHIMIE \Rightarrow (CALCUL NUMERIQUE ET STATISTIQUE \Rightarrow (ALGEBRE ET GEOMETRIE \Rightarrow PROBABILITE ET COMPLEMENTS D'ANALYSE))))
R(21)=	(ADMINISTRATION DE RESEAUX \Rightarrow (PROGRAMMATION STRUCTUREE \Rightarrow (PROGRAMMATION AVANCEE \Rightarrow (PHYSIQUE ET CHIMIE \Rightarrow (CALCUL NUMERIQUE ET STATISTIQUE \Rightarrow (ALGEBRE ET GEOMETRIE \Rightarrow PROBABILITE ET COMPLEMENTS D'ANALYSE))))))
R(22)=	(PRATIQUE SCIENTIFIQUE DE LANGUES ETRANGERES \Rightarrow LANGUES ETRANGERES)
R(23)=	(TECHNOLOGIES ET SOCIETE DE L'INFORMATION \Rightarrow (PRATIQUE SCIENTIFIQUE DE LANGUES ETRANGERES \Rightarrow LANGUES ETRANGERES))
R(24)=	(METHODES NUMERIQUES ET INITIATION A LA RECHERCHE OPERATIONNELLE \Rightarrow DESSINS ET SCHEMA)
R(25)=	(STAGE INDUSTRIEL \Rightarrow STAGE PEDAGOGIQUE)

Tableau 8.15 – Les 25 méta-règles par rapport aux UE du parcours informatique de l'ENSET

Il nous paraît difficile de comparer le curriculum informatique dans le modèle anglo-saxon qui est homogène pour chaque discipline parmi les cinq proposés, par rapport au modèle de formation de l'ENSET qui est hétérogène et multidisciplinaire. L'offre de formation en informatique de l'ENSET a été validé très récemment, et il est à remarquer que la répartition des crédits de chaque UE mérite d'être étudiée et révisée en permanence selon l'expérience vécue pendant chaque année universitaire écoulée. Ce constat a été, vérifié ne serait ce que la règle R(24) qui nous montre qu'il y a un lien implicatif entre *METHODES NUMERIQUES ET INITIATION A LA RECHERCHE OPERATIONNELLE* et *DESSINS ET SCHEMA* et difficile-

ment interprétable. Il est aussi aux responsabilités des chercheurs de l'ENSET de définir et d'établir les capacités d'aptitudes en catégorisant les UE.

D'autre part, les thèmes informatiques pour le parcours PEGINFO apparaissent rarement au 3^e premier semestre du parcours, i.e. à mi-parcours, les étudiants ne font que 2 UE informatiques (DOAINF01 au 1^{er} semestre et ARSINF03 au 3^{me} semestre Cf; Tableau 8.14). Tout d'un coup, apparaissent, 5 thèmes lourds en semestre 4 (PRSINF04 : *PROGRAMMATION STRUCTUREE*, BDDINF04 : *BASE DE DONNEES*, ALAINF04 : *ALGORITHMIQUE AVANCEE*, ADRINF04 : *ADMINISTRATION DE RESEAUX* et TSIINF04 : *TECHNOLOGIES ET SOCIETE DE L'INFORMATION*) en sachant pertinemment que les étudiants n'ont aucune formation initiale et fondamentale pour pouvoir assimiler ces-dits thèmes. Les objectifs « **acquérir les fondamentaux de la programmation** » et être capable de « **réaliser des programmes simples** » et l'objectif énoncé dans 8.4.1 tiret-1. *pré-professionnalisation en milieu scolaire (Collège et niveau seconde) : Capacité d'enseigner les disciplines de sa pré-spécialisation ; être initié en psychopédagogie et didactique spécifiques de la matière adaptée aux adolescents, et aux techniques de la communication (oral, écrit, TICE) ; s'adapter aux contextes évolutifs ; connaître le système éducatif de base de Madagascar ; avoir un esprit ouvert pour transférer les connaissances acquises durant son parcours* sont loin d'être atteints.

Une raison de plus, que nous proposons un *curriculum algorithmique* au lycée en classe de seconde, car tout d'abord, c'est là que les jeunes étudiants choisissent leur futur chemin (*point de bifurcation*) et pallier le vide de formation en informatique à Madagascar.

8.5 Sur la programmation

8.5.1 Qu'est-ce que la programmation ?

La programmation dans le domaine informatique est l'ensemble des activités qui permettent l'écriture des programmes informatiques pour leur exécution dans une machine. D'habitude, on utilise le mot « implémentation » pour désigner la mise en œuvre du programme. L'écriture en langage informatique est sous forme de « codage ». Le « code source » est utilisé pour désigner le texte, en langage de programmation, constituant le programme. Les langages de programmation permettent de

définir les ensembles d'instructions effectuées par l'ordinateur lors de l'exécution d'un programme. « *Un ordinateur sans programme ne fait absolument rien. En fait, c'est même la possibilité de suivre un programme enregistré qui sert, d'un point de vue historique, à distinguer un ordinateur d'une simple machine à calculer. Le premier ordinateur est donc le Manchester Mark I, premier calculateur à programme enregistré(...)* » (Arsac, 1987).

La programmation est alors abordée avec un langage de haut niveau de type OO. Les 6 approches sus-mentionnées exigent les thèmes : « fondamentaux de la programmation » comme une connaissance primordiale en informatique.

En CS (Cassel et al., 2008), cette connaissance est structurée sous l'appellation « **Fondamentaux de la Programmation** » (*Programming Fundamentals*) **PF** divisés en cinq unités d'enseignements (UE) :

- Fondamentaux de la construction de programmes (PF1) ;
- Algorithmes et résolution de problèmes (PF2) ;
- Structures de Données (PF3) ;
- Récursivité (PF4) ;
- Programmation événementielle (PF5).

En CS (Sahami et al., 2013), cette connaissance est structurée sous l'appellation « **Principes de base du développement logiciel** » (*Software Development Fundamentals*) **SDF** divisés en quatre unités d'enseignements (UE) :

- Conception et Analyse d'Algorithmes - *Algorithms and Design* (SDF1) ;
- Concepts de Programmation Fondamentaux - *Fundamental Programming Concepts* (SDF2) ;
- Structures de Données - *Fundamental Data Structures* (SDF3) ;
- Méthodes et Outils de Développement de Logiciels - *Development Methods* (SDF4).

Les thèmes abordés dans ces différentes unités portent sur la manipulation de variables, les structures de contrôle, l'utilisation de débogueurs, les tableaux, les pointeurs et les références, la propagation d'événements, etc.

8.5.2 Comparaison du domaine de connaissances SDF en CS à celle de l'ENSET

Dans le modèle anglo-saxon, le domaine de connaissances SDF fait partie de la discipline CS et est composé de 4 thèmes. Tandis qu'à l'ENSET, le groupement des thèmes en domaine de connaissances n'existe pas, mais nous pouvons déjà identifier l'équivalence de quelques thèmes dans les deux programmes. $SDF1 \Leftrightarrow ??$, $SDF2 \Leftrightarrow PRSINF04$: Le Concepts de Programmation Fondamentaux en CS est de même rang que la PROGRAMMATION STRUCTUREE de l'ENSET enseignée aux étudiants qui poursuivent le parcours PEGMI pendant le semestre 4 (S4), $SDF3 \Leftrightarrow ALAINF04$ et $SDF4 \Leftrightarrow MGLINF05$ mais enseignée en (S5) (Cf. Tableau 8.16).

CS-USA		PEGMI-ENSET Madagascar	
Principes de base du développement logiciel (<i>Software Development Fundamentals</i>)		??	
SDF1	Conception et Analyse d'Algorithmes - <i>Algorithms and Design</i>	??	??
SDF2	Concepts de Programmation Fondamentaux - <i>Fundamental Programming Concepts</i>	PRSINF04	PROGRAMMATION STRUCTUREE
SDF3	Structures de Données - <i>Fundamental Data Structures</i>	ALAINF04	ALGORITHMIQUE AVANCEE
SDF4	Méthodes et Outils de Développement de Logiciels - <i>Development Methods</i>	MGLINF05	METHODES ET OUTILS DE GENIE LOGICIEL

Tableau 8.16 – Tableau représentant l'équivalence de SDF par rapport à l'UE de l'ENSET

Selon les deux composantes essentielles et communes aux cinq disciplines dans le modèle anglo-saxon : « **acquérir les fondamentaux de la programmation** » et être capable de « **réaliser des programmes simples** », n'est autre que la règle **R(4)** du tableau 8.12 parmi les résultats obtenus avec l'outil CHIC- M_{GK} par rapport aux capacités d'aptitudes des diplômés en CS.

R(4) = (Faire de la programmation à petite échelle \Rightarrow (Déterminer si des solutions plus rapides sont possibles \Rightarrow (Développer des programmes

de preuve de concept \Rightarrow (Développer des solutions aux problèmes de programmation \Rightarrow Prouver des résultats théoriques))))).

Selon le tableau 8.8, nous pouvons montrer que :

- Faire de la programmation à petite échelle est dans la catégorie « **Programmation informatique** »,
- Déterminer si des solutions plus rapides sont possibles, développer des programmes de preuve de concept, développer des solutions aux problèmes de programmation, prouver des résultats théoriques sont dans la catégorie « **Algorithmes** ».

En prenant compte des deux composantes essentielles dans le modèle anglo-saxon, on peut dire que «**les fondamentaux de la programmation**» sont des «**Algorithmes**» et les «**programmes simples**» font partie de la « **Programmation informatique** ». D'une autre manière l'*algorithme* est la base de la *programmation*.

8.6 Sur l'algorithme

8.6.1 Historique

Algorithme est un terme dérivé du nom du mathématicien Muhammad ibn Musa al-Khwarizmi (Bagdad, 783-850) qui a notamment travaillé sur la théorie du système décimal (il est l'auteur d'un précis sur l'Al-Jabr qui, à l'époque, désignait la théorie du calcul, à destination des architectes, astronomes, etc.) et sur les techniques de résolution d'équations du 1^{er} et 2^{me} degré (Abrégé du calcul par la restauration et la comparaison, publié en 825). La notion d'algorithme est cependant plus ancienne : Euclide (3^e siècle av. JC, pgcd, division entière), Babyloniens (1800 av. JC, résolution de certaines équations).



Figure 8.8 – Euclide d’Alexandrie III^{ème} siècle avant JC.



Figure 8.9 – Muhammad ibn Musa al-Khwarizmi IX^{ème} siècle.

8.6.2 Qu’est-ce que l’algorithme ?

Un algorithme selon Larousse est « *l’ensemble de règles opératoires dont l’application permet de résoudre un problème énoncé au moyen d’un nombre fini d’opérations. Un algorithme peut être traduit, grâce à un langage de programmation, en un programme exécutable par un ordinateur* ».

La signification moderne de l’algorithme est assez similaire à celle de recette, processus, méthode, technique, procédure, routine, sauf que le mot « *algorithme* » connote quelque chose un peu différent, Knuth (1973) dans le premier volume de l’ouvrage « *The Art of Computer Programming* ». Un algorithme est tout simplement *un ensemble fini de règles qui donne une séquence d’opérations pour résoudre un type spécifique de problème* et qui a cinq caractéristiques suivantes :

1. **Finitude.** Un algorithme doit toujours se terminer après un nombre fini d’étapes. [...]
2. **Définitude.** Chaque étape d’un algorithme doit être définie avec précision ; les actions à mener doivent être rigoureusement et sans ambiguïté spécifiées pour chaque cas. [...]
3. **Entrée.** Un algorithme a zéro ou plusieurs entrées, c’est-à-dire les quantités qui lui sont données initialement avant le début de l’algorithme. Ces entrées sont extraites d’ensembles d’objets spécifiés. [...]

4. **Sortie.** Un algorithme a une ou plusieurs sorties, c'est-à-dire des quantités qui ont une relation spécifiée avec les entrées. [...]
5. **Efficacité.** Un algorithme devrait également être efficace. Cela signifie que toutes les opérations à effectuer dans l'algorithme doivent être suffisamment basiques pour pouvoir être effectuées exactement et dans un temps limité par un homme utilisant un crayon et du papier.

Chaque problème posé a des solutions et la notion de **résolution de problèmes** sont primordiales comme beaucoup d'auteurs (Garey et Johnson, 1979; Beauquier, Berstel, et Chrétienne, 1992; Cormen, Leiserson, Rivest, et Cazin, 1994; Aho, Hopcroft, Ullman, et Moreau, 1989). Cette notion est liée aux notions d'**entrée** et de **sortie**.

Un algorithme nécessite un langage clair (compréhension), structuré (décrire des enchaînements d'opérations), non ambigu (la programmation ne supporte pas l'ambiguïté !). Il doit de plus être « *universel* ».

Modeste (2012), par rapport aux cinq caractéristiques sus-mentionnés (Knuth, 1973) donne une définition qu'« *Un algorithme est une procédure de résolution de problème, s'appliquant à une famille d'instances du problème et produisant, en un nombre fini d'étapes constructives, effectives, non-ambiguës et organisées, la réponse au problème pour toute instance de cette famille.* »

8.6.2.1 Les cinq aspects de l'algorithme

Modeste (2012) relève des points essentiels concernant les algorithmes en cinq groupes suivants :

1. **L'aspect problème** : un algorithme répond à une question d'un problème. Il est général car il doit fonctionner pour toute instance d'une famille d'instances. Il comporte des entrées et des sorties.
Les points essentiels sont :
 - les notions d'entrée et de sortie,
 - la notion d'instance d'un problème auquel répond l'algorithme,
 - le fait qu'un algorithme réponde à une question pour toutes les instances du problème.

2. **L'aspect effectivité** : un algorithme s'exécute en un nombre fini d'étapes sur des données finies. Son traitement se fait à l'aide d'un ordinateur ou une machine sous forme de programme.

Les points essentiels sont :

- un algorithme peut être mis en œuvre par un opérateur quelconque,
- un algorithme est exécutable par un ordinateur/une machine,
- un algorithme peut être exprimé sous forme d'un programme,
- un algorithme agit sur des données finies,
- un algorithme s'exécute en un nombre fini d'étapes,
- la non-ambiguïté de la description des étapes,
- un algorithme permet de décrire des formules, des programmes de calcul ou de construction.

3. **L'aspect preuve** : un algorithme doit fournir le résultat attendu au problème pour lequel il a été conçu et cela, quelle que soit l'instance qui y a été entrée. Le fait de prouver que l'algorithme fournit le bon résultat s'appelle la correction. En outre, ce résultat doit être atteint en un nombre fini d'étapes et mérite d'être prouvé.

Les points essentiels sont :

- la preuve de correction,
- la preuve de terminaison,
- la notion d'invariant,
- l'utilisation d'algorithmes dans des preuves,
- la preuve d'existence,
- la preuve d'une propriété,
- les preuves algorithmiques, par récurrence, par induction,
- un algorithme avec sa preuve équivaut à une preuve constructive,
- la preuve d'optimalité de la solution,
- les preuves formelles par ordinateur et la vérification de preuves et de programmes.

4. **L'aspect complexité** : la notion de complexité est une notion mathématique spécifique à l'algorithmique. Elle se calcule en fonction de l'ampleur des données entrées dans l'algorithme. De plus, elle peut être mesurée en temps ou en taille de mémoire.

Les points essentiels sont :

- la complexité en temps, en espace,
- la complexité calculée au pire, en moyenne,
- la complexité comme critère de comparaison d'algorithmes,
- la recherche d'algorithmes optimaux,
- les heuristiques d'approximation de solution.

5. **L'aspect modèles théoriques** : cet aspect se base sur des modèles théoriques tels que les machines de Turing ou les fonctions récursives de Kleene. Ces différents modèles peuvent être considérés comme des définitions plus théoriques d'algorithme.

Les points essentiels sont :

- les machines de Turing, fonctions récursives et autres modèles,
- les classes de complexité (P, NP, etc...),
- les notions de décidabilité et d'indécidabilité.

8.6.3 La pensée algorithmique

Les algorithmes sont définis différemment dans la littérature, mais pour notre propos, la définition suivante est suffisante : « *Un algorithme est une méthode pour résoudre un problème selon des instructions définies avec précision* ». La pensée algorithmique est un terme qui est très souvent utilisé comme l'une des compétences les plus importantes pouvant être atteintes par l'éducation en informatique (Martinand, 2006). La pensée algorithmique est en quelque sorte un ensemble de capacités liées à la construction et à la compréhension des algorithmes : la capacité d'analyser des problèmes donnés, la capacité de spécifier précisément un problème, la capacité à trouver les actions de base adéquates au problème donné, la capacité construire un algorithme correct à un problème donné en utilisant les actions de base, la capacité de penser à tous les cas spéciaux et normaux possibles d'un problème et la

capacité d'améliorer l'efficacité d'un algorithme. La pensée algorithmique a alors un fort aspect créatif : la construction de nouveaux algorithmes qui résolvent des problèmes donnés. Si quelqu'un veut faire cela, il a besoin de la capacité de la pensée algorithmique.

La pensée algorithmique constitue l'un des concepts centraux de l'informatique. Il s'est avéré un outil polyvalent et indispensable pour la résolution de problèmes et a trouvé des applications bien au-delà de la science. Par conséquent, l'éducation informatique durable devrait être fondée sur la pensée algorithmique comme son objectif principal, développant ainsi les avantages pour une éducation générale et large. Cependant, comment pouvons-nous apporter la pensée algorithmique à l'enseignement de l'algorithme ? Voici quelques principes qu'on peut offrir aux étudiants à différents niveaux. Comme contribution principale, nous décrivons des exemples concrets de la façon d'enseigner ces paradigmes, qui ont déjà fait leurs preuves dans le passé.

L'informatique est un vaste domaine où la pensée algorithmique est au cœur de l'action. Nos curriculum se concentrent donc sur l'étude des algorithmes et de ses différents aspects.

Notre approche comprend trois aspects majeurs de la pensée algorithmique, tels que :

1. la notion de langage de programmation comme langage formel pour exprimer les algorithmes,
2. l'abstraction et l'automatisation comme stratégies centrales de résolution de problèmes,
3. les limites de la calculabilité pratique comme motivation pour améliorer les algorithmes existants.

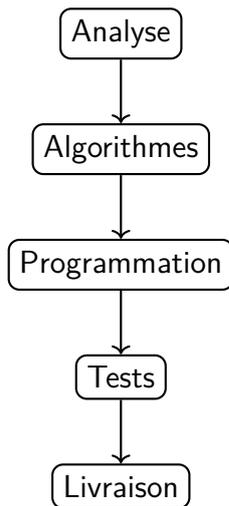
8.6.4 Procédure de résolution d'un problème

Un algorithme appartient au vaste ensemble des *marches à suivre*.

Algorithme : Procédure de résolution d'un problème contenant des opérations bien définies portant sur des informations, s'exprimant dans une séquence définie sans ambiguïté, destinée à être traduite dans un langage de programmation.

Comme toute marche à suivre, un algorithme doit s'exprimer dans un certain lan-

gage : à priori le langage naturel, mais il y a d'autres possibilités : ordinogramme, arbre programmatique, pseudo-code ou LDA (langage de description d'algorithmes).



- Lors de l'**analyse**, le problème doit être compris et clairement précisé. On abordera cette phase dans le cours d'analyse.
- Une fois le problème analysé, et avant de passer à la phase de programmation, il faut réfléchir à l'**algorithme** qui va permettre de résoudre le problème. C'est à cette phase précise que s'attache le curriculum en classe de seconde que nous proposons dans cette thèse.
- On peut alors **programmer** cet algorithme dans le langage de programmation choisi. (Scratch, Java, Pascal, Assembleur, ...)
- Vient ensuite la phase de **tests** qui ne manquera pas de montrer qu'il subsiste des problèmes qu'il faut encore corriger.
- Le produit sans bug (connu) peut être **mis en application** ou **livré** au demandeur (présenté à l'enseignant et la classe).

8.6.5 Qu'est-ce que l'algorithmique ?

L'algorithmique est l'étude et la production de règles et techniques qui sont impliquées dans la définition et la conception d'algorithmes.

L'algorithmique est la science des algorithmes : « *L'objet de l'algorithmique est la conception, l'évaluation et l'optimisation des méthodes de calcul en mathématiques et en informatique* » (Flajoret, 2015). Les objets principaux en algorithmique sont :

- **Des algorithmes fondamentaux**: algorithmes mathématiques, algorithmes de tri, algorithmes de recherche, algorithmes géométriques, algorithmes de graphes, etc. ;
- **Des structures de données**: tableaux, arbres, ensembles, listes, fichiers, etc. ;

- **L'analyse de la complexité** des algorithmes, classes de problèmes NP, etc. ;
- **Des techniques de conception d'algorithmes**: programmation dynamique, algorithmes gloutons, etc.

Ainsi l'algorithmique cherche à étudier l'algorithme indépendamment de sa mise en œuvre dans une machine.

8.7 Curriculum informatique en classe de Seconde du lycée à Madagascar

Nous avons défini dans (Chapitre 7, section 7.4) la méconnaissance, le non accès et le manque de formation en TIC des étudiants/enseignants au lycée et à l'université dans une région et la tendance est la même dans tout Madagascar. Cette étude est appuyée par l'alphabétisation informatique des étudiants dans quelques universités privés et publiques de Madagascar (Chapitre 7, section 7.5), qu'il y a une grosse lacune en algorithmique pour eux. Et aussi, d'après les analyses de curriculum de parcours en informatique, que nous avons menées dans (§8.3 et §8.4), une vide a été aussi constatée dans le programme de base de l'informatique SDF du modèle anglo-saxon à celui de l'ENSET (§8.5.2 - tableau 8.16). A l'ENSET, il manque la conception et l'analyse d'algorithme, et ceci peut être valable dans presque tous les universités de Madagascar. Outre, quelques objectifs des capacités d'aptitude des jeunes diplômés sont loin d'être atteints (§8.4.1 tiret - 6 et 7). Pour combler cette faille, nous proposons un curriculum d'informatique au lycée, spécialement à la classe de seconde pour initier les élèves à l'informatique et ses applications et une formation aussi s'avère utile pour les enseignants. Mais lesquels ?. Peut-être, allons-nous convertir les enseignants de mathématiques en professeur d'informatique, malgré leur effectif très faible ?. Nouveau programme exige de nouveau cadre juridique et institutionnel, des nouvelles méthodes d'enseignement et d'apprentissage.

Ainsi, L'État doit prendre une décision ferme et osée pour l'enseignement obligatoire de la discipline informatique au collège dans tout le territoire ne serait ce qu'à la classe de seconde pour débiter. Élaborer et promulguer le texte et le cadre juridique

pour l'insertion de cette discipline dans le programme scolaire, ainsi et surtout la détermination de nombre d'heures octroyées à cette discipline. Mais, une mesure d'accompagnement s'impose au niveau de recrutement et formation des enseignants de l'informatique. Dans tout Madagascar, il y a au moins 1211 classes de seconde réparties dans 435 lycées publiques encore fonctionnels. Il faut recruter et former au moins 500 enseignants d'informatique.

Puisque les élèves et les enseignants partent de rien, nous essayons d'établir le programme étape par étape, en fonction de la capacité et la portée des apprentis, i.e. l'utilité d'apprendre le b.a.-ba de l'informatique.

- Tout d'abord partant d'une étape visant à fournir des connaissances initiales en informatique non systémiques à la suite d'un type d'enseignement encyclopédique et quelques notions de bases telles que l'ordinateur, l'algorithme, le codage, le décodage, la variable, . . . L'environnement éducatif comprend de fascicule imprimé et/ou de fichier électronique sous forme de mini-dictionnaire bilingue (français-anglais), confectionné(s) par des enseignants et dont le contenu est ordonné alphabétiquement et illustré par des dessins. On a besoin aussi des matériels informatiques, comme des ordinateurs, des tablettes pour la prise en main des matériels et la consultation du fichier.
- Après enseigner l'informatique dans le cadre de la matière scolaire - Langue et Mathématiques¹¹. C'est un cours d'informatique systématique axé sur divers problèmes et activités étroitement liés aux autres matières scolaires. On a besoin de logiciel éducatif offrant une bonne base pour l'acquisition de compétences en résolution de problèmes et en traitement de l'information.

Des manuels spéciaux et des logiciels pédagogiques doivent être alors créés et/ou acquises pour assurer la pratique d'enseignement. Ce qui est presque inexistant dans les établissements que nous avons passé notre expérience.

L'enseignement de l'informatique offre des possibilités d'intégration naturelle de toutes les disciplines scolaires. La réalisation de cette intégration est l'un des objectifs fondamentaux de la rédaction des manuels d'informatique. La diversité des problèmes permet à chaque élève de trouver son chemin vers l'informatique. À la suite de cet enseignement aux étudiants en informatique d'(e) :

11. On entend par, Langue et Mathématiques une base de l'apprentissage : lire - écrire - calculer)

- acquérir de nouvelles habitudes pour travailler avec des micro-ordinateurs ;
- se familiariser avec certaines des applications de base des ordinateurs dans la société ;
- avoir une idée de leurs applications futures ;
- apprendre à utiliser les ordinateurs comme outil d'expérimentation et de résolution de problèmes relevant des différentes disciplines scolaires ;
- apprendre à formuler les problèmes à résoudre à l'aide de l'ordinateur ;
- acquérir des habitudes pour une **activité créatrice collective**.

L'analyse et la conception du programme scolaire (curriculum) sont des conditions nécessaires pour avoir déjà une guide et orientation globale de ce qu'on doit transmettre aux apprentis et l'élaboration et la confection d'un ensemble de manuels intégrés sur les mathématiques et informatique sont des conditions suffisantes pour bien mener la transmission des connaissances aux apprentis.

L'intégration est considérée utile tant pour les mathématiques que pour l'informatique :

- l'informatique offre des moyens de clarifier et d'étendre les concepts mathématiques étudiés et d'illustrer les possibilités de les appliquer à des situations réelles ;
- l'informatique offre des outils appropriés permettant d'obtenir une grande variété de solutions ;
- travailler avec des ordinateurs peut contribuer à considérer les mathématiques comme une science expérimentale et à réduire le haut niveau de formalisme généralement considéré comme le principal obstacle à la réussite d'un enseignement des mathématiques.

Les objectifs globaux du curriculum en informatique se concentrent sur alphabétisation informatique et les applications des TIC à d'autres disciplines. Les élèves devraient être capables d'utiliser les ordinateurs de manière correcte dans la vie de tous les jours et utiliser des technologies de l'information afin de résoudre des problèmes rencontrés dans d'autres domaines (Bosler et al., 1996, p.9), comme en mathématiques. De ce fait, nous allons diviser en quatre parties l'offre de formation en informatique de la classe de seconde du lycée, dont la quatrième offre est optionnelle selon la plage horaire octroyée à la discipline informatique :

1. Généralités sur les systèmes informatiques ;
2. Quelques applications logiciels ;
3. Algorithmique et programmation ;
4. Réseaux et internet.

8.7.1 Le cours sur les généralités sur les systèmes informatiques

Les systèmes informatiques jouent un rôle important dans notre vie quotidienne, car il est toujours en parallèle avec la puissance des ordinateurs qui ne cessent d'évoluer et notre société est de plus en plus dépendante des systèmes informatiques. Le titre du cours est alors : « **Généralités sur les systèmes informatiques** ». Dans ce cours, l'élève apprend quelques lexiques utilisés en informatique, l'histoire de l'informatique et son évolution, les composantes de base d'un ordinateur, quelques domaines d'application dans la vie socio-économique.

Objectifs du cours : Acquérir la culture informatique, acquérir les notions de base en informatique, identifier et comprendre les différents composants d'un micro-ordinateur et leurs fonctions.

Le cours est divisé en quatre points importants, dont les détails seront présentés dans le tableau 8.17.

1. Histoire de l'informatique ;
2. Définitions et vocabulaire de base ;
3. Structure de l'ordinateur ;
4. Domaine d'application de l'informatique.

Généralités sur les systèmes informatiques	
Objectifs	Acquérir les notions de base en informatique Comprendre la démarche de programmation
Objectifs spécifiques	Contenus
L'élève doit être capable de (d') : Situer l'évolution dans le temps ;	<ul style="list-style-type: none"> • Histoire de l'informatique • Définitions et vocabulaire de base

Mettre en évidence la sécurité des informations ; Montrer l'importance des logiciels et en énumérer les types usuels.	<ul style="list-style-type: none"> • Structure de l'ordinateur • Types de logiciels et domaine d'application
--	--

Tableau 8.17 – Généralités sur les systèmes informatiques

8.7.2 Le cours sur les applications logiciels

L'informatique, plus particulièrement est un outil de communication et de présentation des informations. L'informatique qui est utile dans l'enseignement de disciplines : Langue et Mathématiques. Il est utile d'insérer le cours intitulé « **Applications logiciels** ».

Objectifs du cours: Appréhender l'informatique en tant que discipline par une initiation élémentaire au traitement automatique de l'information à la communication et à l'intégration dans les sociétés en mutation constante.

Ce cours est divisé en deux grandes parties - le logiciel de base : le système d'exploitation et le logiciel d'application : le traitement de texte (*Langue*), le tableur (*Mathématiques*) et la présentation dont les détails sont montrés dans le tableau 8.18.

1. Système d'exploitation ;
2. Traitement de texte ;
3. Tableur ;
4. Présentation.

Applications logiciels	
Objectifs	Exploiter les fonctionnalités d'un système d'exploitation pour gérer des documents numériques ; Exploiter les fonctionnalités d'un tableur pour produire un document mis en forme, contenant des tableaux et éventuellement des objets graphiques ; Exploiter les fonctionnalités d'un tableur pour produire une feuille de calcul ;

	Exploiter les fonctionnalités d'un tableur pour représenter graphiquement les données d'une distribution statistique ; Exploiter les fonctionnalités d'un logiciel de présentation pour pouvoir communiquer le résultat d'un travail
Objectifs spécifiques	Contenus
L'élève doit être capable de (d') :	<ul style="list-style-type: none"> • Système d'exploitation • Traitement de texte - Word • Tableur - Excel • Présentation - Powerpoint

Tableau 8.18 – Applications logiciels

8.7.3 Le cours en algorithmique et programmation

Dans le pays avancés, l'algorithmique est de plus en plus présente dans l'enseignement des mathématiques au lycée. Des questions d'algorithmique sont presque toujours posées au baccalauréat, notamment dans les séries scientifique et économique . Les élèves sont évalués sur leurs capacités à comprendre, compléter, élaborer des algorithmes. Le titre du cours que nous adoptons est : « **Initiation à l'algorithmique et programmation** ». Dans ce cours, l'élève apprend à utiliser les instructions algorithmique de base, écrire un algorithme permettant de résoudre un problème, puis le transcrire en utilisant le langage de programmation Scratch.

Objectifs du cours: Savoir transcrire les différentes étapes de résolution d'un problème sous forme d'algorithme, de façon structurée et indépendante de toute contrainte matérielle ou logicielle.

Les **compétences** requises sont :

- comprendre et analyser un algorithme préexistant ;
- analyser la situation : identifier les données d'entrée, de sortie, le traitement. . . ;
- mettre au point une solution algorithmique : comment écrire un algorithme en langage courant en respectant un code, identifier les boucles, les tests, des opérations d'écriture, d'affichage. . . ;

- valider la solution algorithmique par des jeux d'essais simples.

Le cours est divisé en cinq points importants, dont les détails seront présentés par la suite.

1. Généralités sur l'Algorithmique (Tableau 8.19),
2. Les opérations de lecture et d'écriture (Tableau 8.20),
3. Structures de contrôle conditionnel (Tableau 8.21),
4. Les structures itératives (Tableau 8.22),
5. Les tableaux (Tableau 8.23).

8.7.3.1 Généralités sur l'Algorithmique

Généralités sur l'Algorithmique	
Objectifs	Connaître le vocabulaire de base en programmation Comprendre la démarche de programmation
Objectifs spécifiques	Contenus
L'élève doit être capable de (d') : savoir ce qu'est l'ordinateur en tant qu'outil qui exécute le programme	<ul style="list-style-type: none"> • Ordinateur et programmation • Définitions et unités de mesure • Algorithmes • Codage • Les variables et les constantes • Opérandes et operateurs

Tableau 8.19 – Généralités sur l'Algorithmique

8.7.3.2 Les opérations de lecture et d'écriture

Les opérations de lecture et d'écriture	
Objectif	Manipuler correctement les opérations de lecture et d'écriture

Objectifs spécifiques	Contenus
L'élève doit être capable de (d') : écrire une formule permettant un calcul écrire un programme calculant et donnant la valeur d'une fonction : ainsi que les instructions d'entrées et sorties nécessaires au traitement	<ul style="list-style-type: none"> • Syntaxe générale • Entrées-sorties standards : Écriture en format libre • Les formats • Les fichiers

Tableau 8.20 – Les opérations de lecture et d'écriture

8.7.3.3 Structures de contrôle conditionnel

Structures de contrôle conditionnel	
Objectif	Construire des algorithmes comportant des traitements conditionnels.
Objectifs spécifiques	Contenus
L'élève doit être capable de (d') : une instruction conditionnelle	<ul style="list-style-type: none"> • Expression logique • Évaluation d'une expression logique • Tableaux d'évaluations • Test alternatif simple • Test alternatif double • Tests imbriqués

Tableau 8.21 – Structures de contrôle conditionnel

8.7.3.4 Les structures itératives

Les structures itératives

Objectif	Construire des algorithmes comportant des traitements itératifs
Objectifs spécifiques	Contenus
L'élève doit être capable de (d') : programmer un calcul itératif, le nombre d'itérations étant donné programmer un calcul itératif, avec une fin de boucle conditionnelle	<ul style="list-style-type: none"> • La boucle « POUR » • La boucle « TANT QUE » • La boucle « REPETER ... JUSQU'A ... »

Tableau 8.22 – Les structures itératives

8.7.3.5 Les tableaux

Les tableaux	
Objectif	Développer des algorithmes de traitement de tableaux
Objectifs spécifiques	Contenus
L'élève doit être capable de (d') : programmer à l'aide des algorithmes de tris préétablis	<ul style="list-style-type: none"> • Déclaration des tableaux • Terminologie des tableaux • Manipulation d'un tableau • Tri d'un tableau • Tri par sélection

Tableau 8.23 – Les tableaux

8.8 Conclusion

Dans ce chapitre, nous avons mené deux études sur la formation en informatique dans le modèle anglo-saxon et la formation en informatique à l'ENSET qui est un modèle francophone. La comparaison de ces deux formations, sur les thèmes offerts et les capacités d'aptitudes des étudiants poursuivis le parcours informatique dans chaque modèle à l'université, nous a permis de détecter les manques dans notre modèle et a proposé un curriculum en informatique avant premier depuis la classe de seconde pour combler la vide de la formation en informatique dans le cycle supérieur.

Chapitre 9

Utilisation de Scratch dans l'apprentissage de l'algorithmique en classe de Seconde à Madagascar

*“ L’Homme de métier est un
éternel apprentis. ”*

Hery Frédéric Rakotomalala

Sommaire

9.1	Introduction	322
9.2	Choix et utilisation de Scratch	322
9.2.1	L'espace de travail et environnement Scratch	323
9.2.2	La programmation en Scratch	324
9.2.3	Liste des blocs utilisés dans Scratch	325
9.3	Expérimentations	325
9.4	Le déroulement de séances en classe	326
9.4.1	La première séance : Prise de contact	326
9.4.2	La deuxième séance : Activité de découverte de l'algorithme et de la programmation	329
9.4.3	La troisième séance : Activité de programmation - Scratch	340
9.4.4	La quatrième séance : Évaluation finale	348
9.4.5	Analyses des résultats d'évaluation	349
9.5	Conclusion	355

9.1 Introduction

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au collège, les élèves ont rencontré des algorithmes mathématiques (algorithmes opératoires, algorithme des différences, algorithme d'Euclide, algorithmes de construction en géométrie). Ce qui est proposé dans le programme est une formalisation en langage naturel propre à donner lieu à traduction à l'aide d'un logiciel Scratch. Il s'agit de familiariser les élèves avec les grands principes d'organisation d'un algorithme : gestion des entrées-sorties, affectation d'une valeur et mise en forme d'un calcul. Ce chapitre est consacré à préparer un programme afin de former les élèves à la démarche scientifique sous toutes ses formes pour les rendre capables de modéliser et s'engager dans une activité de recherche, de conduire un raisonnement, une démonstration, de pratiquer une activité expérimentale ou algorithmique, de faire une analyse critique d'un résultat, d'une démarche, de pratiquer une lecture active de l'information (critique, traitement), d'utiliser les outils logiciels (Scratch) adaptés à la résolution d'un problème et de communiquer à l'écrit et à l'oral.

9.2 Choix et utilisation de Scratch

Scratch est un langage de programmation, gratuit et graphique que les élèves s'intéressent à son utilisation faute de ne pas beaucoup écrire des textes mais déplacer en général des blocs d'instructions. Scratch est développé par un groupe de chercheurs du MIT86 (Massachusetts Institute Technology) Media Lab, et a été conçu pour l'enseignement et l'apprentissage de l'algorithmique pour les élèves de tout âge et de toutes les classes. Scratch offre un environnement d'édition et d'exécution des programmes. Il s'agit d'un logiciel gratuit et disponible sur toutes les plates-formes usuelles, choisi pour sa simplicité, sa fiabilité et sa robustesse dans la mise en œuvre. Il permet de travailler tous les concepts figurant au programme, en particulier la programmation événementielle et la gestion de scripts s'exécutant en parallèle. Scratch permet aux utilisateurs des activités dans tous les domaines sachant la création d'histoires interactives, la construction des dessins animés, la composition musicale, la simulation numérique. Il permet aussi l'apprentissage collaboratif, l'acquisition des notions mathématiques et informatiques et particulièrement des notions de pro-

grammation ([Muratet, 2012](#)).

En créant des projets Scratch, les élèves peuvent en même temps apprendre des notions importantes de mathématiques, de calcul et de programmation tout en travaillant de manière collaborative leur créativité et leur raisonnement. Scratch est utilisé comme support à l'enseignement de l'algorithmique en seconde générale et technologique au lycée ([EduSCOL, 2009](#)).

Les projets Scratch sont composés d'objets appelés lutins . Ces objets peuvent conserver le nom d'objet, on l'appelle également objet-lutin. Un objet-lutin peut ressembler à une personne ou à une autre chose. On peut donner des instructions à un objet-lutin, lui dire de se déplacer ou de jouer de la musique ou de réagir avec d'autres lutins. Pour dire à un objet-lutin ce qu'il doit faire, on assemble des blocs de programmation pour constituer des piles, que l'on appellera script lorsque la pile sera surmontée d'un bloc de contrôle de type chapeau. Lorsque on clique sur un script, Scratch exécute les blocs de programmation à partir du haut jusqu'en bas, i.e l'exécution se fait séquentiellement.

!!...Je suis un lutin...!!



Scratch est un outil logiciel qui nous permet d'adopter une démarche algorithmique pour faire face à des situations simple, à cause d'une structure algorithmique séquentielle dans la résolution de problème à situation simple ; la possibilité de donner l'instruction au processeur pour la résolution de problème à situation conditionnelle et de réaliser un programme correct en adoptant une démarche algorithmique convenable.

Dans la mesure du possible, les problèmes posés s'inspirent de situations liées à la vie courante ou à d'autres disciplines. Ils doivent pouvoir s'exprimer de façon simple et concise et laisser dans leur résolution une place à l'autonomie et à l'initiative des élèves. Au niveau d'une classe de seconde de détermination, les solutions attendues sont aussi en général simples et courtes.

9.2.1 L'espace de travail et environnement Scratch

Après lancement du programme, la fenêtre principale du Scratch est montrée par la Figure 9.1.

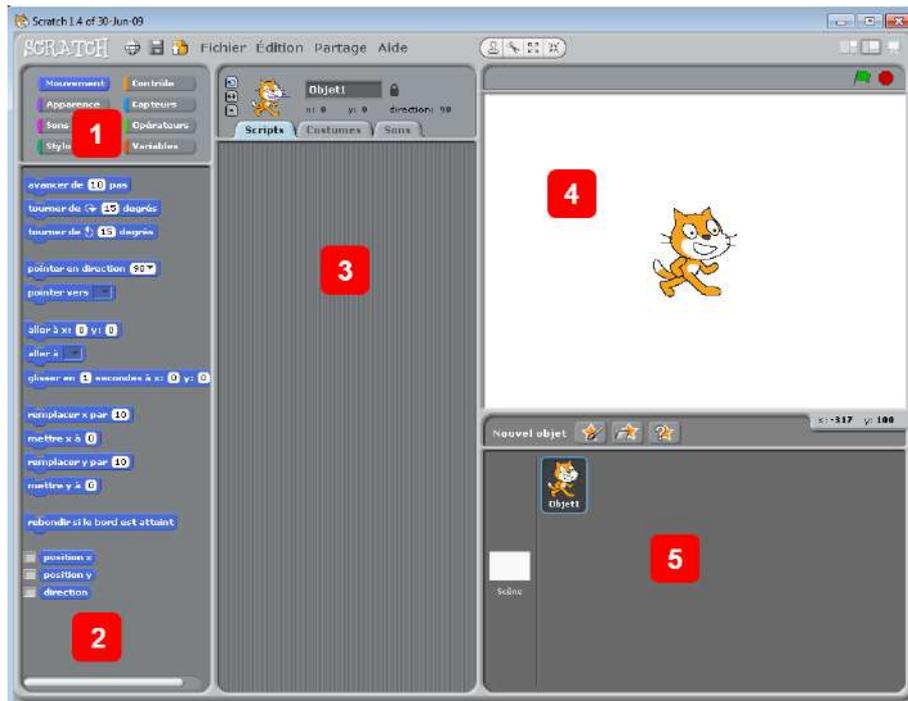


Figure 9.1 – Fenêtre principale Scratch

La fenêtre principale est divisée en cinq zone de travail

1. Zone de menu se trouvant les instructions à faire glisser dans la zone de script ;
2. Zone de sous-menu dépendant de la zone de menu affichant les instructions à faire glisser dans la zone de script ;
3. Zone de script, là où on assemble les instructions du programme ;
4. Zone de scène ;
5. Zone de gestion et de création des lutins et/ou des arrière-plans.

9.2.2 La programmation en Scratch

La programmation se fait en déplaçant des instructions. L'algorithme démarre en cliquant sur la première instruction représentée par le bloc contenant un drapeau vert dans le menu contrôle :



9.2.3 Liste des blocs utilisés dans Scratch

Les blocs représentés dans le tableau 9.1 sont les plus courants et possèdent une encoche d'emboîtement, femelle en haut et mâle en bas.

Blocs	Désignation	Code
	Mouvement	mvt
	Apparence	app
	Sons	son
	Stylo	stylo
	Contrôle	ctrl
	Capteurs	capt
	Opérateurs	ope
	Variables	data

Tableau 9.1 – Les commandes dans la zone de menu

9.3 Expérimentations

L'expérimentation s'est faite en deux séances en classe entière (38 élèves de chaque) de seconde en salle de classe et en salle informatique au lycée St Jean de Diégo Suarez. Ces élèves allaient se familiariser le logiciel de programmation Scratch et n'ont jamais suivi un enseignement de l'algorithmique. Chaque séance dure deux heures par semaine. Remarquons que notre première séance se déroulait au début du premier trimestre de l'année scolaire. Les restes de la séance se déroulaient au troisième trimestre. La raison est simple, il fallait que les élèves se familiarisaient avec les machines, travaillaient et faisaient de traitement de texte, tableur, . . . i.e qu'il finissent les deux grandes parties du curriculum informatique en classe de seconde (Cf. §8.7.1 et §8.7.2) avant d'entamer le cours algorithmique et programmation (Cf. §8.7.3).

9.4 Le déroulement de séances en classe

9.4.1 La première séance : Prise de contact

Cette séance est consacrée seulement de faire un état de lieu de chaque classe concernant leur connaissance sur l'informatique et la culture informatique en général. Il n'y avait pas de préparation très pointu, car c'est seulement une séance de contact. Mais nous avons quand même préparé une série de questions sur la langue et/ou le dialecte que pratique et comprennent chaque élève et l'enseignant (le même enseignant dans les deux classes). Remarquons que la langue d'enseignement est le français. Les manuels que nous choisissons pour le cours et les pratiques sont en français.

Question 1 : Lesquels de ces langues et dialectes maîtrisez-vous ?

1. français
2. anglais
3. malagasy officiel
4. dialecte antakarana
5. dialecte tsimihety
6. dialecte antemoro

Maîtriser veut dire : lire, écrire, parler et comprendre.

Réponse 1 : L'enseignant ne maîtrise que le dialecte tsimihety.

Pour les élèves, nous montrons dans le tableau 9.2 ci-dessous :

Langue/dialecte	seconde I	seconde II
français	0	0
anglais	0	0
malagasy officiel	17	10
antakarana	10	18
tsimihety	3	3
antemoro	8	7
Effectif de chaque classe	38	38

Tableau 9.2 – Langue et dialecte pratiqués dans les deux classes

La comptage s'est fait à main levée, et nous remarquons qu'aucun élève maîtrise des langues étrangères, généralement une manque de compréhension et avoir une

difficulté à l'oral. Mais cela nous a déjà aidé à choisir la langue d'enseignement par la suite.

Question 2 : C'est une question d'ordre général sur la matière informatique.

1. Vous faites quelle matière là, en ce moment ?
2. Travaillez vous déjà avec un ordinateur avant que vous rentriez en classe de seconde ?
3. Pouvez vous citer les périphériques qui relie avec l'ordinateur ?
4. Quels sont les logiciels/applications que nous utilisons à l'école ?
5. Est ce que l'informatique nous aide dans la vie quotidienne ?
6. Entendez-vous parler d'algorithme et connaissez-vous ce que c'est ?.

Nous avons utilisé le crayon-papier pour avoir les réponses. Après avoir traité les données obtenues par les 76 élèves des deux classes en binaires, nous avons obtenu les résultats suivants :

	occurrence	moyenne	ecart-type
Informatique oui	75	0.9868421052631579	0.11470786693528089
Informatique non	1	0.013157894736842105	0.11470786693528089
Ordinateur oui	51	0.6710526315789473	0.47295265146345866
Ordinateur non	25	0.32894736842105265	0.47295265146345866
Périphérique pas du tout	17	0.2236842105263158	0.4194817186489797
Périphérique qq	57	0.75	0.43588989435406733
Périphérique connaît	2	0.02631578947368421	0.16113631583445662
logiciel pas du tout	20	0.2631578947368421	0.44327331080291255
logiciel qq	49	0.6447368421052632	0.48177304112817976
logiciel connaît	7	0.09210526315789473	0.2910959328215754
aide info ne sait pas	9	0.11842105263157894	0.3252529380101132
aide info sait	67	0.881578947368421	0.3252529380101132
algo ne sait pas	73	0.9605263157894737	0.19601288894248498
algo sait	3	0.039473684210526314	0.19601288894248498

Tableau 9.3 – Caractéristiques des réponses des élèves

Le tableau 9.3 nous montre que les deux classes sont homogènes pour la première question. Les réponses obtenues des deux classes sur la deuxième question sont un peu disparates, car 32.89% des élèves ne travaillaient pas avec un ordinateur avant qu'ils viennent au lycée, cela implique que 22.36% n'arrivent pas à citer des périphériques d'un ordinateur et 26.31% ne connaissent pas les logiciels qu'ils utilisent en classe. Pour la cinquième question, les élèves ont répondu instinctivement *oui* sans connaître vraiment l'aide que l'informatique apporte dans la vie quotidienne et le pire que presque tout le monde ne connaît ce qu'est un algorithme. La tâche est

alors lourde.

H	Niveau	Cohesion inter-classe $coh_{supp(n)M_{GK}}(\underline{C})$	Implication inter-classe $\psi(\underline{C}, \underline{C}')$	Niveau significatif $s(\Omega, H)$	Nœud significatif $v(\Omega, H)$
1	(aide info sait=>Informatique oui)	1.0	1.0	0.00951705	0.00951705
2	(Informatique non=>logiciel pas du tout)	1.0	1.0	0.00005258	-0.00946447
3	(logiciel connait=>Ordinateur oui)	1.0	1.0	-0.00306679	-0.00311938
4	(logiciel qqs=>algo ne sait pas)	1.0	1.0	-0.00465262	-0.00158583
5	(Ordinateur non=>Peripherique qqs)	0.048	0.52	-0.00558260	-0.00092998

Tableau 9.4 – Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau pour la question 2 §9.4.1

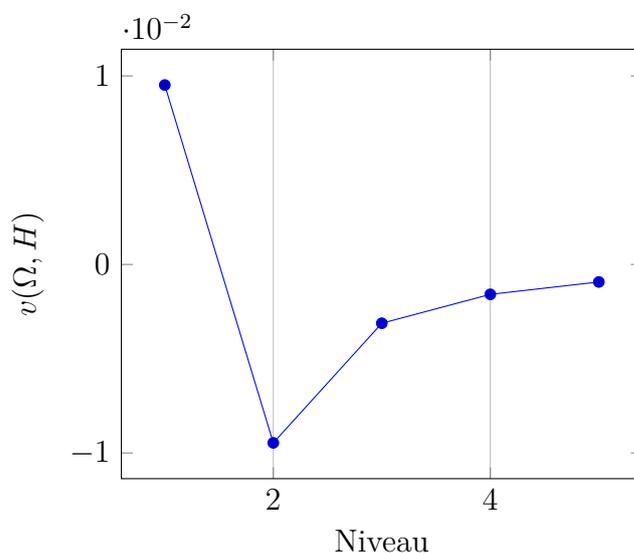


Figure 9.2 – Graphe représentant les nœuds significatifs des 5 classes de question 2 §9.4.1

Les nœuds au niveau $H = \{1, 2, 5\}$ sont significatifs.

Quelques interprétations

Tous ceux qui disaient que l'informatique nous aide dans la vie quotidienne, savaient la matière qu'on les enseigne en classe, c'est « l'informatique ». Ceux qui n'arrivaient pas à citer aucun des cinq logiciels qu'on traite en classe (logiciel de base : windows 7, applications : word, excel, powerpoint et scratch) ne savaient même pas que l'informatique est parmi les matières qu'on enseigne en classe et ils répondaient « je ne sais pas ». La plupart des élèves qui n'arrivaient pas à citer que quelques périphériques ne travaillaient pas avec un ordinateur avant qu'ils rentrent en classe de seconde.

9.4.2 La deuxième séance : Activité de découverte de l'algorithme et de la programmation

Cette séance vise à s'entraîner les élèves d'identifier les problèmes posés, à organiser leur raisonnement de résolution de ces problèmes en utilisant la langue naturelle, en leur montrant que nous faisons des centaines ou même des milliers d'algorithmes dans une journée. C'est une phase de découverte de l'algorithme. Elle vise aussi à initier les élèves au concept de la programmation et à la programmation.

Raisonnement et langage mathématiques

Le développement de l'argumentation et l'entraînement à la logique font partie intégrante des exigences des classes de lycée. à l'issue de la seconde, l'élève devra avoir acquis une expérience lui permettant de commencer à distinguer les principes de la logique mathématique de ceux de la logique du langage courant et, par exemple, à distinguer implication mathématique et causalité.

Nous avons proposé et expliqué, deux exemples ([Meurist, 2014](#), p.214) qu'on voit dans la vie c'est : *réchauffer un plat au micro-onde* et *démarrer une voiture*.

Pour réchauffer un plat au micro-ondes, nous suivons les étapes suivantes :

Algorithme 1

1. Mettre le plat dans le micro-ondes,
2. Fermer la porte du micro-ondes,
3. Choisir la puissance et le temps de chauffe,
4. Mettre en route le micro-ondes,
5. Attendre que le temps se soit écoulé,
6. Ouvrir la porte du micro-ondes,
7. Prendre son plat.

Pour démarrer une voiture (fonctionnant avec une clé), nous procédons comme suit :

Algorithme 2

1. Insérer la clé dans le contacteur à clé,

2. S'assurer que le levier de vitesses se trouve en position neutre (point mort),
3. Mettre le contact,
4. Si le moteur part en moins de six secondes, relâcher la clé,
5. Si le moteur ne part pas en six secondes :
 - relâcher la clé,
 - attendre 10 secondes,
 - répéter les étapes 3, 4 et 5, mais pas plus de 5 fois en tout.
6. Si le moteur ne part pas, appeler le garage.

A chaque étape de l'algorithme, les élèves donnaient d'abord leur avis avant d'écrire la réponse et en parallèle, le professeur montre déjà l'élément de l'algorigramme¹: si c'est un calcul, on met dans un rectangle, si c'est une condition dans un losange. . .

Là, les élèves ont pu déjà savoir ce qu'est un problème ? On a deux problèmes ici : réchauffer un plat et démarrer une voiture. Maintenant comment les résoudre ? La résolution s'est faite par des séquences d'instructions (tous les verbes qu'on utilise sont à l'impératif) : *Insérer, s'assurer, mettre, relâcher, attendre, répéter, appeler. . .* D'où la définition de l'algorithme : *c'est une liste ordonnée et logique d'instructions permettant de résoudre un problème.*

Après une longue échange avec les élèves par rapport aux deux exemples d'algorithmes précédents, nous avons pris deux activités dans ([Sésamath, 2016](#), p.340) ci-dessous.

Activité 1 : Le robot et moi, comment réaliser une action simple

1. Comment fais-tu pour traverser la route ? Peut-on programmer un robot pour qu'il en fasse autant ?
2. Comment fais-tu pour sortir de la salle de classe ? Peut-on programmer un robot pour qu'il en fasse autant ?

Algorithme - Activité 1 - 1 : traverser la route

1. aller vers le trottoir,
2. chercher le passage zébré,
3. regarder à gauche et à droite,

1. L'algorigramme est la représentation graphique de l'algorithme

4. s'il n'y a pas de voitures de deux côtés, traverser la route.
5. sinon attendre et reprendre les instructions 2 et 3,

Nous avons retenu 15 mots clés dans cet algorithme : aller/marcher, trottoir, chercher, passage zébré, regarder, à gauche et à droite, s'il n'y a pas, voitures, deux côtés, traverser, route, sinon, attendre, reprendre, instructions.

Après vérification des copies, dans les deux classes, le nombre des mots clés écrits par les élèves : $2 \leq \text{mots clés} \leq 7$.

Algorithme - Activité 1 - 2 : sortir de la salle de classe

1. Attendre la sonnerie,
2. Ranger ses affaires,
3. Faire la prière,
4. Attendre la permission du professeur,
5. Dire au revoir au prof et à ses amis,
6. Marcher vers la porte par rapport à ses rangés,
7. sortir un par un.

Nous avons retenu aussi 15 mots clés dans cet algorithme : attendre, sonnerie, ranger, affaire, faire, prière, permission du prof, dire, au revoir au prof, amis, marcher, porte, rangés, sortir, un par un.

Après vérification des copies, dans les deux classes, le nombre des mots clés écrits par les élèves : $5 \leq \text{mots clés} \leq 10$.

Voici quelques illustrations des résultats (figures 9.3-9.4-9.5 et 9.6).

Activité ①

1- Je ~~est~~ regarde de gauche et droite, ensuite je traverse la route en marchant lorsqu'il n'y a pas de voiture. Je marche en mettant un pied devant l'autre.

⇒ On peut programmer un robot pour traverser la route si il suit toutes les instructions.

2- pour sortir de la salle de classe je marche en passant par la porte d'entrée.

⇒ Oui, on peut programmer un robot si on le fait suivre chaque instruction à la lettre et par étapes.

Figure 9.3 – élève 1 - activité 1

I - Pour traverser la route

En arrivant sur la route, je me stoppe.

- 1 sur le trottoir
- 2 Je regarde en gauche et en droite
- 3 si n'a pas de voiture ou etc
- 6 en traverse sur la route
- 7* Oui on programme un robot

II - Pour sortir de la salle de classe

- 1 En entendant la sonnerie, nous ont fait
- 2 la prière, et on sort en rentrant dans nos maison, et dire au revoir au prof
- 6* Oui je range mes affaires, marche vers la porte
- 8 ouvrir la porte et on sort de la classe
- 9

Figure 9.4 – élève 2 - activité 1

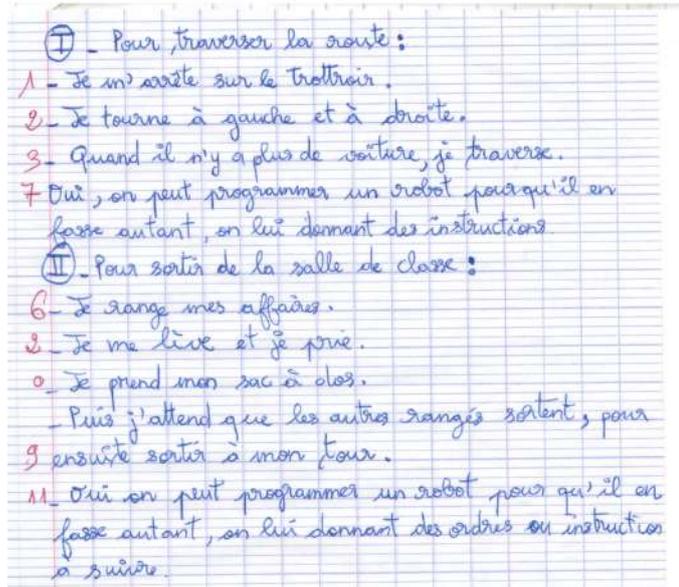


Figure 9.5 – élève 3 - activité 1

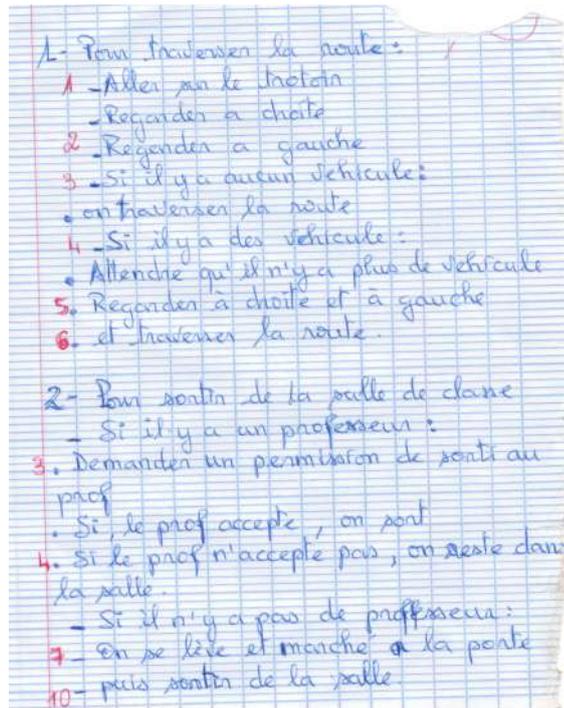


Figure 9.6 – élève 4 - activité 1

Après cette activité un peu ennuyeuse pour les élèves, nous avons expliqué sur la représentation graphique de l'algorithme (algorithme) en utilisant des symboles normalisés représentés dans le figure 9.7 ci-dessous.

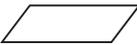
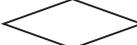
						
Début, fin, interruption	Renvoi	Entrée-Sortie	Symbole général	Branchement	Sous-programme	Commentaire
Début, fin ou interruption d'un algorithme	Symbole utilisé deux fois pour assurer la continuité lorsqu'une partie de ligne de liaison n'est pas représentée	Mise à disposition d'une information à traiter ou enregistrement d'une information traitée	Opération ou groupe d'opérations sur des données, instructions pour laquelle il n'existe aucun symbole normalisé	Exploitation de conditions variables impliquant un choix parmi plusieurs	Portion de programme considérée comme une simple opération	Symbole utilisé pour donner des indications sur les opérations effectuées

Figure 9.7 – Symboles utilisés dans l'algorithme

Définition 9.1. *L'algorithme est une séquence d'instructions finies pour résoudre un problème.*

Puisque les instructions sont finies, il y a le début et la fin des instructions (figure 9.8). La séquence d'instructions est un ensemble des règles ordonnées pour accomplir des tâches bien déterminées (figure 9.9).



Figure 9.8 – Algorithme

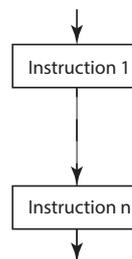


Figure 9.9 – Séquence d'instructions

Nous avons pris comme exemples d'application les deux activités 1-1 et 1-2. Trois symboles ont été utilisés pour ces deux activités : début-fin, symbole général et branchement. Après, nous avons passé aux explications des autres symboles : entrée-sortie, renvoi, commentaire et enfin le sous-programme. Mais nous avons insisté sur l'utilisation de entrée-sortie et avons commencé à apprendre la notion de programmation, plus précisément un programme comparé avec une recette de cuisine (Cf. figure 9.10).

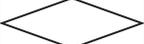
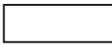
Symbole	Recette de cuisine	Programme
	Nom de la recette : Tarte au pomme	Nom du programme : résolution d'une équation du second degré dans un espace réel
	Ingrédients : Pomme : 3, sucre, farine :150g, eau, oeufs : 5, beurre : 25g...	Déclaration des variables et constantes a=1, b=-2, c=1 : constantes x, delta, solution1, solution 2 : variables
      	Mode d'emploi: Eplucher les pommes; Couper-les en rondelles; Préparation de la pâte Si la pâte n'est pas encore...alors...sinon... <i>C'était la phase de préparation de la recette</i> Présenter sur un plateau	début delta= $b^2-4*a*c$; Programme de calcul de la racine carré sqrt si delta=0 alors solution1= $-b/2*a$; sinon si delta>0 alors solution1= $-b+sqrt(delta)/2*a$; solution2= $-b-sqrt(delta)/2-a$; sinon «pas de solution dans R» print(solution1,solution2) fin

Figure 9.10 – Recette de cuisine et programme

9.4.2.1 Utilisation d'outil logiciel Scratch

L'utilisation de logiciel Scratch, développe la possibilité d'expérimenter, ouvre largement la dialectique entre l'observation et la démonstration et change profondément la nature de l'enseignement.

L'utilisation régulière de cet outil peut intervenir selon trois modalités :

- par le professeur, en classe, avec un dispositif de visualisation collective adapté ;
- par les élèves, sous forme de travaux pratiques de mathématiques ;
- dans le cadre du travail personnel des élèves hors du temps de classe.

Les opérations élémentaires relatives à la résolution d'un problème peuvent être organisées selon quatre structures :

- structures linéaires ;
- structures alternatives ;
- structures de choix ;
- structures itératives.

9.4.2.2 Structure linéaire

C'est une suite d'actions à exécuter successivement dans l'ordre de leur énoncé. Généralement, on utilise ici la commande d'affectation de la forme :

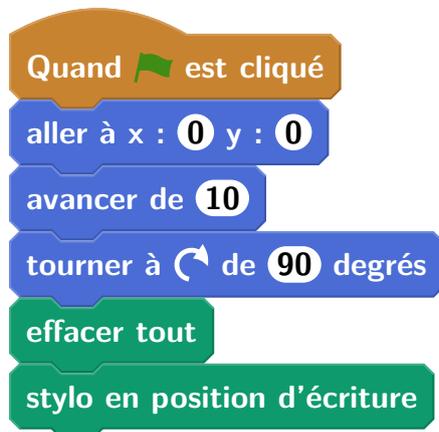
variable ← *expression*

L'expression est une suite d'opérations sur des constantes ou des variables déjà déclarées. Exemple de liste des commandes simples montré par le professeur.

Pour que tout le monde démarre dans la même position à chaque fois que le drapeau vert est cliqué, commence toujours par les blocs suivants avant d'ajouter ses propres instructions :

- Quand le drapeau vert est cliqué
- Aller à $x = 0, y = 0$
- S'orienter à 90° (vers la droite)
- Effacer tout
- Stylo en position d'écriture

Positionne ces blocs, puis fais avancer Scratch !



9.4.2.3 Liste des commandes spéciales

Événement



Tourner

tourner à ↻ de **-145** degrés

Application : écrire le programme suivant en Scratch :

- Faire avancer Scratch de 50 pas ;
- Faire une pause d'une seconde ;
- Faire encore avancer Scratch de 50 pas, puis une pause d'une seconde ;
- Faire avancer Scratch de 50 pas une dernière fois ;
- Effacer tout ;
- Mettre le stylo à la position initiale.



Activité 2 : Recettes et algorithmes

L'objectif de cette activité est de lire, d'écrire et de comprendre le déroulement d'algorithme. Un robot d'aide à la personne sait faire des recettes de cuisine s'il est bien programmé. La recette des crêpes a beaucoup de succès : Recette pour 15 crêpes

Ingrédients

- 300 g de farine
- 3 oeufs entiers
- 3 cuillères à soupe de sucre
- 2 cuillères à soupe d'huile
- 50 g de beurre fondu
- lait (environ 30 cl), à doser jusqu'à jusqu'à obtenir la consistance souhaitée

Préparation de la recette Mettre la farine dans une terrine et former un puits. Mettre les œufs entiers, le sucre, l'huile et le beurre. Mélanger délicatement avec un fouet en ajoutant au fur et à mesure le lait. La pâte ainsi obtenue doit avoir la consistance d'un liquide légèrement épais. Faire chauffer une poêle anti-adhésive et y déposer quelques gouttes d'huile. Faire cuire les crêpes à feu vif.

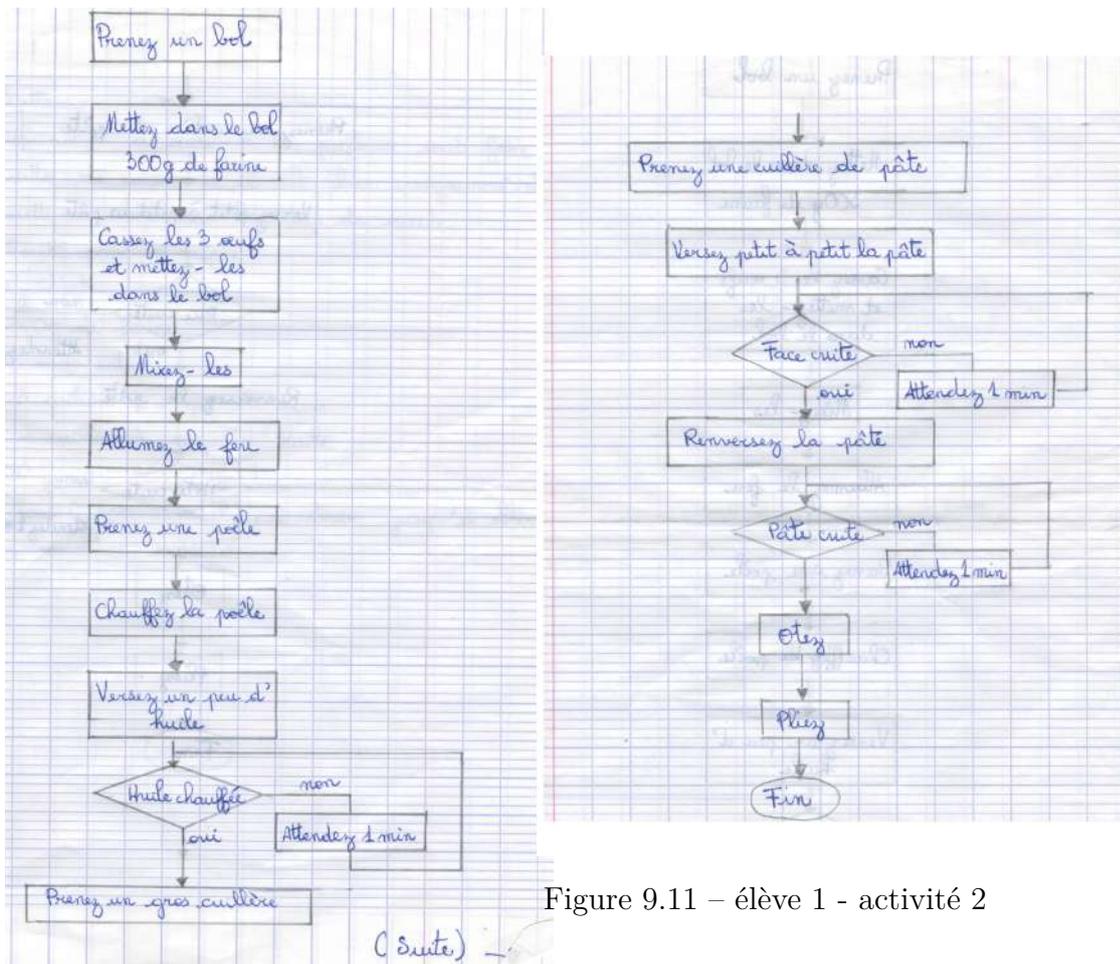


Figure 9.11 – élève 1 - activité 2

Réponses

1. Non, la préparation n'est pas simple pour un robot car il faut préciser certains points comme :

- mettre de la farine dans un bol
- ajouter des œufs et battre une pâte.

2. Le nombre minimum de œufs possible est : 2 œufs.

3. Algorithme :

1. On commence, on prend de la farine 300g de farine dans un bol.
2. On casse 2 œufs et on mélange le blanc et le jaune d'œuf dans la farine.
3. On ajoute la pâte avec une spatule.
4. On ajoute 2 millions de sucre et 2 millions de beurre dans la pâte.
5. On ajoute 2 millions de sucre et 2 millions de beurre dans la pâte.
6. On ajoute 2 millions de sucre et 2 millions de beurre dans la pâte.
7. On met sur le feu la pâte pour la chauffer.
8. On met dans la pâte un morceau de beurre.
9. On verse une petite quantité de sucre et de beurre.
10. On fait cuire 15 minutes ou 1 minute 15 en fonction de la température.
11. Après 1 minute ou 1 minute 30, on inverse la pâte pour qu'elle soit bien cuite.
12. On prend une spatule pour mettre les 15 minutes.

Figure 9.12 – élève 2 - activité 2

1. La recette fait 15 œufs

1. Non, la préparation n'est pas simple pour un robot car il faut préciser certains points :
Il faut faire un algorithme.

2. Le nombre minimum de œufs possible est 5

3. Algorithme d'une préparation de 15 œufs à l'aide d'un robot.

1. Prendre un grand bol.
2. Verser 300g de farine dans ce bol.
3. Casser 2 œufs dans le bol contenant la farine.
4. Mélanger les ingrédients pendant 1 min à la main droite. C'est un mélange mais le contenu C est gauche.
5. Prendre environ 30 cl de lait et verser dans le bol.
6. Mélanger le mélange de farine, les œufs et le lait jusqu'à obtenir une consistance pâteuse (ex: pâte à pain).
7. Ajouter 2 millions de sucre et 2 millions de beurre dans le bol pour mélanger les ingrédients.
8. Faire cuire le mélange. Allumer le robot à 150°C.
9. Prendre une spatule et la mettre sur le robot en attendant qu'elle chauffe.
10. Quand la spatule se chauffe on ajoute 5g de beurre dans la pâte jusqu'à ce que on finisse les 15 minutes.
11. Prendre une spatule.
12. Prendre la pâte de œufs avec une spatule et la verser dans la pâte contenant le beurre fondu et ainsi de suite.
13. Quand C est sur le robot dans un plateau bon appétit!

Figure 9.13 – élève 3 - activité 2

Nous avons remarqué qu'avec cette activité 1 - 2, les élèves ont pu déjà faire des représentations graphiques de l'algorithmme (figure 9.11) et d'écrire quelques instructions (figures 9.12-9.13).

9.4.3 La troisième séance : Activité de programmation - Scratch

Cette fois-ci, nous avons travaillé dans la salle informatique, dotée d'une dizaine d'ordinateurs. La classe a été divisée en groupe de trois élèves, soit 12 groupes par classe. Remarquons que nous avons 2 classes de seconde au lycée St Jean de Diégo Suarez. Le cours de programmation consiste à étudier les autres structures telles que les structures alternatives, les structures de choix et les structures d'itératives (répétitions). Cette séance était la plus dure, mais amusante, car le cours est trop long, mais les groupes d'élèves travaillaient en parallèle avec les machines.

9.4.3.1 Structure alternative réduite - Si...Alors

La structure alternative réduite marque par le fait que seule la situation correspondant à la validation de la condition entraîne l'exécution du traitement, l'autre situation conduit à la sortie de la structure.



9.4.3.2 Structure alternative complète - Si...Alors...Sinon

Dans cette structure (Cf. figure 9.14), l'exécution d'un des deux traitements distinctes ne dépend que du résultat d'un test effectué sur la condition qui peut être une variable ou un événement ;

- si la condition est vérifiée, seul le premier traitement est exécuté ;
- si la condition n'est pas vérifiée, seul le second traitement est effectué.

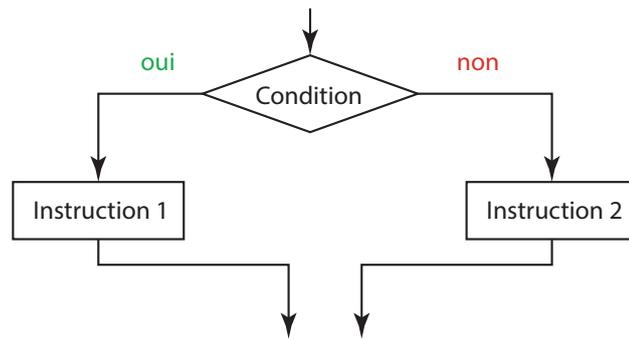


Figure 9.14 – Structure alternative complète



9.4.3.3 Structure de choix ou structure alternative emboîtée

La structure de choix permet, en fonction de plusieurs conditions de type booléen, d'effectuer des actions différents suivant les valeurs que peut prendre une même variable (Cf. figure 9.15).

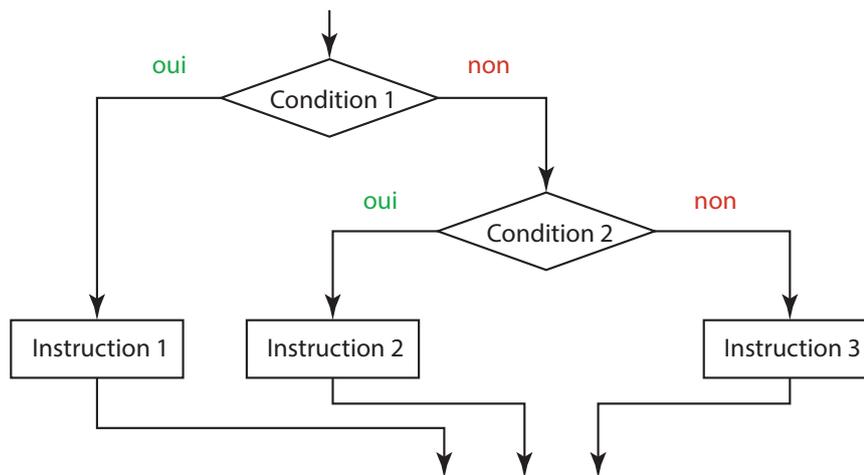
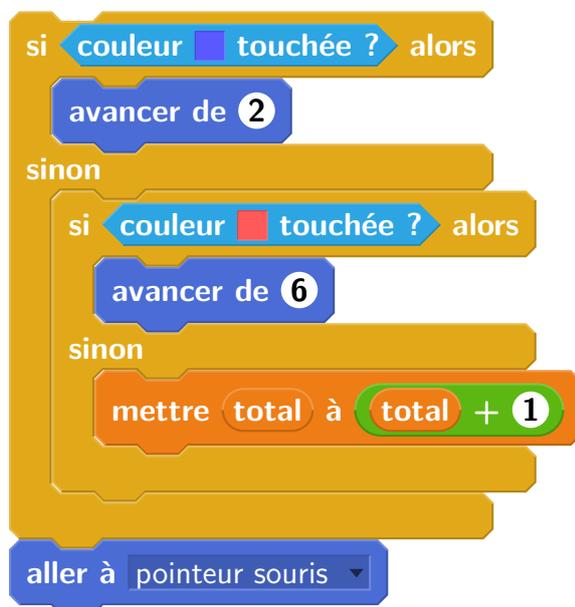


Figure 9.15 – Structure de choix



9.4.3.4 Structure itérative

Par itération on désigne toute répétition de l'exécution d'un traitement. Trois types de structures itératives sont à distinguer :

- la structure REPETER... TANT QUE..., figure 9.16;
- la structure REPETER... JUSQU'A..., figure 9.17;
- la structure POUR... A... REPETER..., figure 9.18.

Dans les deux premier cas, le nombre de répétitions n'est pas connu à l'avance et dépend d'un ou plusieurs événements extérieurs.

Dans le dernier cas, le nombre de répétitions est connu à l'avance, il est consigné.

Boucles

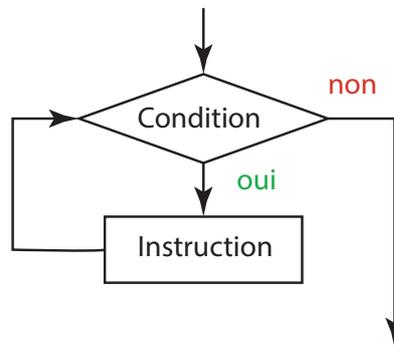


Figure 9.16 – Boucle tant que... faire

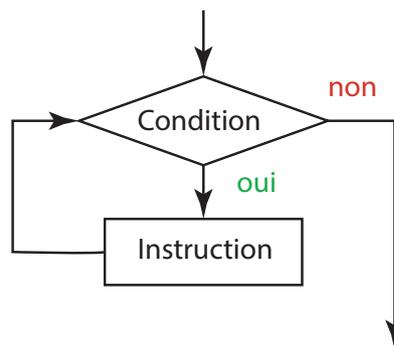


Figure 9.17 – Boucle répéter... jusqu'à

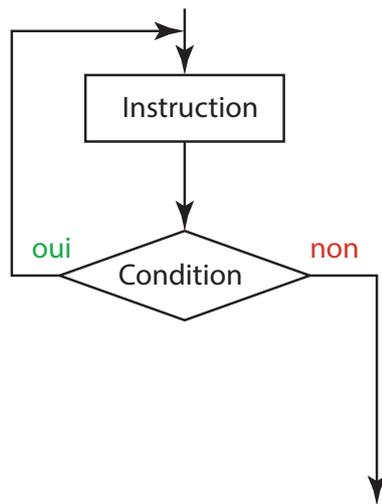


Figure 9.18 – Boucle répéter... jusqu'à

```
définir trace_carré coté_carré
répéter 4 fois
  avancer de coté_carré
  tourner de 90 degrés
stylo en position d'écriture
répéter indéfiniment
  s'orienter à nombre aléatoire entre 0 et 359
  avancer de nombre aléatoire entre 1 et 10
  si bord touché ?
    aller à x : 0 y : 0
```

9.4.3.5 Boucle imbriquée

C'est une boucle complètement incluse dans une autre, de sorte que cette dernière ne peut poursuivre son itération qu'après la sortie de la première.

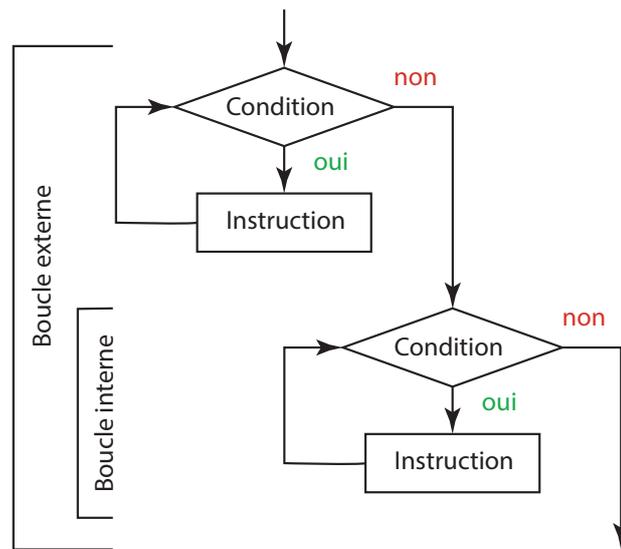
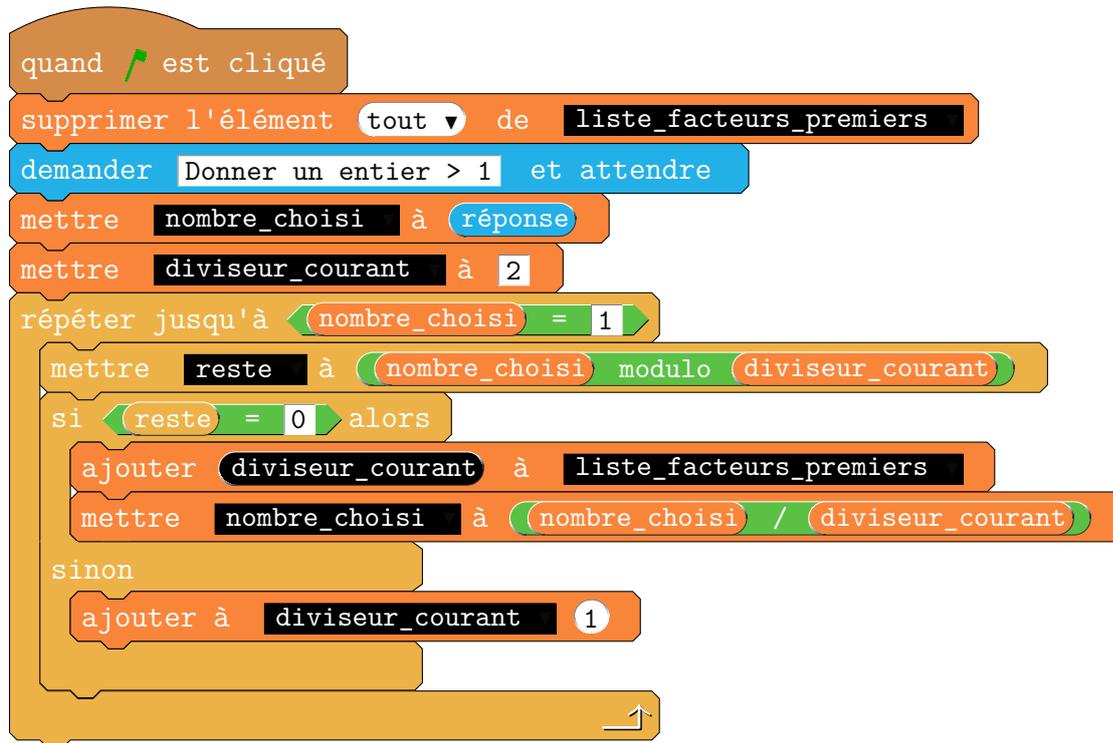


Figure 9.19 – Boucles imbriquées



9.4.3.6 Exemple des commandes plus riches



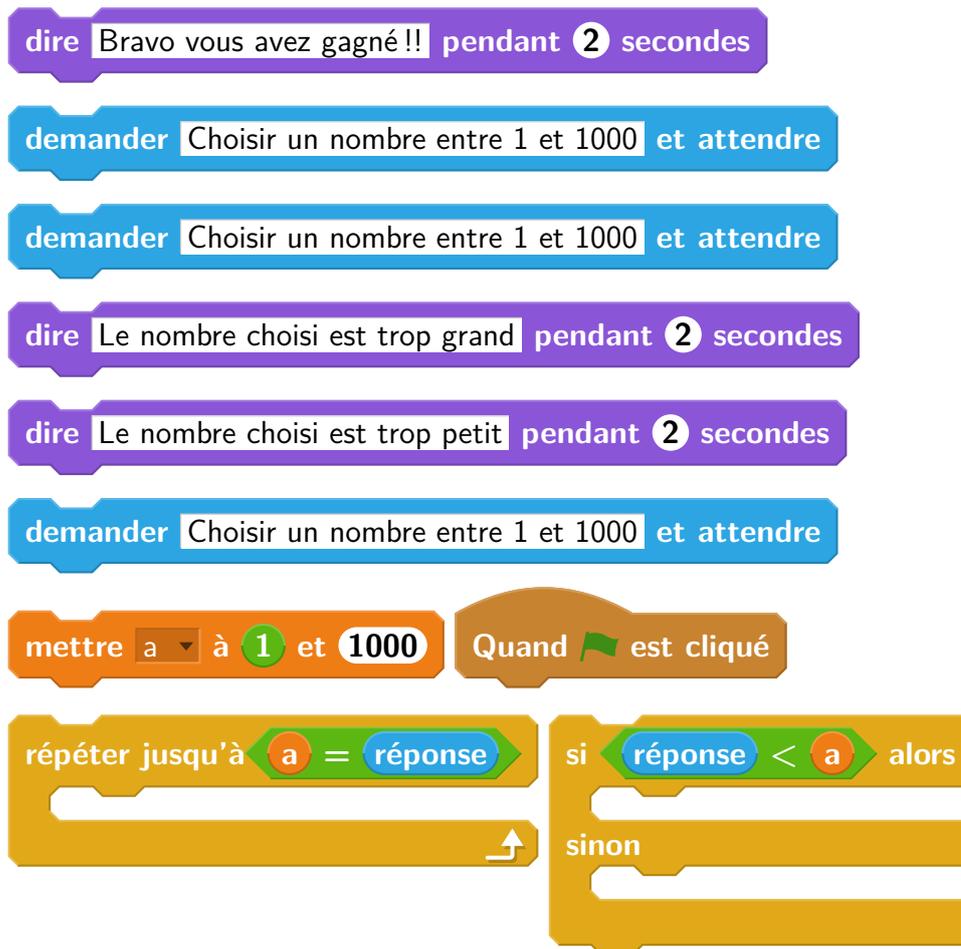
Activité - 3 : L'objectif de cette activité est d'écrire, mettre au point et exécuter un programme simple :

- savoir interpréter une condition ;
- mettre au point un programme en réponse à un problème donné.

On donne l'algorithme suivant,

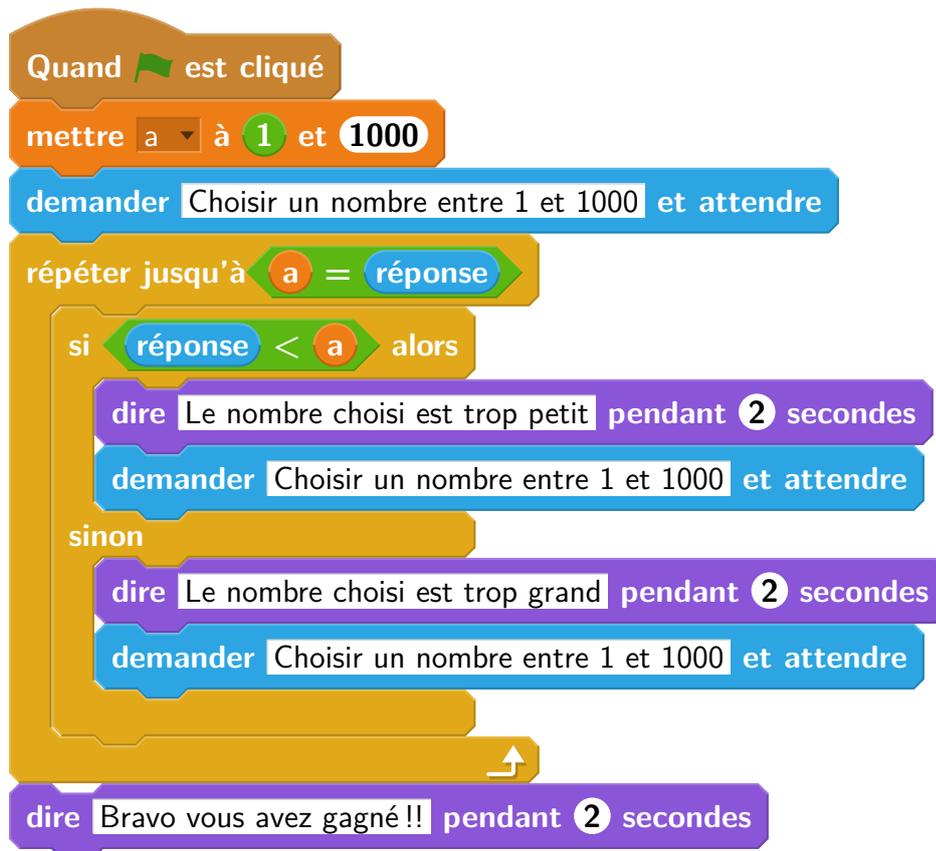
1. L'ordinateur choisit un nombre entier a aléatoire entre 1 et 1000 ;
2. L'ordinateur demande au joueur de donner un nombre entier n entre 1 et 1000 ;
3. TANT QUE n est différent de a
4. SI $n < a$ ALORS
Scratch dit « le nombre choisi est trop petit »
5. SINON
Scratch dit « le nombre choisi est trop grand »
FIN TANT QUE
6. Scratch dit « Bravo, vous avez gagné ! »

Question : Dessiner l'algorithme de cet algorithme et écrire le programme avec Scratch en combinant les commandes ci-dessous.



Dans les deux classes de seconde composées de 24 groupes d'élèves, seulement 5 groupes n'arrivaient pas à dresser l'algorithme et 3 groupes pour l'écriture du programme en Scratch. Plus précisément, il n'arrivaient pas à composer et remettre en ordre les commandes en désordres sus présentées.

Correction : Nous présentons ci-dessous l'ordre de séquences de commandes de l'algorithme.



Les deux premières séances sont consacrées plutôt sur l'algorithmique, tandis que la troisième séance sur la programmation. Après les trois séances très chargées, on attendait que les élèves soient capables de lire, d'exécuter et de comprendre un algorithme. De plus, ils arrivent aussi à concevoir un algorithme simple et de le traduire avec un langage de programmation Scratch.

9.4.4 La quatrième séance : Évaluation finale

Cette séance d'évaluation vise à vérifier la compétence de chaque élève par rapport aux résultats attendus des trois séances précédentes : *lire, exécuter, comprendre et concevoir un algorithme simple et écrire un petit programme en Scratch*. La séance durait 2 h, dont 1 h s'est consacrée à une évaluation individuelle sur papier et 1 h une évaluation en groupe sur machine (Cf. Tableau 9.5).

<p>Séance d'évaluation de la classe de seconde Discipline : Algorithmique et programmation Durée : 2 h</p>
<p>Travail individuel sur papier - durée 1 h</p>
<p>Exercice 1 A l'issue de l'algorithme suivant, quel nombre est stocké dans la variable A? Dans la variable B? $A \leftarrow 3$ $B \leftarrow 4$ $C \leftarrow A$ $A \leftarrow B$ $B \leftarrow C$</p> <p>Exercice 2 Que fait l'algorithme suivant ? Faire l'algorithme correspondant Saisir A; Saisir B; $C \leftarrow A * B$; $D \leftarrow 2 * (A + B)$; Afficher C; Afficher D.</p> <p>Exercice 3 Faire l'algorithme et écrire un programme sur Scratch qui demande l'âge de l'utilisateur et répond « vous êtes mineur » ou « vous êtes majeur » suivant le cas.</p>
<p>Travail en groupe de 4 par machine - durée 1 h</p>
<p>Programmation En tenant compte de l'activité 3 : 1. L'ordinateur choisit un nombre entier a aléatoire entre 1 et 1000 ; 2. L'ordinateur demande au joueur de donner un nombre entier n entre 1 et 1000 ; 3. TANT QUE n est différent de a 4. SI $n < a$ ALORS Scratch dit « le nombre choisi est trop petit » 5. SINON Scratch dit « le nombre choisi est trop grand » FIN TANT QUE 6. Scratch dit « Bravo, vous avez gagné ! » On demande au groupe d'améliorer le programme en répondant à ces deux questions : a) On souhaite compter le nombre de coups nécessaires par le joueur avant de gagner. Utiliser la variable Score et coder cette nouvelle fonctionnalité. b) On souhaite conserver les nombres choisis par le joueur</p>

Tableau 9.5 – Sujet d'évaluation

9.4.5 Analyses des résultats d'évaluation

9.4.5.1 Analyse des résultats de la classe de seconde I

Nous pouvons construire une table binaire (Cf. tableau D.2) à partir des notes montées dans le tableau D.1. Après traitement par CHIC- M_{GK} , nous avons 105 règles d'association valides selon M_{GK} , avec un seuil de risque de 10% fixé par

nous-même, dont parmi elles, 44 sont des règles positives et 61 sont des règles négatives, avec $0.0104 \leq \text{supp}_{(n)M_{GK}}^f \leq 1$ qui forment 8 Paires Orientées PO, i.e. $\text{card}(\text{coh}_{\text{supp}_{(n)M_{GK}}}) = 8$ avec $0.067 \leq \text{coh}_{\text{supp}_{(n)M_{GK}}} \leq 1$ (Cf. Tableau 9.6). Le tableau 9.7 nous montre les valeurs des cohésions inter-classes $\text{coh}_{\text{supp}_{(n)M_{GK}}}(\underline{C})$, des implications inter-classes $\psi(\underline{C}, \underline{C}')$, des niveaux $s(\Omega, H)$ et nœuds significatifs $v(\Omega, H)$ de chaque niveau de la hiérarchie H , ainsi que les 5 méta-règles avec un risque de se tromper à $\alpha = 10\%$, pour les résultats de la seconde I.

couple de variables	$\text{coh}_{\text{supp}_{(n)M_{GK}}}$
((FRS faible,MATHS faible))	1.0
((MATHS fort,INFO fort))	1.0
((INFO faible,ANG faible))	0.547
((MALG faible,MATHS faible))	0.383
((MATHS moy,INFO moy))	0.369
((ANG fort,FRS fort))	0.277
((ANG fort,INFO fort))	0.218
((INFO fort,MALG fort))	0.067

Tableau 9.6 – Tableau montrant les valeurs de cohésions des couples de variables formées pour la classe de seconde I

D'après le tableau 9.6, nous pouvons dire qu'il y a une forte relation entre les variables **FRS faible** et **MATHS faible** de même pour **MATHS fort** et **INFO fort**, alors on peut voir l'orientation de ces relations; tandis que la relation entre **INFO fort** et **MALG fort** est très faible, alors le fait de dire que les élèves forts en informatique sont généralement fort en Malagasy pourrait fausser interprétation et vice versa.

Niveau	Règle	Cohesion inter-classe $\text{coh}_{\text{supp}_{(n)M_{GK}}}(\underline{C})$	Implication inter-classe $\psi(\underline{C}, \underline{C}')$	Niveau significatif $s(\Omega, H)$	Noeud significatif $v(\Omega, H)$
1	(FRS faible=>MATHS faible)	1	1	0,00824805	0,00824805
2	(MATHS fort=>INFO fort)	1	1	0,00003947	-0,00820858
3	(INFO faible=>ANG faible)	0,547	0,73333333	-0,00267016	-0,00270963
4	(MATHS moy=>INFO moy)	0,369	0,65454545	-0,00404461	-0,00137445
5	(ANG fort=>FRS fort)	0,277	0,61616162	-0,00485319	-0,00080857

Tableau 9.7 – Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau de la classe de seconde I.

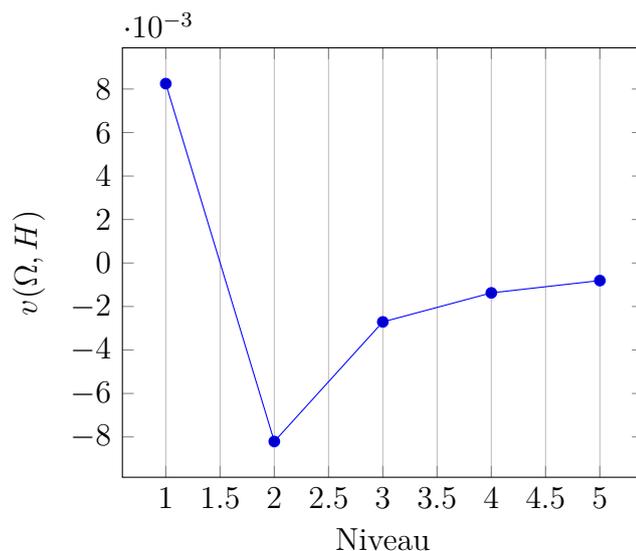


Figure 9.20 – Graphe représentant les nœuds significatifs des 5 classes de la seconde I

D'après la figure 9.20, les nœuds significatifs sont les extremums de la courbe, i.e. les nœuds au niveau $H = \{1, 2, 5\}$, représentés en croix dans la figure 9.21, les cohésions inter-classes $coh_{supp(n)M_{GK}}(\underline{C})$ à l'axe des ordonnées et les implications inter-classe $\psi(\underline{C}, \underline{C}')$ au dessus des rectangles. 60% des nœuds sont significatifs.

Interprétations : Les élèves de la seconde I qui sont faibles en français sont généralement faibles en mathématiques, mais ceux qui sont forts en maths sont généralement forts en informatique. Les élèves de la classe de seconde I qui sont forts en anglais sont généralement aussi forts en français. Dans cette classe, les langues étrangères jouent leurs rôles importants dans l'apprentissage des matières scientifiques. Remarquons que la moyenne de la classe en informatique est de 10.34/20.

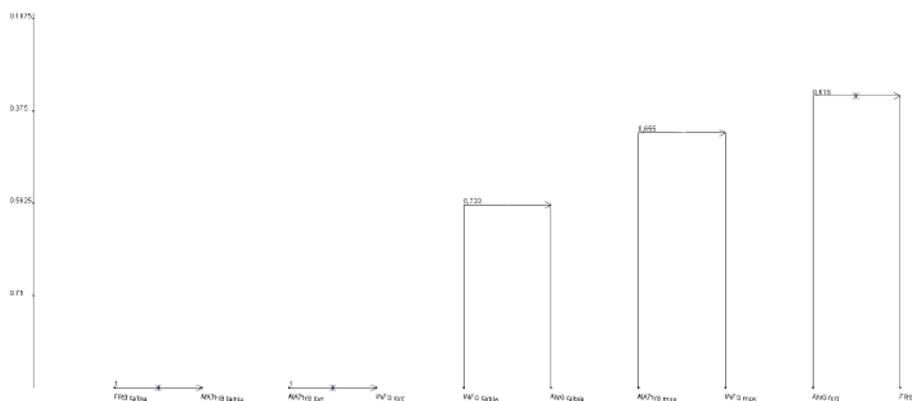


Figure 9.21 – Extrait du dendrogramme représentant des 5 classes de la seconde I

De plus, l'ensemble du groupe optimal des élèves qui participent à la règle (FRS faible=>MATHS faible) est

$$Gr_{opt} = \{e3, e4, e5, e6, e8, e9, e10, e11, e12, e13, e14, e15, e16, e18, e20, e22, e23, e24, e25, e26, e27, e28, e29, e30, e31, e33, e34, e35, e36, e37, e38\}$$

9.4.5.2 Analyse des résultats de la classe de seconde II

Nous pouvons construire une table binaire (Cf. tableau D.4) à partir des notes montées dans le tableau D.3. Après traitement par CHIC- M_{GK} , nous avons 105 règles d'association valides selon M_{GK} , avec un seuil de risque de 10% fixé par nous-même, dont parmi elles, 40 sont des règles positives et 65 sont des règles négatives, avec $0.0129 \leq supp_{(n)M_{GK}}^f \leq 1$ qui forment 17 Paires Orientées PO, i.e. $card(coh_{supp_{(n)M_{GK}}}) = 17$ avec $0.249 \leq coh_{supp_{(n)M_{GK}}} \leq 1$ (Cf. Tableau 9.8). Le tableau 9.9 nous montre les valeurs des cohésions inter-classes $coh_{supp_{(n)M_{GK}}}(\underline{C})$, des implications inter-classes $\psi(\underline{C}, \underline{C}')$, des niveaux $s(\Omega, H)$ et nœuds significatifs $v(\Omega, H)$ de chaque niveau de la hiérarchie H , ainsi que les 7 méta-règles avec un risque de se tromper à $\alpha = 10\%$, pour les résultats de la seconde I.

couple de variables	$coh_{supp(n)M_{GK}}$
((MALG faible,MATHS faible))	1.0
((FRS fort,MALG fort))	1.0
((ANG fort,MALG fort))	1.0
((MATHS moy,MALG fort))	1.0
((MATHS fort,MALG fort))	1.0
((FRS faible,MATHS faible))	1.0
((FRS fort,ANG moy))	1.0
((MATHS fort,FRS fort))	1.0
((MATHS moy,INFO fort))	1.0
((ANG faible,FRS faible))	0.832
((MALG fort,ANG moy))	0.658
((FRS faible,MALG faible))	0.554
((FRS moy,ANG moy))	0.5
((ANG faible,MALG faible))	0.468
((MALG fort,FRS moy))	0.322
((MALG moy,INFO moy))	0.287
((INFO faible,MALG faible))	0.249

Tableau 9.8 – Tableau montrant les valeurs de cohésions des couples de variables formées pour la classe de seconde II

Niveau	Règle	Cohesion inter-classe $coh_{supp(n)M_{GK}}(\mathcal{C})$	Implication inter-classe $\psi(\mathcal{C}, \mathcal{C}')$	Niveau significatif $s(\Omega, H)$	Noeud significatif $v(\Omega, H)$
1	(MALG faible=>MATHS faible)	1	1	0,00824805	0,00824805
2	(FRS fort=>MALG fort)	1	1	0,00003947	-0,00820858
3	(MATHS fort=>(FRS fort=>MALG fort))	1	1	0,00422393	0,00418447
4	(MATHS moy=>INFO fort)	1	1	-0,00485319	-0,00907712
5	(FRSfaible=>(MALGfaible=>MATHS faible))	0,82130271	1	-0,00099360	0,00385958
6	(FRS moy=>ANG moy)	0,5	0,71212121	-0,00608546	-0,00509186
7	(MALG moy=>INFO moy)	0,287	0,62	-0,00630746	-0,00022200

Tableau 9.9 – Tableau représentant les cohésions inter-classes, les implications inter-classes, les niveaux et nœuds significatifs de chaque niveau de la classe de seconde II.

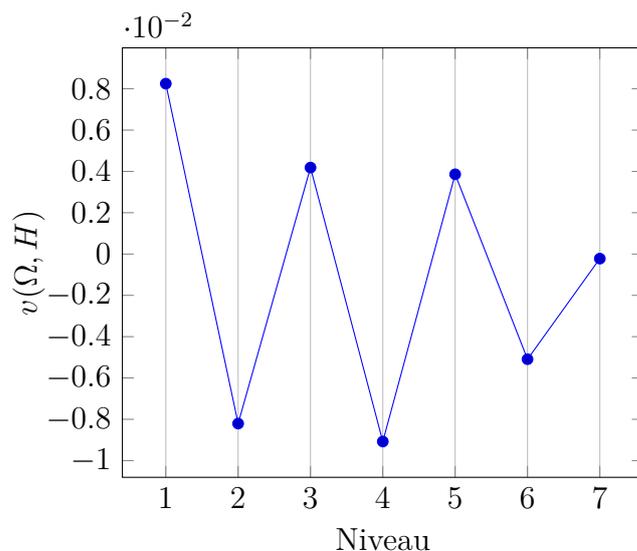


Figure 9.22 – Graphe représentant les nœuds significatifs des 7 classes de la seconde II

D'après la figure 9.22, les nœuds significatifs sont les extremums de la courbe, i.e. les nœuds au niveau $H = \{1, 2, 3, 4, 5, 6, 7\}$, représentés en croix dans la figure 9.23, les cohésions inter-classes $coh_{supp(n)M_{GK}}(\underline{C})$ à l'axe des ordonnées et les implications inter-classe $\psi(\underline{C}, \underline{C}')$ au dessus des rectangles.

Interprétations : Les élèves de la classe de seconde qui sont faibles en malagasy sont généralement faibles en mathématiques. Les élèves qui sont forts en maths sont généralement fort en français et en malagasy. Dans cette classe, les élèves qui ont de niveau moyen en maths sont forts en informatique. La moyenne de la classe en informatique est de 9.76/20.

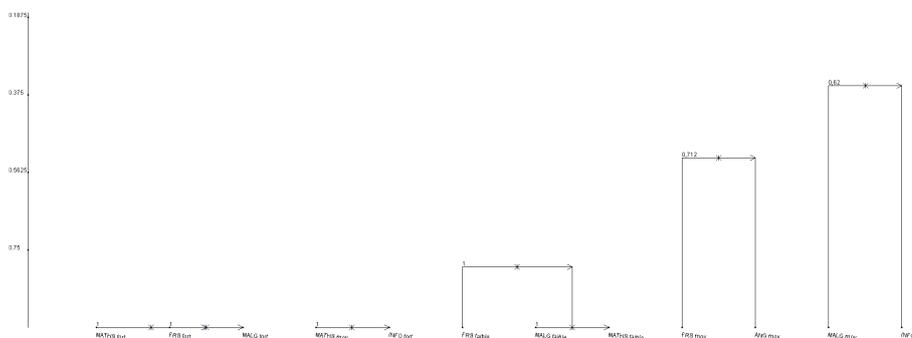


Figure 9.23 – Extrait du dendrogramme représentant des 7 classes de la seconde II

Classe	Matières				
	MALG	FRS	ANG	MATHS	INFO
Seconde I	10,16	10,41	10,20	6,84	10,34
Seconde II	7,43	7,54	8,84	4,57	9,76

Tableau 9.10 – Moyenne générale des classes en matières littéraires et scientifiques

9.5 Conclusion

Dans ce chapitre, nous avons testé notre curriculum en informatique de la classe de seconde dans un lycée privé d'Antsiranana, et avons mis les élèves en situation d'apprentissage selon le modèle que nous avons pris dans le chapitre 6, §6.3.4. Après avoir passé 3 séances de deux heures par classe pour enseigner l'algorithmique et programmation et une séance d'évaluation, nous avons eu un résultat juste à la moyenne pour la classe de seconde I et en-dessous de la moyenne pour la classe de seconde II. Une remarque s'avère pertinente que les disciplines linguistiques jouent un rôle clé dans l'apprentissage des matières scientifiques, plus particulièrement l'informatique.

Conclusion générale et perspectives

Les travaux de recherche présentés dans ce mémoire concernent la fouille et la classification des données dans un contexte binaire selon la mesure de qualité des règles d'association M_{GK} . Un nouvel outil informatique d'analyse de données a été conçu en conséquence afin de pouvoir résoudre des problèmes dans le domaine de didactique de l'informatique. Ces travaux de recherche se concentrent à la fois d'ordre mathématiques et proba-statistique, où des propriétés et algorithmes sont établis pour le nouveau logiciel informatique sus-mentionné, mais également, d'ordre pratique, pour valider nos nouvelles méthodes par des études expérimentales sur le curriculum en informatique et la situation d'apprentissage de ce programme scolaire. Ainsi, nous avons organisé notre mémoire en trois grandes parties.

La première partie est consacrée sur l'état de l'art autour du domaine de recherche sur la Data science, le Big data qui nous a permis déjà de baser sur le principe de traitement *map-reduce* et l'ECD et ses étapes de processus afin de cadrer le travail sur le Data mining descriptif et ses méthodes dont parmi elles, les « *règles d'association* » - *map* et le « *clustering* » - *reduce*. Nous avons présenté dans cette partie quelques notions mathématiques et des fondamentaux sur les treillis, les familles de Moore, les correspondances de Galois, le système implicatif et les hiérarchies et qui seront utiles pour les deux méthodes choisies sus-mentionnées. Enfin, concernant les règles d'association, nous avons montré quelques algorithmes d'extraction des règles d'association déjà parus dans des littératures, parmi eux, le premier algorithme d'extraction des règles d'association Apriori qui génère les motifs fréquents et filtre les candidats qui ne sont pas valides.

La deuxième partie concerne nos contributions sur l'extraction des règles d'association M_{GK} -valides et apport par considération du *support* de ces règles M_{GK} -valides noté $supp_{M_{GK}}$ (Rakotomalala et al., 2017a). Un algorithme et un algorithme détaillé d'extraction des règles d'association M_{GK} -valides ont été établis (Rakotomalala et Totohasina, 2018). La normalisée du support M_{GK} -valides, notée $supp_{(n)M_{GK}}$ donne un contraste à la valeur de $supp_{M_{GK}}$ et fait apparaître des pépites de connaissances. De ce fait, nous avons remarqué le double rôle de la *confiance*, non seulement en normalisation (faire apparaître les règles à forte confiance), mais aussi à l'évaluation des cohésions (filtre). La confiance joue alors un rôle de régulateur. Une question se pose, « *peut-on formaliser mathématiquement et établir une équation algébrique pour pouvoir donner la valeur de la confiance en tant que régulateur ?* ». Le $supp_{(n)M_{GK}}$ nous a permis ensuite de créer un nouvel indice de cohésion implicative selon M_{GK} (Rakotomalala et al., 2017b), (Rakotomalala et Totohasina, 2018), mais « *est-il possible d'évaluer la fiabilité et la portabilité de cet indice ?, et pourquoi pas tous les indices de mesure de qualité recensés dans les littératures ?* ».

Cet nouvel indice de cohésion nous est utile pour la classification hiérarchique implicative et cohésitive selon la mesure M_{GK} (Rakotomalala, Totohasina, et Diatta, 2018), (Rakotomalala, Ralahady, et Totohasina, 2018), (Rakotomalala et Totohasina, 2019). Nous avons utilisé ces contributions dans la conception et la réalisation du logiciel informatique CHIC- M_{GK} en java B.1. Pour respecter le cycle de vie d'un logiciel, il s'avère utile de « *mettre à jour et d'insérer d'autres fonctionnalités à notre logiciel CHIC- M_{GK} et surtout de prendre en compte le pré-traitement de données (conversion des données réelles en binaires)* ».

La troisième partie est consacrée sur le volet didactique. D'abord, nous avons mené une étude sur les stratégies à mettre en œuvre pour l'intégration effective des TIC dans la pratique pédagogique à Madagascar selon les méthodes AFM et CAH (Rakotomalala et al., 2012), une autre étude sur l'analyse et diagnostique sur l'alphabétisation en informatique des étudiants de l'enseignement supérieur de Madagascar selon notre méthode CHIC- M_{GK} (Rakotomalala, Ralahady, et Totohasina, 2018). Une étude sur l'analyse de curriculum dans le modèle anglo-saxon et comparé avec celui de l'ENSET a été faite pour pouvoir proposer le programme en informatique dans le lycée - classe de seconde pour combler le vide détecté lors de l'analyse du curricu-

lum de l'ENSET ([Rakotomalala et Totohasina, 2019](#)). Enfin, une expérimentation divisée en 4 séances a été menée au lycée St Jean d'Antsiranana, pour l'apprentissage du cours algorithmique et programmation en classe de seconde avec l'outil de programmation Scratch. Une évaluation et analyse des résultats ont été faite. Le résultat obtenu lors de l'apprentissage de l'algorithmique en seconde est encore maigre. Des recherches approfondies s'avèrent primordiales pour l'enseignement de l'algorithmique et programmation au lycée au niveau de tous les acteurs, le rôle de l'état, la capacité des enseignants, les prérequis des élèves et leur environnement social. Le terrain est donc vaste sur le problème de didactique de l'informatique, malgré les plusieurs paramètres qui entrent en jeu dans l'enseignement d'un discipline.

Références bibliographiques

- Aggarawal, C., Charu, C., et Yu, P. (1998). A new framework for item-set generation. Dans *Proceedings of the seventeenth acm sigact-sigmod-sigart symposium on principles of database systems* (p. 18-24). Récupéré de http://www09.sigmod.org/disc/disc99/disc/sigmod_papers/slipp_a_new_framework_for_/slides.pdf
- Agrawal, R., Imielinski, T., et Swami, A. (1993a). Database mining : A performance perspective. *IEEE Trans. Knowledge and Data Eng.*, 5(6), 914-925. Récupéré de <http://wwwcssrv.almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/tkde93.pdf>
- Agrawal, R., Imielinski, T., et Swami, A. (1993b). Mining association rules between sets of items in large databases. Dans P. Buneman et S. Jajodia (dir.), *Acm sigmod international conference on management of data* (Vol. 22, p. 207-216). Washington, USA. Récupéré de <https://pdfs.semanticscholar.org/9938/8a0d3beb526a13421ed088743b30a75e2b29.pdf>
- Agrawal, R., et Shafer, J. C. (1996). Parallel mining of association rules. *IEEE Trans. On Knowledge And Data Engineering*, 8(6), 962-969. doi : 10.1109/69.553164
- Agrawal, R., et Srikant, R. (1994). Fast algorithms for mining association rules in large databases. Dans *Proceedings of the 20th international conference on very large data bases, VLDB* (Vol. 1215, p. 487-499). San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.,. Récupéré de <https://dl.acm.org/citation.cfm?id=672836>
- Agrawal, R., et Srikant, R. (1995). Mining sequential patterns. Dans W. D. U. .-. m. ICDE '95 IEEE Computer Society (dir.), *Proceedings of*

- the 11th international conference on data engineering* (Vol. 95, p. 3-14). Washington, USA. Récupéré de <https://pdfs.semanticscholar.org/d6a0/e0b04a020ac6422b98b8e63027a6178060fd.pdf>
- Aho, A. V., Hopcroft, J. E., Ullman, U. D., et Moreau, J. M. (1989). *Structures de données et algorithmes*. Paris, Inter Ed. D. L.
- Alessandri, M. (2010). *Extraction des motifs fréquents pertinents pour la préconisation d'achats en marketing one-to-one*. (Thèse de doctorat). Paris, CNAM.
- Altet, M. (1994). *La formation professionnelle des enseignants*. Paris, PUF.
- Arnaud, P. (1989). Contribution à une histoire des disciplines d'enseignement : la mise en forme scolaire de l'éducation physique. *Revue française de pédagogie*, 29-34.
- Arsac, J. (1987). *Les machines à penser*. Paris : Seuil.
- Arsac, J. (1988). La didactique de l'informatique : un problème ouvert? *Actes du colloque francophone sur la didactique de l'informatique, AFDI, René Descartes, Paris, septembre 1988*, 9–18. Récupéré de <https://edutice.archives-ouvertes.fr/edutice-00359090/document>
- Astolfi, J. P., et Develay, M. (1989). La didactique des sciences. *Que sais-je*(2448), 9-10.
- Audigier, F. (1996). La didactique comme un oignon. *Educations, janv-fev*(7), 34-38. Récupéré de http://www.ressources-de-la-formation.fr/index.php?lvl=notice_display&id=25678
- Avanzini, G. (1976). *Introduction aux sciences de l'éducation : les orientations de la recherche et le développement des méthodes dans le champ de l'éducation moderne*. E Privat (Mésopé) Toulouse. Récupéré de https://www.persee.fr/doc/rfp_0556-7807_1977_num_38_1_2103_t1_0035_0000_1
- Azé, J., Lucas, N., et Sebag, M. (2003). Fouille de données visuelle et analyse de facteur de risque médical. Dans *Extraction et gestion des connaissances*

- (EGC'2003), *actes des troisièmes journées extraction et gestion des connaissances, lyon, france, 22-24 janvier 2003* (Vol. 17, p. 183-188). Récupéré de <https://dblp.org/rec/bib/conf/f-egc/AzeLS03>
- Bachelard, G. (1938). *La formation de l'esprit scientifique : contribution à une psychanalyse de la connaissance objective*. Librairie philosophique Vrin, Paris, 1ere édition.
- Barbut, M., et Monjardet, B. (1970). *Ordre et classification algèbre et combinatoire*. Paris, Hachette.
- Barchechath, E., et Pouts-Lajus, S. (1990). Sur l'interactivité postface. *K., Crossley & L., Green. Le design des didacticiens. Paris : OTE (Observatoire des Technologies pour l'Éducation en Europe)*.
- Barma, S. (2010). Analyse d'une démarche de tranformation des pratiques en sciences dans le cadre du nouveau programme de formation pour le secondaire, à la lumière de la théorie de l'activité. *Revue canadienne de l'éducation*, 33(4), 677-710. Récupéré de https://www.jstor.org/stable/pdf/canajeducrevucan.33.4.677.pdf?seq=1#page_scan_tab_contents
- Baron, G.-L. (1981). Dix ans après (où en est l'INRP?). *Bulletin de l'EPI*, 21, 19-23.
- Baron, G.-L., et Brouillard, E. (1996). *L'informatique et ses usagers dans l'éducation*. Presses Universitaires de France, L'Éducateur, Paris.
- Baron, G. L., et Bruillard, E. (2001). Une didactique de l'informatique? *Revue Française de Pédagogie*, 135, 163-172. Récupéré de <https://www.jstor.org/stable/41201696>
- Basque, J. (2005). Une réflexion sur les fonctions attribuées aux TIC en enseignement universitaire. *Revue internationale des technologies en pédagogie universitaire, Télé-université*, 2(1), 30-41. Récupéré de www.profetic.org/revue
- Beauné, A. (2010). Théorie de l'activité : applications au domaine des TICE. Récupéré de <http://www.adjectif.net/spip/spip.php?article77&lang=fr>

- Beauquier, D., Berstel, J., et Chrétienne, P. (1992). *Eléments d'algorithmique*. Masson. Récupéré de <http://www-igm.univ-mlv.fr/>
- Bendoly, E. (2003). Theory and support for process frameworks of knowledge discovery and data mining from ERP systems. *Information and Management*, 40(7), 639-647. doi : 10.1016/S0378-7206(02)00093-9
- Bessé, H. (1995). Méthodes, méthodologie, pédagogie. *Le français dans le monde Recherches et Applications, Méthodes et méthodologies*,(n spécial), 96-108.
- Birkhoff, G. (1933). On the combination of subalgebras. Dans *Mathematical proceedings of the cambridge philosophical society* (Vol. 29, p. 441-464). doi : 10.1017/S0305004100011464
- Birkhoff, G. (1940). *Lattice theory* (Vol. 25). American Mathematical Society, 1st edition.
- Blanchard, J. (2005). *Un système de visualisation pour l'extraction, l'évaluation, et l'exploration interactives des règles d'association*. (Thèse de doctorat, Université de Nantes). Récupéré de <https://www.theses.fr/2005NANT2062>
- Blanchard, J., Guillet, F., Briand, H., et Gras, R. (2005). Assessing rule with a probabilistic measure of deviation from equilibrium. Dans ENST (dir.), *Actes du 11th international symposium on applied stochastic models and data analysis asmda* (p. 191-200). Brest, France. Récupéré de <https://hal.archives-ouvertes.fr/hal-00420982>
- Blanchard, J., Kuntz, P., Guillet, F., et Gras, R. (2004). Mesure de la qualité des règles d'association par l'intensité d'implication entropique. *Revue Nationale des Technologies de l'Information*(1), 33-44. Récupéré de <https://hal.archives-ouvertes.fr/hal-00442535>
- Blin, F., et Munro, M. (2008). Why has'nt technology disrupted academic's teaching practices ? understanding resistance to change through the lens of activity theory. *Computers & Education*, 50(2), 475-490. doi : 10.1016/j.compedu.2007.09.017
- Bonneau, C. (2011). Co-configuration d'une plateforme à code source ouvert en organisation : analyser la transformation d'un outil et des pratiques de travail

- avec la théorie de l'activité. *Revue de communication sociale et publique*, 5, 23-36. Récupéré de <https://journals.openedition.org/communiquer/417>
- Bosler, U., Gumbo, S., Taylor, H., Abas, Z. W., Duchâteau, C., Morel, R., et Waker, P. (1996). *Informatique pour l'enseignement secondaire : programme scolaire* (T. V. Weert et D. Tinsley, dir.). Elaboré par un groupe de travail de l'IFIP sous l'égide de l'UNESCO, Paris Récupéré de https://unesdoc.unesco.org/ark:/48223/pf0000097323_fre
- Bourguin, G., Derycke, A., et Tarby, J. (2005). Systèmes interactifs en co-évolution réflexions sur les apports de la théorie de l'activité au support des pratiques collectives distribuées. *Revue d'Interaction Homme-Machine*, 6(1). Récupéré de <http://europia.org/RIHM/V6N1/2-RIHM-Article%20Bourguin-Derycke%20PDF.pdf>
- Brin, S., Motwani, R., et Silverstaein, C. (1997). Beyond market baskets : Generalizing association rules to correlationn. [s.l.] : *Proceedings of the ACM SIGMOD Conference*, 265-276.
- Brin, S., Motwani, R., Ullman, J., et Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. *ACM SIGMOD RECORD*, 26(2), 255-264. Récupéré de <https://dl.acm.org/citation.cfm?id=253325> doi : 10.1145/253262.253325
- Brisson, L. (2004). *Mesure d'intérêt subjectif et représentation des connaissances*. (Thèse de doctorat). Université de Nice.
- Bronckart, Y., et Chiss, J. (2005 [ref. du 06-05-2018]). Cours et outils de formation : Petit dictionnaire de pédagogie - didactique. « *Didactique* », *Site Encyclopaedia Universalis*. Récupéré de <http://www.universalisedu.com>
- Brousseau, G. (1978). L'observation des activités didactiques. *Revue française de pédagogie*, 45, 130-140. Récupéré de https://www.persee.fr/doc/rfp_0556-7807_1978_num_45_1_1669
- Brousseau, G., Wittwer, J., Delis, M., et Caid, M. (1969). Essai d'un système d'appréciation continue des connaissances. *Bulletin de psychologie*, XXII 6-8(282), 458-473.

- Brun, J. (1994). Evolution des rapports entre la psychologie du développement congénitif et la didactique des mathématiques. Dans C. L. P. T. d. M. Artigue R. Gras (dir.), *Vingt ans de didactique des mathématiques en France. hommage à Guy Brusseau et Gérard Vergnaud* (p. 67-83). La pensée sauvage. Récupéré de <https://rdm.penseesauvage.com/Vingt-ans-de-didactique-des.html>
- Cassel, L., Clements, A., Guzdial, M., McCauley, R., McGettrick, A., Sloan, B., ... W Weide, B. (2008). *Computer science curricula 2008 : An interim revision of CS 2001* (Rapport technique). New York : ACM Press. and IEEE Computer Society Press.
- Cazellet, L. (2003). *Les NTIC, "outil" ou "instrument" pour l'apprentissage et la formation ? réflexions sur l'impact de l'utilisation des NTIC dans un dispositif expérimental de formation auprès de professionnels de la santé* (Thèse de doctorat). Université de Provence Aix-Marseille.
- Chafai, D. (2001). *Sur les inégalités de Sobolev logarithmiques en théorie de l'information et pour des systèmes de spins conservatifs en mécanique statistique* (Thèse de doctorat). Université Paul Sabatier :s.n. Vol. 1. 2001TOU30139.
- Chang, C., Denning, P., James, H., Cross, I., Engel, G., Sloan, R., ... Wolz, U. (2001). *Computing curricula 2001, CC 2001* (Rapport technique). New York : ACM Press. and IEEE Computer Society Press.
- Chaptal, A. (2007, 12). Usages prescrits ou annoncés, usages observés. réflexions sur les usages scolaires du numérique par les enseignants. *Document numérique*, 10, 136. Récupéré de https://www.researchgate.net/publication/240978659_Usages_prescrits_ou_annonces_usages_observes_Reflexions_sur_les_usages_scolaires_du_numerique_par_les_enseignants doi : 10.3166/dn.10.3-4.81-106
- Charlionet, R., Gascuel, P., Gaudin, F., Guespin-Michel, J., Gayoso, J., et Ripoll, C. (2005). *Emergence, complexité et dialectique : Sur les systèmes dynamiques non linéaires (Y a-t-il de la non-linéarité dans la sémantique ?). F Gaudin*. Odile Jacob, Paris.

- Chattratchat, J., Darlington, J., Ghanem, M., Guo, Y., Huning, H., Kohlerand, M., ... Yang, D. (1997). Large scale data mining : Challenges and responses. Dans *Proceedings of the 3th international conference on knowledge discovery and data mining* (p. 143-146). Récupéré de https://pdfs.semanticscholar.org/8a9a/fe674403f1920549659d7e4d95c5eecd2882.pdf?_ga=2.44401511.2108748133.1553978407-631268581.1553837932
- Chen, M., Han, J., et Yu, P. (1996). Data mining : an overview from a database perspective. *IEEE Trans. on Knowledge and Data Engineering*, 8(6), 866-881. Récupéré de http://cs.nju.edu.cn/zhouzh/zhouzh.files/course/dm/reading/reading01/chen_tkde96.pdf doi : 10.1109/69.553155
- Chessel, D. (1997). L'analyse factorielle multiple. Récupéré de <http://pbil.univlyon1.fr/ADE-4/.27pp>
- Cheung, D. W., Ng, V. T., Fu, A. W., et Fu, Y. J. (1996). A fast distributed algorithm for mining association rules. Dans I. C. S. T. C. on Data Engineering et A. SIGMOD (dir.), *Fourth international conference on parallel and distributed information systems* (p. 31-42). Miami Beach, FL, USA. Récupéré de <https://hub.hku.hk/bitstream/10722/45576/1/26205.pdf?accept=1> doi : 10.1109/PDIS.1996.568665
- Chevallard, Y. (1991). *La transposition didactique : du savoir savant au savoir enseigné*. La Pensée sauvage. Récupéré de https://www.persee.fr/doc/rfp_0556-7807_1986_num_76_1_2401_t1_0089_0000_1
- Choudary, A. K., Harding, J. A., et Tiwari, M. K. (2009). Data mining in manufacturing : a review based on the kind of knowledge. *Journal of Intelligent Manufacturing*, 20(5), 501-521. Récupéré de https://dspace.lboro.ac.uk/dspace-jspui/bitstream/2134/9795/9/Data%20mining_review%20Paper_JIM_final_accepted.pdf
- Cissé, D. D. (2008). Les TIC : instruments de médiation socioconstructiviste. Dans T. . K. T. Toure K. Tchombe (dir.), *ICT and changing mindsets in education*. Bamenda, Cameroon : Langaa; Bamako, Mali : ERN-WACA / ROCARE.. Récupéré de <https://docplayer.fr/19115592-15-les-tic-instruments-de-mediation-socioconstructiviste-1.html>

- Cole, M., et Engeström, Y. (1993). A cultural-historical approach to distributed cognition. Dans *G. salomon (ed.), distributed cognitions : Psychological and educational considerations* (p. 1-46). New York : Cambridge University Press. Récupéré de https://s3.amazonaws.com/academia.edu.documents/36965097/A_Cultural-Historical_Approach_to_Distributed_Cognition.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1553983932&Signature=0qLDrp0Ym4uSj4V%2BdhHuQjIWXhQ%3D&response-content-disposition=inline%3B%20filename%3DA_Cultural-Historical_Approach_to_Distri.pdf
- Coquidé, M. (2000). *Le rapport à l'expérimental*. (HDR). Université, ENS de Cachan.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., et Cazin, X. (1994). *Introduction à l'algorithmique*. Paris, Dunod.
- Cornu, F., et Vergnioux, A. (1992). *La didactique en questions*. CNDP, Hachette éducation.
- Cuq, J., et Gruca, I. (2002). *Cours de didactique du français langue étrangère et seconde*. Grenoble.
- Dasu, T., et Johnson, T. (2003). *Exploratory data mining and data cleaning*. (Vol. 479). John Wiley Sons, Inc, New York NY. Récupéré de <https://epdf.tips/queue/exploratory-data-mining-and-data-cleaning50403.html> doi : 10.1002/0471448354
- Daudé, D. (1992). *Analyse et justification de la notion de ressemblance entre variables qualitatives dans l'optique de la classification hiérarchique par avl* (Thèse de doctorat). Université de Rennes I.
- Davey, B. A., et Priestley, H. A. (1991). *Introduction to lattices and orders*. Cambridge University Press, 2nd edition.
- Develay, M. (1992). De l'apprentissage à l'enseignement : pour une épistémologie scolaire. *ESF*.
- Dhar, V. (2012). Data science and prediction. Dans *Communication of ACM* (Vol. 12, p. 64-73). Récupéré de <https://archive.nyu.edu/bitstream/2451/31553/2/Dhar-DataScience.pdf>

- Diatta, J., Ralambondrainy, H., et Totohasina, A. (2007). Towards a unifying probabilistic implicative normalised quality measure for association rules. Dans F. G. . H. J. Hamilton (dir.), *In quality measures in data mining* (Vol. 43, p. 242-255). Springer Berlin/Heidelberg.
- Dilworth, R. (1990). *Proof of a conjecture on finite modular lattices* (T. D. T. C. Mathematicians, dir.). Birkhäuser, Boston, MA. Récupéré de https://link.springer.com/chapter/10.1007/978-1-4899-3558-8_21#citeas
- Donnay, J. (1988). *Enseigner et apprendre*. Première partie, Département Education et Technologie, FUNDP.
- EduSCOL. (2009). *Ressources pour la classe de seconde - algorithmique*. Récupéré de http://media.eduscol.education.fr/file/Programmes/17/8/Doc_ress_algo_v25_109178.pdf
- Engeström, Y. (1987). *Learning by expanding : An activity-theoretical approach to developmental research*. Helsinki : Orienta-Konsultit. Récupéré de <http://lchc.ucsd.edu/mca/Paper/Engestrom/Learning-by-Expanding.pdf> doi : 10.1017/CBO9781139814744
- Engeström, Y. (2000a). Activity theory as a framework for analyzing and redesigning work. *Ergonomics*, 43(7), 960-974.
- Engeström, Y. (2000b). New forms of learning in co-configuration work. *Journal of Workplace Learning*, 16, 11-21.
- Engeström, Y. (2001). Expansive learning at work : Toward an activity-theoretical conceptualization. *Journal of Education and Work*, 14(1), 133-156.
- Engeström, Y. (2005). *Developmental work research : expanding activity theory into practice*. Berlin : Lehmanns Media.
- Engeström, Y., et Karkkainen, M. (1995). Polycontextuality and boundary crossing in expert cognition : learning and problem solving in complex work activities. *Learning and Instruction*, 5, 319-336.

- Escofier, B., et Pagès, J. (1984). L'analyse factorielle multiple : une méthode de comparaison de groupe de variables. Dans *Data analysis and informatics iii* (p. 41-55). North-Holland : Diday, E. Coll. (Eds.) Elsevier.
- Escofier, B., et Pagès, J. (1986). Le traitement des variables qualitatives et des tableaux mixtes par analyse factorielle multiple. Dans *Data analysis and informatics iv* (p. 179-191). North-Holland : Diday, E. Coll. (Eds.) Elsevier.
- Escofier, B., et Pagès, J. (1994). Multiple factor analysis : results of a three-year utilisation. Dans *Multway data analysis* (p. 277-285). North-Holland : Coppi, R. Bolasco, S. (Eds.) Elsevier Science Publishers B.V.
- Faerber, R. (2002). Le groupe d'apprentissage en formation à distance : ses caractéristiques dans un environnement virtuels. Dans E. du CRP (dir.), *Larose f karsenti t., la place des tice en formation initiale et continue à l'enseignement : bilan et perspectives* (p. 99-128). Université de Sherbrook.
- Fauré, C. (2007). *Découvertes de motifs pertinents par l'implémentation d'un réseau bayésien : application à l'industrie aéronautique*. (Thèse de doctorat). L'Institut National des Sciences appliquées de Lyon.
- Fayyad, U. M., Piatetsky-Shapiro, G., et Smyth, P. (1996a). From data mining to knowledge discovery : an overview. Dans *Advances in knowledge discovery and data mining* (p. 1-36). AAAI Press/MIT Press.
- Fayyad, U. M., Piatetsky-Shapiro, G., et Smyth, P. (1996b). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3), 37-54.
- Fayyad, U. M., Piatetsky-Shapiro, G., et Smyth, P. (1996c). Knowledge discovery and data mining : towards a unifying framework. Dans *Proceedings of the second international conference on knowledge discovery and data mining* (p. 82-88). Portland, OR.
- Fayyad, U. M., et Storloz, P. (1997). Data mining and KDD: promise and challenges. future generation computer systems. , 13, 99-115.
- Feno, D. (2007). *Mesures de qualité des règles d'association : normalisation et caractérisation des bases* (Thèse de doctorat). Université de la Réunion.

- Ferré, S. (2002). *Systèmes d'information logique : un paradigme logico-contextuel pour interroger, naviguer et apprendre*. (Thèse de doctorat). Université de Rennes I : [s.n.].
- Flajoret. (2015). *Encyclopaedia universalis*.
- Fleury, L. (1996). *Découverte de connaissances pour la gestion des ressources humaines* (Thèse de doctorat). Université de Nantes Vol. 2. 1996NANT2109.
- Frawley, W. J., Piatetsky-Shapiro, G., et Matheus, C. J. (1992). Knowledge discovery in databases - an overview. *AI Magazine*, 13(3), 57-70.
- Freitas, A. (1999). On rule interestingness measures. *Knowledge-Based System*, 12, 309-315.
- Galisson, R., et Coste, D. (1976). *Dictionnaire de didactique des langues*. Paris, Hachette.
- Ganascia, J.-G. (1998). *Dictionnaire de l'informatique et des sciences de l'information*. Edition Flammarion Paris.
- Ganter, B., et Wille, R. (1999). Formal concept analysis. Dans M. foundations (dir.), . Berlin, Springer Verlag.
- Gardarin, B. (1999). *Internet/intranet et bases de données*. Eyrolles Informatique.
- Garey, M. R., et Johnson, D. S. (1979). *Computers and intractability : A guide to the theory of np-completeness*. W. H. Freeman.
- Giordan, A. (1994). *L'élève et/ou les connaissances scientifiques : approche didactique de la construction des concepts scientifiques par les élèves*,. Peter lang, [1^{re} ed. 1983].
- Giordan, A. (1996). Les conceptions de l'apprenant : Un tremplin pour l'apprentissage. *Sciences humaines, Hors série n12*(fev-mars), 48-50.
- Giudici, P. (2003). *Applied data mining : Statistical methods for business and industry*. John Wiley Sons Ltd.

- Gras, R. (1979). *Contribution à l'étude expérimentale et à l'analyse de certaines acquisitions cognitives et de certains objectifs didactiques en mathématiques* (Thèse de doctorat). [éd.] Rennes : s.n., 1979. p. 332. Vol. 1, Thèse doctorat d'Etat. 1979REN1S024.
- Gras, R. (2013). *L'analyse statistique implicative : Méthode exploratoire et confirmatoire à la recherche de causalités*. Cépaduès, 2^{ème} édition.
- Gras, R., Couturier, R., et Bodin, A. (2003). Classification Hiérarchique Implicative et Cohésitive. *N poveda ku tatistickému softveru CHIC, Version, 3*, 2003. (00002)
- Gras, R., Kuntz, P., et Briand, H. (2001). Les fondements de l'analyse statistique implicative et quelques prolongements pour la fouille de données. *Mathématiques et sciences humaines. Mathematics and social sciences*(154).
- Gras, R., Kuntz, P., et Briand, H. (2003). *Hiérarchie orientée de règles généralisées en analyse implicative, extraction des connaissances et apprentissage*. Hermès.
- Gras, R., et Régnier, J.-C. (2009). Fondements théoriques de l'analyse statistique implicative. *Revue des Nouvelles Technologies de l'Information, Analyse Statistique Implicative - Une méthode d'analyse de données pour la recherche de causalités, RNTI-E-16*, 17-130.
- Grätzer, G. (1978). *General lattice theory*. Birkhäuser-Verlag and Springer, 1st edition.
- Guillaume, S. (2000). *Traitement des données volumineuses. mesures et algorithmes d'extraction de règles d'association et règles ordinales* (Thèse de doctorat). [éd.] Grenoble : ANRT. Nantes : s.n., 2000.
- Guillaume, S. (2010). Amélioration de la mesure M_{GK} . Dans *Actes de la 10eme rencontre de la société francophone de classification* (p. 41-44). Université de La Réunion.
- Hajek., P., Havel., et Cyril. (1966). The guha method of automatic hypothesis determination. *s.n. : Computing,, vol. 1*, pp. 293-308.
- Halbwachs, F. (1975). La physique du maître. *Revue française de pédagogie*, 33, 19-29.

- Halte, J. F. (1992). La didactique du français. *PUF*.
- Han, E., Karypis, G., et Kumar, V. (1997). Scalable parallel data mining for association rules. Dans *Proceedings of the acm sigmod international conference on management of data* (p. 277-288).
- Han, J., et Kamber, M. (2001). Data mining : concepts and technology. , 70-72.
- Han, J., et Kamber, M. (2006). Data mining : concepts and techniques. *Morgan Kaufmann, 2*.
- Han, J., Pei, J., et Yin, Y. (2000). Mining frequent patterns without candidate generation. Dans A. Press (dir.), *Weidong chen, jeffrey naughton, and philip a. bernstein, editors, 2000 acm sigmod intl. conference on management of data* (p. 1-12).
- Han, J., Tseng, C. W., et Keleher, P. (1997). Reducing synchronization overhead for compiler-parallelized codes on software dsms. Dans *Languages and compilers for parallel computing*.
- Harding, J. A., Shahbaz, M., et Kusiak, A. (2005). Data mining in manufacturing : A review. *Journal of Manufacturing Science and Engineering - transactions of the ASME, 128(4)*, 969-976.
- Hauglustaine-Charlier, B. (1989). *Images pour apprendre à programmer*. Archive-EduTice.
- Heinich, R., Molenda, M., Russekk, J. D., et Smaldino, S. E. (2002). *Instructional media and technologies for learning*. (7e Ed.). Addison-Wesley.
- Hellerstein, J. M. (2008). *Quantitative data cleaning for large databases*. United Nations Economic Commission for Europe (UNECE).
- Hipp, J., Gntzer, U., et Nakhaeizadeh, G. (2000). Algorithms for association rule mining - a general survey and comparison. *SIGKDD Explorations, 2*, 58-64.
- Host, V. (1991). Le rapport à l'expérimental dans la physique de l'enseignement secondaire. *Aster, Paris INRP, 2(8)*, 29-54.

- Houssaye, J. (1988). Le triangle pédagogique. Dans Berne (dir.), *Coll. exploration : « pédagogie : Histoire et pensée »* (p. 454). Peter Lang.
- Houssaye, J. (1993). *La pédagogie : une encyclopédie pour aujourd'hui*. ESF Editeur.
- Houtsma, M., et Swami, A. (1995). Set-oriented mining for association rules in relational databases. Dans I. C. S. Press (dir.), *P. s. yu and a. l. p. chen, editors, proceedings of the 11th international conference on data engineering* (p. 25-34). Los Alamitos, CA, USA.
- Huang, J. J., Tzeng, G. H., et Ong, C. S. (2007). Marketing segmentation using support vector clustering. *Expert Systems with Applications*, 32(2), 313-317.
- Jonnaert, P. (1996). Apprentissages mathématiques en situation : une perspective constructiviste. *Revue des sciences de l'éducation*, 22(2), 233-252.
- Jonnaert, P. (2009). *Compétence et socioconstructivisme : un cadre théorique*. Bruxelles : De Boeck Université (2e éd.).
- Jonnaert, P., et V-Borghet, C. (2003). *Créer des conditions d'apprentissage : un cadre de référence socioconstructiviste pour une formation didactique des enseignants*. Bruxelles : De Boeck Université.
- Kamsu-Foguem, B., et Mathieu, Y. (2014). Software architecture knowledge for intelligent light maintenance. *Advances in Engineering Software*, 67, 125-135.
- Knuth, D. E. (1973). *The art of computer programming, volume i : Fundamental algorithms*. (2e éd.). Addison-Wesley.
- Kodratoff, Y. (1999). Quelques contraintes symboliques sur le numérique en ECD et ECT. Dans *Sfds* (p. 183-188). Grenoble, France.
- Kodratoff, Y., Napoli, A., et Zighed, D. (2001). Extraction de connaissances dans des bases de données. Dans *Bulletin de l'association française de l'intelligence artificielle*.
- Köksal, G., Batmaz, I., et Testik, M. (2011). A review of data mining applications for quality improvement in manufacturing industry. *Expert Systems with Applications*, 38(10), 13448-13467.

- Kuutti, K. (1996). Activity theory as a potential framework for human computer interaction research. Dans *Nardi, b. a. (ed.), context and consciousness : Activity theory and human-computer interaction* (p. 17-44). Cambridge, MA : The MIT Press.
- Lacombe, D. (1989). La didactique des disciplines,. Dans P. . E. Universalis (dir.), *P.-f. baumberger (dir.), encyclopaedia universalis* (Vol. Tome 7, p. 393-396).
- Ladjili, T. (2009). La didactique des disciplines . [Manuel de logiciel]. Récupéré de <http://pf-mh.uvt.rnu.tn/64/1/didactique-disciplines.pdf>
- Lahrer, A. (1991). *Implication statistique et application à l'analyse de démarches de preuve mathématique*. (Thèse de doctorat). Université Rennes I.
- Lallich, S., et Teytaud, O. (2004). Evaluation et validation des mesures d'intérêt des règles d'association. *RNTI-E-1, spécial*, 193-217.
- Langlade, G. (1997). Essai de délimitation du champ propre d'une didactique de discipline : l'exemple des textes littéraires. *Didactiques Recherches et pratiques, Les cahiers du CeRF n5*(1er semestre), 1-25.
- Larose, D. T. (2005). *Discovering knowledge in data : an introduction to data mining*. Wiley- Interscience.
- Lavrac, N., Flach, P., et Zupan, B. (1999). Rule evaluation measures : A unifying view. Dans e. S Dzeroski et P Flach (dir.), *Ninth international workshop on inductive logic programming (ilp99)* (Vol. 1634 of Lecture Notes in Artificial Intelligence-Springer Verlag, p. 174 - 185).
- Lee, D. H., et Kim, M. (1997). Database summarization using fuzzy isa hierarchies. *IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics*, 27(1), 68-78.
- Legendre, R. (1993). *Dictionnaire actuel de l'éducation* . (2ème ed.). Montréal/Paris : Guérin/Eska.
- Lenca, P., Meyer, P., Picouet, P., Vaillant, B., et Lallich, S. (2003). Critères d'évaluation des mesures de qualité des règles d'association. *Revue des Nou-*

- velles Technologies de l'information (RNTI), RNTI-1, 123-134. Récupéré de <https://hal.archives-ouvertes.fr/hal-00632772>
- Lenca, P., Meyer, P., Vaillant, B., Picouet, P., et Lallich, S. (2004). Evaluation et analyse multi-critères des mesures de qualité des règles d'association. *Revue des Nouvelles Technologies de l'information (RNTI)*, RNTI-E-1, 219-246. Récupéré de <https://hal.archives-ouvertes.fr/hal-00632772>
- Leontiev, A.-N. (1975). *Activité, conscience, personnalité*. Trad. Française, Moscou, Editions du progrès.
- Leontiev, A.-N. (1978). *Activity, consciousness, and personality*. Cliffs, NJ : Prentice Hall.
- Leontiev, A.-N. (1981). Psychology and the language learning process. , 21-28.
- Lerman, I. C. (1970). *Les bases de la classification automatique*. Paris, Gauthier-Villars.
- Lerman, I. C. (1981). *Classification et analyse ordinale des données*. Dunod.
- Lerman, I. C. (1983). Sur la signification des classes issue d'une classification automatique. *Springer-Verlag In J. Felsenstein, esitor, Numerical Taxonomy*, 179-198.
- Lerman, I. C. (1984). Justification et validité statistique d'une échelle $[0, 1]$ de fréquence mathématique pour une structure de proximité sur un ensemble de variables observées. Dans *Rapport de recherche inria*. Centre de Rennes IRISA.
- Lerman, I. C. (1989). Formules de réactualisation en cas d'agrégation multiples. *RAIRO, série RO*, 23, 151-163.
- Lerman, I. C. (2008). Analyse logique, combinatoire et statistique de la construction d'une hiérarchie binaire implicative ; niveaux et nœuds significatifs. *Math. Sci. hum. / Mathematics and Social Sciences*, 4(184), 47-107.
- Lerman, I. C., et Ghazzali, N. (1991). What do we retain from a classification tree. Dans E. Diday (dir.), *Y. lechevallier* (p. 27-42). Symbolic-Numeric data analysis and learning, Nova Science.

- Lerman, I. C., Gras, R., et Rostam, H. (1981). Elaboration et évaluation d'un indice d'implication pour des données binaires. Dans *Math sc. hum* (Vol. 74, p. 5-35).
- Lewis, R. (1998). Apprendre conjointement : une analyse, quelques expériences et un cadre de travail. Récupéré de <http://edutice.archives-ouvertes.fr/edutice-00000486>
- Liao, S. H. (2003). Knowledge management technologies and applications - literature review from 1995 to 2002. *Expert Systems with Applications*, 25(2), 155-164.
- Linard, M. (1990). *Des machines et des hommes. apprendre avec les nouvelles technologies*. Paris : Edition universitaire, Savoir et Formation.
- Linard, M. (1994). Vers un sujet narratif de la connaissance dans les modélisations de l'apprentissage. *Intellectica ARC, Paris*, 2(19), 117-165.
- Loevinger, J. (1947). A symmetric approach to the construction and evaluation of tests of ability. *Psychological Monographs*, 61, 1-49.
- Lombard, F. (2003). Du triangle de houssaye au tétraèdre des tic : Comment l'analyse des productions TIC permet d'approcher une compréhension des interactions entre les savoirs d'expérience et de recherche. *Paper presented at the Colloque REF 03*.
- Lui, B., Hsy, W., Chen, S., et Ma, Y. (2000). Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems*, 15(5), 47-55.
- Luria, A. (1976). *The cognitive development : Its cultural and social foundations*. Cambridge, MA : Harvard University Press.
- Mangenot, F. (1996). L'apprenant, l'enseignant et l'ordinateur : un nouveau triangle didactique. Dans V. d'Aoste (dir.), *Actes du congrès "linguaggi della formazione : l'informatica"*, organisé par l'IRRSAE. Saint-Vincent.
- Mangenot, F. (1999). Vers les campus virtuels. principes et fondements technosémiopragmatiques des dispositifs de formation virtuels. Dans C. Editions (dir.), *G. j. e. l. montoyer (ed.), le dispositif. entre usage et concept* (Vol. num spécial, p. 153-168). Paris.

- Mannila, H., Toivonen, H., et Verkamo, A. I. (1995). Efficient algorithms for discovering association rules. Dans A. Press (dir.), *Usama m. fayyad and ramasamy uthurusamy, editors, aaai workshop on knowledge discovery in databases (kdd-94)* (p. 181-192). Seattle, Washington.
- Marban, O., Segovia, J., Menasalvas, E., et Fernandez-Baizan, C. (2009). Toward data mining engineering : A software engineering approach. *Information Systems*, 34(1), 87-107.
- Martinand, J.-L. (1996). Introduction à la modélisation. Dans *Actes du séminaire de didactique des disciplines technologiques*. ENS de Cachan.
- Martinand, J.-L. (2006). Algorithmic thinking : The key for understanding computer science. Dans L. N. in Computer Science Springer (dir.), *International conference on informatics in secondary schools - evolution and perspectives* (Vol. 4226, p. 159-168).
- Meirieu, P. (1988). Apprendre, ... oui, mais comment ? *Revue française de pédagogie*, 83.
- Meirieu, P. (2005 [ref. du 06-05-2018]). Cours et outils de formation : Petit dictionnaire de pédagogie-didactique. *Site de Philippe Meirieu Histoire et actualité de la pédagogie.*, <http://www.meirieu.com/DICTIONNAIRE/didactique.htm>.
- Meurist, A. (2014). *Introduire des éléments d'algorithmique dans un cours de mathématiques : une expérience dans l'enseignement secondaire belge* (Mémoire master en sciences mathématiques). Université de Mons, Place du Parc 20 - B-7000, Mons.
- Meyers, E. (2007). From activity to learning : using cultural historical activity theory to model school library programmes and practices. *Information Research*, 12(3).
- Miettinen, R. (1997). The concept of activity in the analysis of heterogeneous networks in innovation process. actor network and after. *Centre for Social Theory and Technology Workshop : University of Helsinki*. Retrieved on February 01 2011. Récupéré de <http://communication.ucsd.edu/MCA/Paper/Reijo/Reijo.html#Introduction>

- Miller, R., et al., Y. Y. (1997). Association rules over interval data. Dans *Proceedings of the acm sigmod international conference on management of data* (p. 452-461).
- Modeste, S. (2012). *Enseigner l'algorithme pour quoi ? quelles nouvelles questions pour les mathématiques ? quels apports pour l'apprentissage de la preuve ?*. (Thèse de doctorat). Université de Grenoble.
- Moore, E. (1909). On a form of general analysis with applications to linear differential and integral equations. Dans *Atti del iv congresso internazionale dei matematici* (p. 98-114). Roma.
- Muratet, M. (2012). *Conception, réalisation et évaluation d'un jeu sérieux de stratégie temps réel pour l'apprentissage des fondamentaux de la programmation*. (Thèse de doctorat). Université Paul Sabatier- Toulouse III.
- Nakache, J., et Confais, J. (2005). Approche pragmatique de la classification. *TECHNIP*.
- Nardi, B. (1996). *Context and consciousness : Activity theory and human-computer interaction*. Cambridge : MIT Press.
- Ngai, E. W. T., Xiu, L., et Chau, D. C. K. (2009). Application of data mining techniques in customer relationship management : A literature review and classification. *Expert Systems with Applications*, 36(2), 2592-2602.
- Nguifo, E. M., Grissa, D., et Guillume, S. (2012). Catégorisation des mesures d'intérêt pour l'extraction des connaissances. Dans D. A. Z. et Gilles Venturini (dir.), *Revue des nouvelles technologies de l'information* (p. 117-143).
- Norman, D. A., et Lindsay, P. H. (1977). *Human information processing*. Academic Press.
- Öre, O. (1944). Galois connexion. *Transactions of the American Mathematical Society*, 55.
- Owen, C. (2008). Analyser le travail conjoint entre différents systèmes d'activités. *Revue Activités*, 5(2), 70-89.

- Papert, S. (1993). *The children's machine : Rethinking school in the age of the computer*. Basic Books ; Reprint edition. (ISBN-13 : 978-0465010639)
- Paquelin, D. (2002). *Eléments pour une problématisation de l'analyse de dispositif de formation en ligne*. L'Harmattan.
- Park, J. S., Chen, M. S., et Yu, P. S. (1995). An effective hash based algorithm for mining association rules. Dans M. J. Carey et D. A. Schneider (dir.), *Proceedings of the 1995 acm sigmod international conference on management of data* (p. 175-186). San Jose, California, USA.
- Park, J. S., Chen, M. S., et Yu, P. S. (1997). Using a hash-based method with transaction trimming for mining association rules. *IEEE Trans. Knowledge and Data Eng.*, 9(5), 813-824.
- Pasquier, N. (2000). *Data mining : Algorithmes d'extraction et de réduction des règles d'association dans les bases des données* (Thèse de doctorat). Université Clermont-Ferrand II, FRANCE.
- Pea, R. D., et Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New Ideas Psychol.*, 2(12), 137-168.
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems : Networks of plausible inference*. Morgan Kaufmann.
- Pelgrum, W. J., et Law, N. (2004). *Les tic et l'éducation dans le monde : tendances, enjeux et perspectives*. UNESCO, Institut international de planification de l'éducation , Paris.
- Perkins, D. N. (2005). L'individu-plus : une vision distribuée de la pensée et de l'apprentissage. *Revue Française de Pédagogie*, 111, 57-71.
- Piatetsky-Shapiro, G. (1991). Discovery, analysis and presentation of strong rules. Dans *Knowledge discovery in databases* (p. 229-248). Cambridge, Mass, 1991 AAAI/MIT Press.
- Piatetsky-Shapiro, G., et Frawley, W. J. (1991). Knowledge discovery in databases. *AI Magazine*, 11.

- Press, G. (2013). Data science : What's the half-life of a buzzword?. *Forbes*. Récupéré de <https://www.forbes.com/sites/gilpress/2013/08/19/data-science-whats-the-half-life-of-a-buzzword/#4ce387547bfd> (accédé le 26/09/2018)
- Rahm, E., et Do, H. (2000). Data cleaning : Problems and current approaches. *IEEE Bulletin on Data Engineering*, 23(4), 3-13.
- Rakotomalala, H. F., Ralahady, B., et Totohasina, A. (2018). A Novel Cohesitive Implicative Classification based on M_{GK} and application on diagnostic on informatics literacy of students of higher education in Madagascar. Dans X.-S. Yang, S. Sherratt, N. Dey, et A. Joshi (dir.), *Third international congress on information and communication technology, icict 2018* (Springer éd., Vol. 797, p. 161-174). London, UK. doi : 10.1007/978-981-13-1165-9
- Rakotomalala, H. F., et Totohasina, A. (2018). An efficient new cohesion indice based on the quality measure of association rules M_{GK} . Dans X.-S. Yang, N. Dey, et A. Joshi (dir.), (IEEE Xplorer digital library éd., Vol. *, p. 28-35). London, UK.
- Rakotomalala, H. F., et Totohasina, A. (2019). On Hierarchical Classification Implicative and Cohesitive M_{GK} -based : application on analysis of the computing curricula and students abilities according the Anglo-Saxon model. Dans J. Kacprzyk (dir.), (Springer éd., Vol. 798, p. 16-21). London, UK.
- Rakotomalala, H. F., Totohasina, A., et Diatta, J. (2017a). Extraction des règles d'associations M_{GK} -valides avec contribution de support. *Actes des 24èmes rencontres de la Société Francophone de Classification SFC 2017*, 29-32.
- Rakotomalala, H. F., Totohasina, A., et Diatta, J. (2017b). Une mesure de cohésion basée sur la mesure de qualité des règles d'association M_{GK} . *Actes des 24èmes rencontres de la Société Francophone de Classification SFC 2017*, 21-24.
- Rakotomalala, H. F., Totohasina, A., et Diatta, J. (2018). Classification des mesures des règles d'association selon la CHIC- M_{GK} . *Actes des 25èmes Rencontres de la Société Francophone de Classification - SFC 2018*, 13-16.

- Rakotomalala, H. F., Totohasina, A., et Sambatra, E. J. (2012). Etude sur les stratégies à mettre en œuvre pour l'intégration effective des TIC dans la pratique pédagogique de l'enseignement supérieur technique de la région DIANA : Cas Antsiranana ville. *Actes du colloque international sur les TIC. juin 2012 - ENS Ampefiloha Antananarivo, TIC pour enseigner, TIC pour apprendre : Quelles stratégies ? Quelles perspectives ?*.
- Ramanantsoa, H. (2016). *Contribution à l'amélioration des générations des bases des règles d'association mgk-valides et applications en didactiques des mathématiques*. (Thèse de doctorat). Université d'Antananarivo.
- Ratsimba-Rajohn, H. (1992). *Contribution à l'étude de la hiérarchie implicative : Application à l'analyse de la gestion didactique des phénomènes d'ostension et de contradictions* (Thèse de doctorat). U.F.R. de Mathématiques, Université de RENNES I, France.
- Renaud, J., Bonjour, E., Chebel-Morello, B., Fuchs, B., et Matta, N. (2008). *Retour et capitalisation d'expérience - outils et démarches*. AFNOR Editions, Paris.
- Rencher, A. C. (2003). *Methods of multivariate analysis*. Wiley.
- Reuter, Y., Cohen-Azria, C., Daunay, B., Delcambre-Delville, I., et Lahanier, D. (2007). *Dictionnaire des concepts fondamentaux des didactiques*. Bruxelles : De Boeck.
- Revuz, A. (1980). *Est-il impossible d'enseigner les mathématiques ?* Paris, PUF.
- Robert, A., et Rogalski, J. (2002). Le système complexe et cohérent des pratiques des enseignants de mathématiques : une double approche. *Canadian Journal of Math, Science Technology Education*, 2(4), 505-528.
- Roberto, J., et Bayardo, J. (1998). Efficiently mining long patterns from databases. Dans A. Press (dir.), *Laura m. haas and ashutosh tiwary, editors, sigmod 1998, proceedings acm sigmod international conference on management of data, june 2-4, 1998* (p. 85-93). Seattle, Washington, USA.
- Robitaille, M. (2007). Quand un dispositif de développement professionnel sur l'apprentissage coopératif devient un lieu de régulations entre pairs. Dans *Régulation des apprentissages en situation scolaire et en formation* (p. 171-189). De boeck.

- Rogalski, J. (1985). Alphabétisation informatique, problèmes conceptuels et didactiques. *Bulletin de l'APMEP*, W 347, 61-74.
- Rogalski, J. (2012). Théorie de l'activité et didactique, pour analyse conjointe des activités de l'enseignant et de l'élève. *Jornal Internacional de Estudos em Educação Matemática*, 5(1).
- Romainville, M. (1988). *Une analyse critique de l'initiation à l'informatique. quels apprentissages et quels transferts ?*
- Sahami, M., Danyluk, A., Fincher, S., Fisher, K., Grossman, D., Hawthorne, E., ... Thompson, A. (2013). *Curriculum guidelines for undergraduate degree programs in computer science, CS 2013*. (Rapport technique). ACM Press. and IEEE Computer Society Press. doi : 10.1145/2534860
- Savasere, A., Omiecinski, E., et Navathe, S. B. (1995). An efficient algorithm for mining association rules in large databases. Dans M. K. P. Inc. (dir.), *Vldb '95 : Proceedings of the 21th international conference on very large data bases* (p. 432-444). San Francisco, CA, USA.
- Schneider, D. (2003). Socioconstructivist scenarios with the internet for secondary and higher education. Dans Lucerne (dir.), *5th international conference on new educational environments : The knowhow hub for blended learning*.
- Schön, D. A. (1994). *Le praticien réflexif, à la recherche du savoir caché dans l'agir professionnel*. Montréal, Canada : Montréal : Logiques.
- Shackelford, R., Cross, J., Davies, G., Impagliazzo, J., Kamali, R., LeBlanc, R., ... Topi, H. (2005). *The overview report covering undergraduate degree programs in CE-CS-IS-IT-SE, CC 2005*, (Rapport technique). New York : ACM Press. and IEEE Computer Society Press. doi : 1-59593-359-X
- Shannon, C. (1948). A mathematical theory of communication. *s.l. : American Telephone and Telegarph Co., Vol. 27*, 379-423 ;623-656.
- Shintani, T., et Kitsuregawa, M. (1996). Rash based parallel algorithms for mining association rules. Dans I. C. S. T. C. on Data Engineering et A. SIGMOD (dir.), *Pdis : International conference on parallel and distributed information systems*.

- Shortliffe, E., et Buchanan, B. (1975). A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23, 351-379.
- Srikant, R., et Agrawal, R. (1997). Mining generalized association rules. *Future Generation Computer Systems*, 13, 161-180.
- Sésamath. (2016, mai). Le manuel de cycle 4 (Magnard éd.) [Manuel de logiciel]. (ISBN : 978-2-210-10634-5)
- Suzuki, E. (1997). Autonomous discovery of reliable exception rules. Dans *Third international conference on knowledge discovery and data mining*.
- Suzuki, E., et Kodratoff, Y. (1998). Discovery of surprising exception rules based on intensity of implication. Dans e. Springer-Verlag (dir.), *Principles of data mining and knowledge discovery (pkdd)*.
- Szasz, G. (1971). *Théorie des treillis*. Dunod.
- Tan, P. N., Kumar, V., et Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. Dans *Actes de 8th acm intl. conf. on knowledge discovery and data mining, kdd'02*, (p. 31-42).
- Tan, P. N., Kumar, V., et Srivastava, J. (2004). Selecting the right interestingness measure for association analysis. *Information Systems*, 29(4), 293-313.
- Taouil, R., Pasquier, N., Bastide, Y., et Lakhal, L. (1999). Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1), 25-46.
- Taurisson, A. (2005). *La pédagogie de l'activité, un nouveau paradigme ?*. (Thèse de doctorat). Université Lumière Lyon II.
- Toivonen, H. (1996). Sampling large databases for association rules. Dans C. M. T. M. Vijayaraman Alejandro P. Buchmann et N. L. Barda (dir.), *Int. conf. very large data bases* (p. 134-145). San Jose, California : Morgan Kaufman.
- Totohasina, A. (2003). Normalisation de mesures probabilistes de la qualité des règles. Dans *Sfds'03, xxxv^{ème} journées de statistiques* (p. 985-988). Lyon 2, France.

- Totohasina, A. (2008). *Contribution à l'étude des mesures de la qualité des règles d'association : normalisation sous cinq contraintes et cas de m_{GK} : propriétés, bases composites des règles et extension en vue d'applications en statistique et en sciences physiques* (HDR). Université d'Antsirananana.
- Totohasina, A., et Feno, D. (2008). De la qualité des règles d'association : étude comparative des meures mgk et confiance. *Actes du 9ème colloque Africain sur la recherche en Informatique et Mathématiques Appliquées, CARI-2008*, 561-568.
- Totohasina, A., et Ralambondrainy, H. (2005). Ion : a pertinent new measure for mining information from many types of data. Dans *Sitis* (p. 202–207).
- Totohasina, A., Ralambondrainy, H., et Diatta, J. (2004). Une vision unificatrice des mesures de la qualité des règles d'association booléennes et un algorithme efficace d'extraction des règles d'association implicite.
- Vergnaud, G. (1983). Didactique et acquisition du concept. *Recherches en Didactique des Mathématiques, n spécial(4)*.
- Voulgre, E. (2011). *Une approche systémique des TICE dans le système scolaire français : entre finalités prescrites, ressources et usages par les enseignants*. (Thèse de doctorat). Université de Rouen.
- Vygotsky, L. (1934). *Pensée et langage*. Trad. 1985 Paris : Editions Sociales.
- Wille, R. (1982). Restructuring lattice theory : an approach based on hierarchies of concepts. Dans R. I (dir.), *Ordered sets* (p. 445-470). Dordrecht-Boston, Ridel.
- Wille, R. (1983). Subdirect decomposition of concept lattices. Dans A. universalis (dir.), (Vol. 17, p. 275-287). Dordrecht-Boston, Ridel.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3).
- Wu, X., Zhang, C., et Zhang, S. (2004). Efficient mining of both positive and negative association rules. éd. *ACM Transactions on Information Systems (TOIS)*. New York : s.n., 2004, Vol. 22, Issue 3, July 2004 ISSN : 1046-8188, 381-405.
- Zaki, M. J. (1999). Parallel and distributed association mining : A survey. *IEEE Concurrency*, 7(4), 14-25.

- Zaki, M. J., et Hsiao, C. J. (2002). An efficient algorithm for closed itemset mining. Dans *Actes de second siam international conference on data mining* (p. 1 - 17). Arlington.
- Zaki, M. J., Parthasarathy, S., Ogihara, M., et Li, W. (1996). *Parallel data mining for association rules on shared-memory multiprocessors*. (Rapport technique N° TR618).
- Zaki, M. J., Parthasarathy, S., Ogihara, M., et Li, W. (1997). *New algorithms for fast discovery of association rules*. (Rapport technique N° TR651).
- Zighed, D. A., et Rakotomalala, R. (2000). Graphes d'induction : apprentissage et data mining. *Hermes*.
- Zighed, D. A., et Rakotomalala, R. (2002). Extraction de connaissances à partir de données ECD. *Base de données, H3744*.

ANNEXES

Annexe A

Quelques algorithmes d'extraction des règles d'association

Algorithme 5 Algorithme AprioriTid

```

1 : Calculer  $L_1$ 
2 :  $\hat{C}_k \leftarrow D$ 
3 :  $k \leftarrow 2$ 
4 : tant que  $L_{k-1} \neq \emptyset$  faire
5 :    $C_k \leftarrow \text{apriori} - \text{gen}(L_{k-1})$ 
6 :    $\hat{C}_k \leftarrow \phi$ 
7 :   tant que  $t \in C_{k-1}$  faire
8 :      $C_k = \{c \in C_k \mid (c[1].c[2]..c[k-1]) \in t.\text{ensemble} \wedge (c[1].c[2]..c[k-2].c[k]) \in t.\text{ensemble}\}$ 
9 :     tant que  $c \in C_t$  faire
10 :        $c.\text{count}++$ 
11 :     fin tant que
12 :     si  $C_t \neq \emptyset$  alors
13 :        $\hat{C}_k \leftarrow + \langle t.TID, C_t \rangle$ 
14 :     fin si
15 :   fin tant que
16 :    $L_k \leftarrow \{c \in C_k \mid c.\text{count} \geq \text{minSup}\}$ 
17 :    $k \leftarrow k + 1$ 
18 : fin tant que
19 : Retourner  $\cup_k L_k$ 

```

Algorithme 6 Algorithme SAMPLING

Entrée : Une base \mathcal{B} ; un seuil minsup ; un seuil diminuée de λ ; un échantillon n_e .

Sortie : Ensemble L_k des k-itemsets fréquents.

```

1 :  $E_a \leftarrow$  Echantillon aléatoire de taille  $n_e$ 
2 :  $S_g \leftarrow \text{Sampling} - \text{Gen}(E_a, \lambda)$ 
3 : Calculer  $\mathcal{B}d^-(S_g)$ 
4 :  $C \leftarrow S_g \cup \mathcal{B}d^-(S_g)$ 
5 :  $S_c \leftarrow \text{Sampling} - \text{Count}(\mathcal{B}, C, \text{minsup})$ 
6 : si  $S_c \cap \mathcal{B}d^-(S_g) \neq \emptyset$  alors
7 :   message "Erreur possible"
8 : fin si
9 : Retourner  $S_c$ 

```

Algorithme 7 Algorithme Sampling-Gen

Entrée : Echantillon E_a du contexte ; seuil minimal du support diminuée de λ ;**Sortie :** Ensemble S_k des k-itemsets fréquents.

```

1 :  $C_1 \leftarrow \{\{i\} | i \in \mathcal{I}\}$ 
2 :  $k \leftarrow 1$ 
3 : tant que ( $C_k \neq \emptyset$ ) faire
4 :    $S_k \leftarrow \text{Sampling} - \text{Count}(E_a, C_k, \lambda)$ 
5 :    $C_{k+1} \leftarrow \text{Apriori} - \text{Gen}(S_k)$ 
6 :    $k++$ 
7 : fin tant que
8 : Retourner  $\cup_k S_k$ 

```

Algorithme 8 Algorithme Sampling-Count

Entrée : Ensemble de transaction τ ; Ensemble d'itemsets candidats C ; valeur minimale du support *valsupp***Sortie :** Ensemble des k-itemsets fréquents dans τ selon *valsupp*.

```

1 : pour tout (item  $i \in \mathcal{I}$ ) faire
2 :    $i.\text{contd} \leftarrow \emptyset$ 
3 :   pour tout (candidat  $c \in C$ ) faire
4 :     pour tout (item  $i \in c$ ) faire
5 :        $i.\text{contd} \leftarrow i.\text{contd} \cup \{c\}$ 
6 :     fin pour
7 :   fin pour
8 :   pour tout (candidat  $c \in C$ ) faire
9 :      $c.\text{count} \leftarrow t$ 
10 :    pour tout (objet  $t \in \tau$ ) faire
11 :      pour tout (candidat  $c \in C$ ) faire
12 :         $c.\text{itemcount} \leftarrow t$ 
13 :        pour tout (item  $i \in \tau$ ) faire
14 :          pour tout (candidat  $c \in i.\text{contd}$ ) faire
15 :             $c.\text{itemcount}++$ 
16 :            si ( $c.\text{itemcount} = |c|$ ) alors
17 :               $c.\text{count}++$ 
18 :            fin si
19 :          fin pour
20 :        fin pour
21 :      fin pour
22 :    fin pour
23 :  fin pour
24 : pour tout (item  $i \in \mathcal{I}$ ) faire
25 :   Retourner  $\cup\{c, c.\text{support} | c \in C \wedge c.\text{count}/|\tau| \geq \text{valsupp}\}$ 

```

Algorithme 9 Algorithme PARTITION**Entrée :** Contexte \mathcal{B} ; seuil minimal de support $minsupp$; nombre de partition n ;**Sortie :** Ensemble F^g des itemsets fréquents.

```

1 : Partitionner( $\mathcal{B}, n$ )
2 : pour tout ( $r \leftarrow 1; r \leq n; r++$ ) faire
3 :   Lire partition  $P_r$ 
4 :    $F^r \leftarrow Partition - Gen(P_r, minsupp)$ 
5 : fin pour
6 : pour tout ( $r \leftarrow 1; F_k^r \neq emptyset; k++$ ) faire
7 :    $C_k^g \leftarrow \bigcup_{r=1}^{r=n} F_k^r$ 
8 : fin pour
9 : pour tout ( $r \leftarrow 1; r \leq n; r++$ ) faire
10 :   Lire partition  $P_r$ 
11 :    $F^r \leftarrow Partition - Count(C^g, P_r)$ 
12 : fin pour
13 :  $F^g \leftarrow \{c \in C^g | c.support \geq minsupp\}$ 
14 : Retourner  $F^g$ 

```

Algorithme 10 Algorithme Partition-Gen**Entrée :** Partition P_r du contexte \mathcal{B} ; seuil $minsupp$;**Sortie :** Ensemble F^r des itemsets fréquents locaux.

```

1 :  $F_1^{P_r} \leftarrow \{1\text{-itemsets fréquents dans } P_r \text{ avec liste de TID}\}$ 
2 : pour tout ( $k \leftarrow 2; F_k^{P_r} \neq \emptyset; k++$ ) faire
3 :   pour tout (itemsets  $p \in F_{k-1}^{P_r}$ ) faire
4 :     pour tout (itemsets  $q \in F_{k-1}^{P_r}$ ) faire
5 :       si ( $p[1] = q[1] \wedge \dots \wedge p[i-2] = q[i-2] \wedge p[i-1] < q[i-1]$ ) alors
6 :          $c \leftarrow p[1] \cup p[2] \cup \dots \cup p[i-1] \cup q[i-1]$ 
7 :         si ( $\forall$  sous-ensemble de  $|c| - 1$  nous avons  $s \in F_{k-1}^{P_r}$ ) alors
8 :            $s.tidset \leftarrow p.tidset \cap q.tidset$ 
9 :           si ( $|c.tidset| / |P_r| \geq minsupp$ ) alors
10 :             $F_k^{P_r} \leftarrow F_k^{P_r} \cup \{c, c.support\}$ 
11 :          fin si
12 :        fin si
13 :      fin si
14 :    fin pour
15 :  fin pour
16 : fin pour
17 : Retourner  $\bigcup F_k^{P_r}$ 

```

Algorithme 11 Algorithme Partition-Count

Entrée : Partition C^g de candidats globaux ; partition P_r de la base \mathcal{B} ;**Sortie :** Champs du support de C^g mis à jour

```

1 : pour tout (1-itemset  $c_1 \in C_1^g$ ) faire
2 :    $c_1.tidset \leftarrow \{t.TID \mid t \in P_r \text{ et } c_1 \text{ contenu dans } t\}$ 
3 :    $c_1.support \leftarrow c_1.support + |c_1.tidset|$ 
4 : fin pour
5 : pour tout ( $k \geq 2; C_k^g \neq \emptyset; k++$ ) faire
6 :   pour tout (k-itemset  $c_1 \in C_k^g$ ) faire
7 :      $tempset \leftarrow c_1.tidset \cap c_2.tidset \cap \dots \cap c_k.tidset$ 
8 :      $c.support \leftarrow c.support + |tempset|$ 
9 :   fin pour
10 : fin pour

```

Algorithme 12 Algorithme ECLAT

Entrée : un ensemble de motifs \mathcal{I} ; une base \mathcal{B} ; un $minsupp$ **Sortie :** $\mathcal{F}[\mathcal{I}]$, un ensemble des itemsets fréquents.

```

1 :  $\mathcal{F}[\mathcal{I}] \leftarrow \{\}$ 
2 : pour tout  $i \in \mathcal{I}$  occurring in  $\mathcal{B}$  faire
3 :    $\mathcal{F}[\mathcal{I}] \leftarrow \mathcal{F}[\mathcal{I}] \cup \{\mathcal{I} \cup \{i\}\}$ 
4 :    $\mathcal{B} \leftarrow \{\}$ 
5 :   pour tout  $j \in \mathcal{I}$  occurring in  $\mathcal{B}$  such that  $j > i$  faire
6 :      $C \leftarrow cover(\{i\}) \cap cover(\{j\})$ 
7 :     si  $|C| \geq minsupp$  alors
8 :        $\mathcal{B}^i \leftarrow \mathcal{B}^i \cup \{j, C\}$ 
9 :     fin si
10 :   fin pour
11 :   compute  $\mathcal{F}[\mathcal{I} \cup \{i\}](\mathcal{B}^i, minsupp)$ 
12 :    $\mathcal{F}[\mathcal{I}] \leftarrow \mathcal{F}[\mathcal{I}] \cup \mathcal{F}[\mathcal{I} \cup \{i\}]$ 
13 : fin pour

```

Algorithme 13 Algorithme FP-Growth(\mathcal{B} , minsupp)**Entrée :** Set of items \mathcal{I} ; Database \mathcal{B} ; minimum support *minsupp*.**Sortie :** F-List \mathcal{F}

```

1 : Define and clear F-list  $\mathcal{F}$ 
2 : pour tout transaction  $t_i \in \mathcal{B}$  faire
3 :   pour tout Items  $a_i \in t_i$  faire
4 :      $\mathcal{F}[a_i] ++$ 
5 :   fin pour
6 : fin pour
7 : Sort  $\mathcal{F}$ 
8 : Define and clear the root of FP-Tree :  $r$ 
9 : pour tout transaction  $t_i \in \mathcal{B}$  faire
10 :   Make  $t_i$  ordered according to  $\mathcal{F}$ 
11 :   Call ConstructTree( $t_i, r$ )
12 : fin pour
13 : pour tout transaction  $a_i \in \mathcal{I}$  faire
14 :   Growth( $r, a_i, minsupp$ )
15 : fin pour

```

Algorithme 14 Algorithme FP-Tree(\mathcal{B} , a, minsupp)**Entrée :** Set of items \mathcal{I} ; Database \mathcal{B} ; minimum support *minsupp*.**Sortie :** FP-Tree : Frequent-Pattern-Tree of \mathcal{B}

```

1 : si contains a single path  $Z$  alors
2 :   pour tout combination (denoted as  $\gamma$ ) of the node in  $Z$  faire
3 :     Generate pattern  $\beta = \gamma \cup a$ 
4 :     si  $\beta.support > minsupp$  alors
5 :       Call Output( $\beta$ )
6 :     fin si
7 :   fin pour
8 : sinon
9 :   pour tout  $b_i \in r$  faire
10 :     Generate pattern  $\beta = b_i \cup a$  with  $support = b_i.support$ 
11 :     si  $\beta.support > minsupp$  alors
12 :       Call Output( $\beta$ )
13 :     fin si
14 :     si  $Tree_\beta \neq \emptyset$  alors
15 :       Call Growth( $Tree_\beta, \beta, minsupp$ )
16 :     fin si
17 :   fin pour
18 : fin si

```

Algorithme 15 Algorithme Max-Miner

Entrée : Contexte \mathcal{B} ; seuil minimal support $minsupp$.**Sortie :** Ensemble F_m des itemsets fréquents maximaux

```

1 :  $C \leftarrow \emptyset$ ;
2 :  $F_m \leftarrow Gen - Initial - Groups(\mathcal{B}, C, minsupp)$ ;
3 : tant que  $C \neq \emptyset$  faire
4 :   lire Contexte  $\mathcal{B}$ 
5 :    $support - Count(\mathcal{B}, C)$ 
6 :   pour tout candidat  $c \in C | h(t) \cup t(c)$  est fréquent faire
7 :      $F_m \leftarrow F_m \cup \{h(c) \cup t(c)\}$ 
8 :   fin pour
9 :    $C_{new} \leftarrow \emptyset$ 
10 :  pour tout candidat  $c \in C | h(t) \cup t(c)$  est infrequent faire
11 :     $F_m \leftarrow F_m \cup Gen - Sub - Nodes(c, C_{new}, minsupp)$ 
12 :  fin pour
13 :   $C_{new} \leftarrow \emptyset$ 
14 :  supprimer de  $F_m$  les items  $f$  tel que  $\exists f' \in F_m, f \subseteq f'$ 
15 :  supprimer de  $C$  les groupes  $c$  tel que  $\exists f' \in F_m$  avec  $h(c) \cup t(c) \subseteq f'$ 
16 : fin tant que
17 : Retourner  $F_m$ 

```

Algorithme 16 Algorithme Gen-Initial-Groups

Entrée : Contexte \mathcal{B} ; Ensemble C de groupe candidat; Seuil minimal support $minsupp$.**Sortie :** Ensemble C initialisé; 1-itemset possédant le plus grand support.

```

1 :  $F_{m1} \leftarrow \{1\text{-itemsets fréquents dans } \mathcal{B}\}$ ;
2 : ordonner les itemsets contenus dans  $F_{m1}$  par supports croissants;
3 : pour tout item  $i \in F_{m1}$  autre que le plus grand item dans l'ordre faire
4 :   créer un groupe candidat  $c$ 
5 :    $h(c) \leftarrow \{i\}$ 
6 :    $t(c) \leftarrow \{i' \in \mathcal{I} | i < i' \text{ dans l'ordre sur les items}\}$ 
7 :    $C \leftarrow C \cup \{c\}$ 
8 : fin pour
9 : Retourner l'itemset  $f \in F_{m1}$  contenant le plus grand item dans l'ordre

```

Algorithme 17 Algorithme Gen-Sub-Nodes

Entrée : Groupe candidat c ; Ensemble C de groupe candidat; Seuil minimal support $minsupp$.

Sortie : Ensemble C mis à jour; itemset fréquent $h(c) \cup \{i\}$ possédant le plus grand support pour $i \in t(c)$ ou $h(c)$

```

1 : pour tout item  $i \in t(c)$  faire
2 :   si  $h(c) \cup \{i\}$  est fréquent alors
3 :      $t(c) \leftarrow t(c) \cup \{i\}$ 
4 :   fin si
5 : fin pour
6 : ordonner les itemsets  $i$  de  $t(c)$  contenus par supports de  $h(c) \cup \{i\}$  croissants;
7 : pour tout item  $i \in t(c)$  autre que le plus grand itemset dans  $t(c)$  faire
8 :   créer un groupe candidat  $c'$ 
9 :    $h(c') \leftarrow h(c) \cup \{i\}$ 
10 :   $t(c') \leftarrow \{i' \in t(c') \mid i' < i \text{ dans } t(c)\}$ 
11 :   $C \leftarrow C \cup \{c'\}$ 
12 : fin pour
13 : si  $t(c) = \emptyset$  alors retourner  $h(c)$ 
14 : sinon retourner  $h(c) \cup \{i\}$  avec  $i$  plus grand item dans  $t(c)$ 

```

Algorithme 18 Algorithme CLOSE

Entrée : Contexte \mathcal{B} ; Seuil minimal de support $minsupp$.

Sortie : F : Ensemble des motifs fermés fréquents

```

1 :  $FC_1 \leftarrow \{1 - motifs\}$ 
2 : pour tout ( $k = 1; FC_k.générateurs \neq \emptyset; k++$ ) faire
3 :    $FC_k.fermés \leftarrow \emptyset$ 
4 :    $FC_k.supports \leftarrow 0$ 
5 :    $FC_k \leftarrow Gen - Closure(F_k)$ 
6 :   pour tout  $c \in F_k$  faire
7 :     si  $c.support \geq minsupp$  alors
8 :        $F_k \leftarrow F_k \cup \{c\}$ 
9 :     fin si
10 :  fin pour
11 :   $FC_{k-1} \leftarrow Gen - Generator(F_k)$ 
12 : fin pour
13 : Retourner  $F_k \cup_k F_k$ 

```

Algorithme 19 Algorithme Gen-Closure

Entrée : FC_k ; Contexte \mathcal{B} **Sortie :** FC_k .fermés; FC_k .supports

```

1 : pour tout  $e \in E$  faire
2 :    $G_e \leftarrow$  sous-ensemble( $FC_k$ .générateurs,  $f(e)$ )
3 :   pour tout p.générateur  $\in G_e$  faire
4 :     si p.fermé =  $\emptyset$  alors
5 :       p.fermé  $\leftarrow$   $f(e)$ 
6 :     sinon
7 :       p.fermé  $\leftarrow$  p.fermé  $\cup$   $f(\{e\})$ 
8 :     fin si
9 :     p.support  $++$ 
10 :   fin pour
11 : fin pour
12 : Retourner  $\cup\{p \in FC_k \mid p.fermé \neq \emptyset\}$ 

```

Algorithme 20 Algorithme MaxCLQ(G, C, LB), un algorithme de séparation et évaluation de base pour Maxclique

Entrée : Un graphe $G = (V, E)$, une clique C , et une borne inférieure LB représentant le cardinalité de la plus grande clique trouvé jusqu'à présent**Sortie :** $C \cup C'$, où C' est une clique maximum de G , si $|C \cup C'| > LB$, \emptyset sinon

```

1 : begin
2 : si  $|V| = 0$  alors retourner  $C$ ;
3 :  $UB \leftarrow$  surestimation( $G$ ) +  $|C|$ ;
4 : si  $LB \geq UB$  alors retourner  $\emptyset$ ;
5 : choisir un sommet  $v$  de degré minimum de  $G$ 
6 :  $C_1 \leftarrow$  MaxCLQ( $G_v, C \cup \{v\}, LB$ );
7 : si  $|C_1| > LB$  alors  $LB \leftarrow |C_1|$ ;
8 :  $C_2 \leftarrow$  MaxCLQ( $G \setminus v, C, LB$ );
9 : si  $|C_1| \geq |C_2|$  alors retourner  $C_1$ ; sinon
10 : Retourner  $C_2$ ;
11 : end

```

Algorithme 21 Algorithme surestimation(G), une surestimation de la cardinalité d'une clique maximum de G

Entrée : Un graphe $G=(V, E)$

Sortie : Borne supérieure pour une clique maximum de G

```
1 : begin
2 :  $P \leftarrow \emptyset$ ;
3 :  $G' \leftarrow G$ ;
4 : tant que  $G'$  est non vide faire
5 :    $v \leftarrow$  le sommet de degré maximum de  $G'$ 
6 :   si il existe un ensemble indépendant  $S$  dans  $P$  où  $v$  est non-connecté à tous
       les sommets alors
7 :     insérer  $v$  dans  $S$ ;
8 :   sinon
9 :     créer un nouveau ensemble indépendant  $S$ ;
10 :    insérer  $v$  dans  $S$ ;
11 :     $P \leftarrow P \cup \{S\}$ ;
12 :     $s \leftarrow 0$ ;
13 :   fin si
14 : fin tant que
15 : Coder  $G$  en une instance MaxSAT  $\phi$  en basant sur  $P$ ;
16 : tant que  $\phi$  contient une clause souple qui n'est pas testée faire
17 :    $c \leftarrow$  la clause souple de  $\phi$  de taille minimum qui n'est pas testée;
18 :   marquer  $c$  comme "testée";
19 :   si chaque littéral dans  $c$  est d'échec alors
20 :     retirer  $c$  et toutes les clauses souples faisant les littéraux de  $c$  d'échec de  $\phi$ ;
21 :    $s \leftarrow s + 1$ ;
22 : fin si
23 : fin tant que
24 : Retourner  $|P| - s$ ;
```

Annexe B

Listing des codes sources du
CHIC- M_{GK} et affichage des
résultats d'expérience

B.1 Programme CHIC- M_{GK}

Pour l'implémentation de notre logiciel, nous avons comme interface principale (Figure B.1), composée de la fenêtre principale, 7 boutons et une fenêtre d'affichage de résultats à droite :

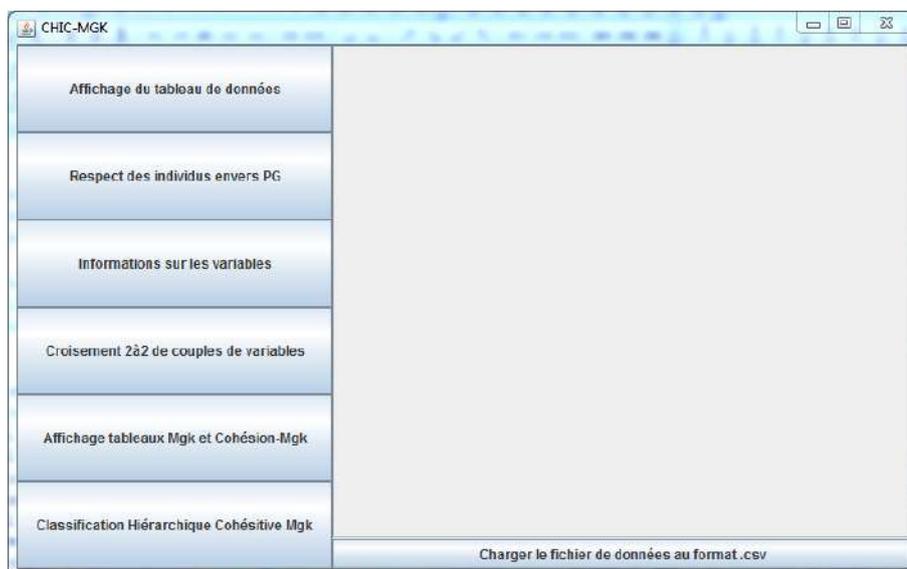


Figure B.1 – Fenêtre d'interface principale

Le listing B.1 montre les codes sources de création de ces objets sus-mentionnés : B.1-lignes[109-158]. Et tous ces objets font partie de la classe principale du programme nommée : Fenetre_Principal B.1-ligne[36]

Pour les environnements API, tels que les classes du toolkit AWT (Abstract Windows Toolkit) *java.awt* et SWT (Standard Widget Toolkit) *java.swt* qui permettent d'écrire des interfaces graphiques indépendantes du système d'exploitation sur lesquels elles vont fonctionner. Ces bibliothèques utilisent le système graphique de la plateforme d'exécution (Windows, MacOS, X-Window) pour afficher les objets graphiques. Les toolkits contiennent des classes décrivant les composants graphiques, les polices, les couleurs et les images.

La classe *java.io* entrée-sortie qui permet de lister les fichiers d'un répertoire, de savoir si un fichier existe, de renommer un fichier, de supprimer un fichier..., de faire des échanges de données entre le programme et une autre source. Et aussi, la classe des collections *java.util* B.1-lignes[1-35].

Maintenant nous allons voir les étapes de traitements du programme.

B.1.1 Chargement de données

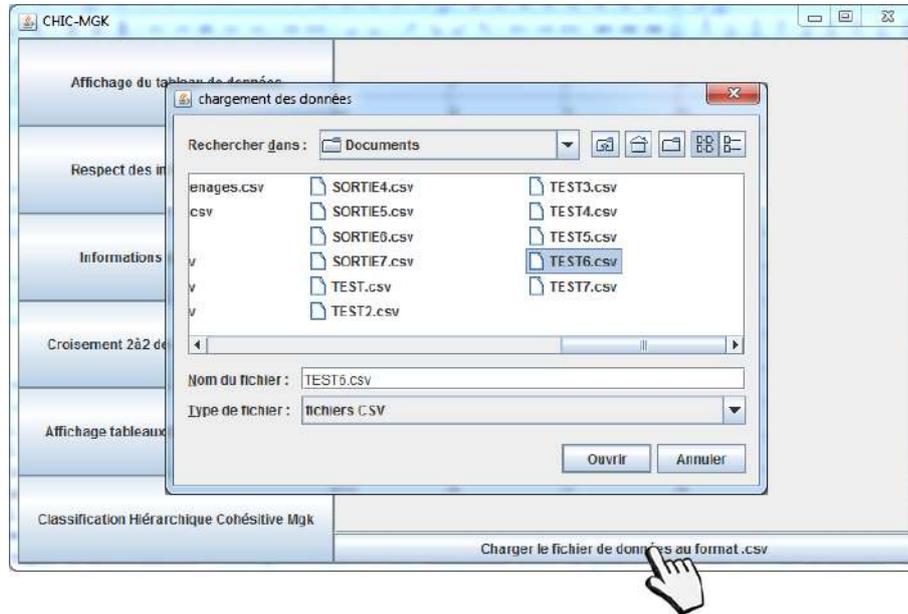


Figure B.2 – Chargement des données externes au format .csv

Affichage des données récupérées :

	V1	V2	V3	V4
i1	0	1	0	1
i2	0	1	0	1
i3	0	1	0	1
i4	0	1	0	1
i5	0	1	1	1
i6	0	1	1	0
i7	1	1	1	0
i8	1	1	1	0
i9	1	1	1	0
i10	1	1	1	0
i11	0	1	0	0
i12	0	1	0	1
i13	0	1	0	1

Figure B.3 – Récupération et affichage des données externes

B.1.2 Informations sur les variables

Nous pouvons aussi avoir les descriptions de chaque variable formant le contexte en cliquant sur le bouton Information sur les variables. Cela nous montre le nombre des individus i comportant la variable $v(i)$: Occurrence, sa moyenne et son écart-type.

		occurrence	moyenne	ecart-type
V1	17	0.566666666666667	0.5040069329937309	
V2	19	0.633333333333333	0.490132517853561	
V3	10	0.333333333333333	0.4794633014853842	
V4	20	0.666666666666666	0.4794633014853842	
V5	25	0.833333333333334	0.3790490217894517	
V6	14	0.466666666666667	0.6074162634049248	
V7	25	0.833333333333334	0.3790490217894517	
V8	18	0.6	0.49827287912243984	
V9	19	0.633333333333333	0.490132517853561	
V10	20	0.666666666666666	0.4794633014853842	

Figure B.4 – Descriptions des variables

B.1.3 Croisement 2 à 2 de couples de variables

Avec ce menu que nous pouvons calculer les valeurs de tableau de contingence de couples de variables croisés deux à deux, les valeurs de leurs supports, confiances, corrélations, similarités, χ^2 et M_{GK} respectifs.

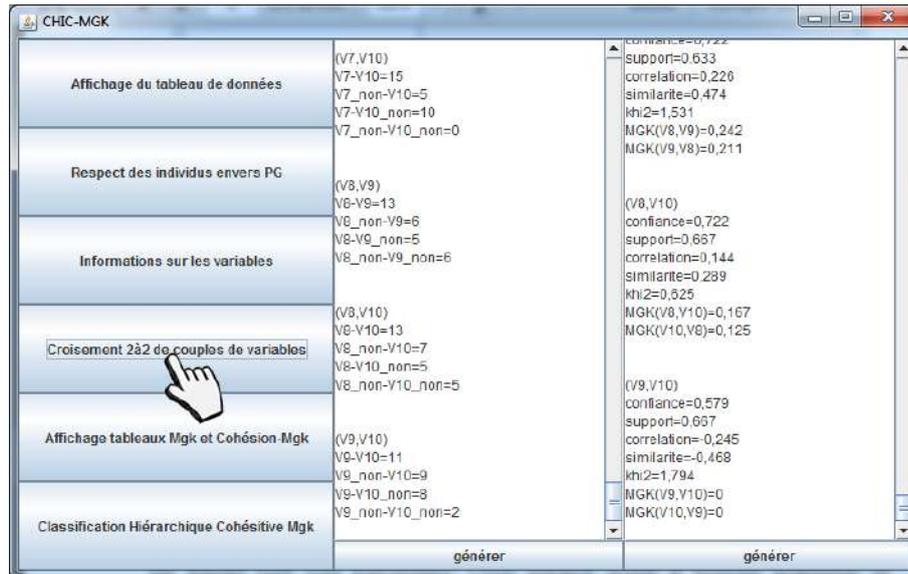


Figure B.5 – Croisement 2 à 2 de couples de variables

B.1.4 Affichage des tableaux M_{GK} et cohésion à base de M_{GK}

Ce menu nous permet d'afficher les résultats des valeurs de M_{GK} -valides ainsi que leurs cohésions respectives sous forme de tableau.

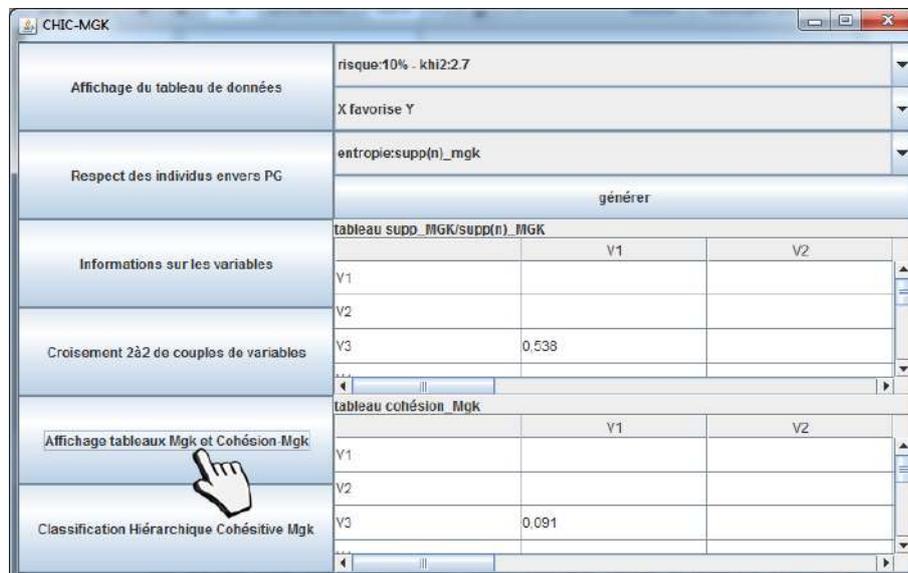


Figure B.6 – Affichage des résultats de M_{GK} et cohésion- M_{GK}

B.1.5 Classification Hiérarchique et Cohésitive selon M_{GK}

Ce menu affiche les 4 boutons, tels que :

Extraction des règles permet d'afficher

- les valeurs des cohésions des couples de variables formant les règles valides triées par ordre décroissante,
- les méta-règles obtenues ainsi que les cohésions inter-classes de chaque méta-règle,
- les valeurs des niveaux et nœuds significatifs.

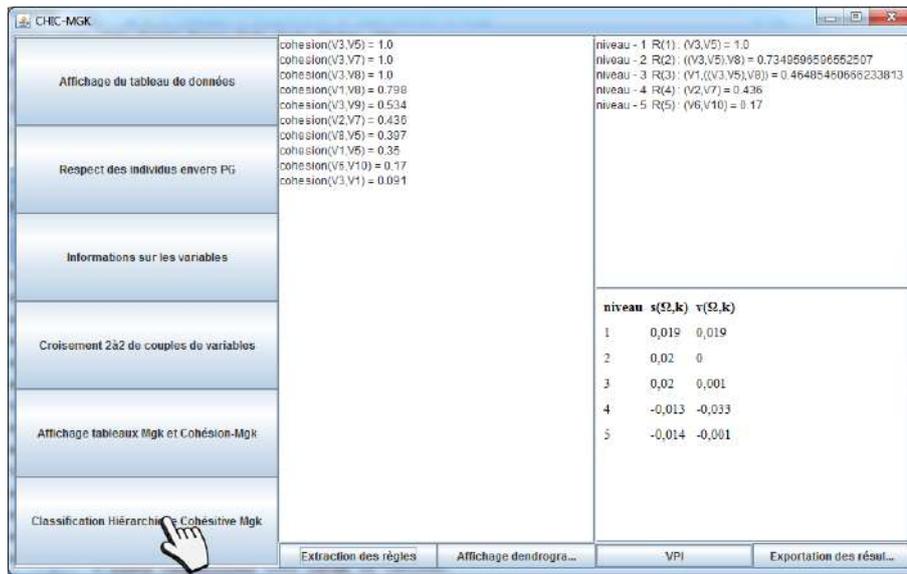


Figure B.7 – CHIC- M_{GK}

Affichage dendrogramme: permet d’afficher le dendrogramme (arbre de la classification)

VPI: permet d’afficher le tableau contenant les valeurs de respect Φ , de distance empirique, de distance idéale théorique et de contribution de chaque individu qui participe à la formation d’une classe à un niveau donné.

Exportation de résultats: permet d’exporter tous les résultats dans un fichier au format .csv pour faciliter la récupération et la présentation de ces résultats généralement sous forme de tableaux.

B.2 Classe principale

```
1 import java.awt.BorderLayout;
2 import java.awt.CardLayout;
3 import java.awt.Dimension;
4 import java.awt.GridLayout;
5 import java.awt.Image;
6 import java.awt.event.ActionEvent;
7 import java.awt.event.ActionListener;
8 import java.awt.event.WindowAdapter;
9 import java.awt.event.WindowEvent;
10 import java.awt.image.RenderedImage;
11 import java.io.BufferedReader;
12 import java.io.File;
13 import java.io.FileNotFoundException;
14 import java.io.FileReader;
15 import java.io.FileWriter;
16 import java.io.IOException;
17 import java.io.PrintWriter;
18 import java.text.DecimalFormat;
19 import java.util.ArrayList;
20 import java.util.Iterator;
21 import java.util.Vector;
22 import javax.imageio.ImageIO;
23 import javax.swing.JButton;
24 import javax.swing.JComboBox;
25 import javax.swing.JEditorPane;
26 import javax.swing.JFileChooser;
27 import javax.swing.JFrame;
28 import javax.swing.JLabel;
29 import javax.swing.JOptionPane;
30 import javax.swing.JPanel;
31 import javax.swing.JScrollPane;
32 import javax.swing.JTable;
33 import javax.swing.JTextArea;
34 import javax.swing.filechooser.FileNameExtensionFilter;
35 import javax.swing.table.DefaultTableModel;
36 public class Fenetre_Principale extends JFrame implements ActionListener{
37 private static final long serialVersionUID = 1L;
38 //affichage
```

```
39 private CardLayout card=new CardLayout();
40 private JPanel card_pan;
41 //affichage des valeurs
42 private JPanel pan_valeur;
43 private JButton bouton_valeur;
44 private JButton bouton_charger;
45 private JScrollPane scroll_valeur;
46 private JTable tab_valeur;
47 //affichage des variables
48 private JPanel pan_variables;
49 private JButton bouton_variables;
50 private JScrollPane scroll_variables;
51 private JTable tab_variables;
52 //affichage individus
53 private JPanel pan_individus;
54 private JButton bouton_individus;
55 private JTable tab_individus;
56 private JScrollPane scroll_individus;
57 //private JTable tab_individus;
58 //affichage couples
59 private JPanel pan_couples;
60 private JPanel pan_txt_couple;
61 private JPanel pan_txt_calcul;
62 private JScrollPane scroll_couple;
63 private JScrollPane scroll_calcul;
64 private JButton bouton_couples;
65 private JTextArea text_couple;
66 private JTextArea text_calcul;
67 private JButton bouton_generer_couple;
68 private JButton bouton_generer_calcul;
69 //affichage tableau
70 private JPanel pan_tableau;
71 private JPanel pan_MGK;
72 private JPanel pan_cohesion;
73 private JScrollPane scroll_MGK;
74 private JScrollPane scroll_cohesion;
75 private JPanel pan_head_tableau;
76 private JTable tab_MGK;
77 private JTable tab_cohesion;
78 private JButton bouton_generer_tableau;
```

```

79 private JButton bouton_tableau;
80 String[] risk={"risque:aucun","risque:0.1% - khi2:10.83","risque:1.0% - khi2:6.63
    ", "risque:5.0% - khi2:3.84","risque:10% - khi2:2.7"};
81 String[] favorise={"aucun","X favorise Y","X defavorise Y"};
82 String[] entropy={"entropie:supp(n)_mgk","entropie:supp_mgk"};
83 private JComboBox combo_risk;
84 private JComboBox combo_favorise;
85 private JComboBox combo_entropy;
86 //affichage classification
87 private JPanel pan_classification;
88 private JPanel pan_centre_classification;
89 private JPanel pan_classement_classification;
90 private JPanel pan_bouton_classification;
91 private JScrollPane scroll_classification;
92 private JScrollPane scroll_couple_classification;
93 private JScrollPane scroll_significatif_classification;
94 //private JScrollPane scroll_vpi_classification;
95 private JButton bouton_classification;
96 private JTextArea text_classification;
97 private JTextArea text_couple_classification;
98 //private JTextEditorPane text_vpi_classification;
99 private JTextEditorPane text_significatif_classification;
100 private JButton bouton_classifieur;
101 private JButton bouton_dendogramme;
102 private JButton bouton_exporter;
103 private JButton bouton_vpi;
104 String path="";
105 //menu
106 private JPanel pan_menu;
107 //fichier sortie excel
108 PrintWriter sortie=null;
109 //construction de la fenetre principale
110 public Fenetre_Principal() {
111 this.setTitle("CHIC-MGK");
112 this.setSize(800,500);
113 this.getContentPane().setLayout(new BorderLayout());
114 //les menus
115 pan_menu=new JPanel();
116 this.getContentPane().add(pan_menu, BorderLayout.WEST);
117 pan_menu.setLayout(new GridLayout(6,1));

```

```
118 bouton_valeur=new JButton("Affichage du tableau de donnees");
119 bouton_valeur.addActionListener(this);
120 pan_menu.add(bouton_valeur);
121 bouton_individus=new JButton("Respect des individus envers PG");
122 bouton_individus.addActionListener(this);
123 pan_menu.add(bouton_individus);
124 bouton_variables=new JButton("Informations sur les variables");
125 bouton_variables.addActionListener(this);
126 pan_menu.add(bouton_variables);
127 bouton_couples=new JButton("Croisement 2a2 de couples de variables");
128 bouton_couples.addActionListener(this);
129 pan_menu.add(bouton_couples);
130 bouton_tableau=new JButton("Affichage tableaux Mgk et Cohesion-Mgk");
131 bouton_tableau.addActionListener(this);
132 pan_menu.add(bouton_tableau);
133 bouton_classification=new JButton("Classification Hierarchique Cohesitive Mgk");
134 bouton_classification.addActionListener(this);
135 pan_menu.add(bouton_classification);
136 //affichage
137 card_pan=new JPanel();
138 this.getContentPane().add(card_pan, BorderLayout.CENTER);
139 card_pan.setLayout(card);
140 pan_valeur=new JPanel();
141 card_pan.add(pan_valeur, "valeur");
142 pan_individus=new JPanel();
143 card_pan.add(pan_individus, "individus");
144 pan_variables=new JPanel();
145 card_pan.add(pan_variables, "variables");
146 pan_couples=new JPanel();
147 card_pan.add(pan_couples, "couples");
148 pan_tableau=new JPanel();
149 card_pan.add(pan_tableau, "tableau");
150 pan_classification=new JPanel();
151 card_pan.add(pan_classification, "classification");
152 //affichage des valeurs
153 pan_valeur.setLayout(new BorderLayout());
154 scroll_valeur=new JScrollPane();
155 pan_valeur.add(scroll_valeur, BorderLayout.CENTER);
156 bouton_charger=new JButton("Charger le fichier de donnees au format .csv");
157 bouton_charger.addActionListener(this);
```

```
158 pan_valeur.add(bouton_charger, BorderLayout.SOUTH);
159 //affichage des individus
160 scroll_individus=new JScrollPane();
161 pan_individus.setLayout(new BorderLayout());
162 pan_individus.add(scroll_individus, BorderLayout.CENTER);
163 //affichage des variables
164 scroll_variables=new JScrollPane();
165 pan_variables.setLayout(new BorderLayout());
166 pan_variables.add(scroll_variables, BorderLayout.CENTER);
167 pan_couples.setLayout(new GridLayout(1,2));
168 pan_txt_couple=new JPanel();
169 text_couple=new JTextArea();
170 bouton_generer_couple=new JButton("generer");
171 bouton_generer_couple.addActionListener(this);
172 pan_txt_couple.setLayout(new BorderLayout());
173 scroll_couple=new JScrollPane();
174 scroll_couple.setViewportView(text_couple);
175 pan_txt_couple.add(scroll_couple, BorderLayout.CENTER);
176 pan_txt_couple.add(bouton_generer_couple, BorderLayout.SOUTH);
177 pan_couples.add(pan_txt_couple);
178 pan_txt_calcul=new JPanel();
179 text_calcul=new JTextArea();
180 bouton_generer_calcul=new JButton("generer");
181 bouton_generer_calcul.addActionListener(this);
182 pan_txt_calcul.setLayout(new BorderLayout());
183 scroll_calcul=new JScrollPane();
184 scroll_calcul.setViewportView(text_calcul);
185 pan_txt_calcul.add(scroll_calcul, BorderLayout.CENTER);
186 pan_txt_calcul.add(bouton_generer_calcul, BorderLayout.SOUTH);
187 pan_couples.add(pan_txt_calcul);
188 pan_tableau.setLayout(new GridLayout(3,1));
189 pan_head_tableau=new JPanel();
190 pan_head_tableau.setLayout(new GridLayout(4,1));
191 combo_risk=new JComboBox(risk);
192 pan_head_tableau.add(combo_risk);
193 combo_favorise=new JComboBox(favorise);
194 combo_favorise.setSelectedIndex(1);
195 pan_head_tableau.add(combo_favorise);
196 combo_entropy=new JComboBox(entropy);
197 pan_head_tableau.add(combo_entropy);
```

```
198 bouton_generer_tableau=new JButton("generer");
199 bouton_generer_tableau.addActionListener(this);
200 pan_head_tableau.add(bouton_generer_tableau);
201 pan_tableau.add(pan_head_tableau);
202 pan_MGK=new JPanel();
203 JLabel lab_mgk=new JLabel("tableau supp_MGK/supp(n)_MGK");
204 pan_MGK.setLayout(new BorderLayout());
205 pan_MGK.add(lab_mgk, BorderLayout.NORTH);
206 scroll_MGK=new JScrollPane();
207 pan_MGK.add(scroll_MGK, BorderLayout.CENTER);
208 pan_tableau.add(pan_MGK);
209 pan_cohesion=new JPanel();
210 pan_cohesion.setLayout(new BorderLayout());
211 JLabel lab_cohesion=new JLabel("tableau cohesion_Mgk");
212 pan_cohesion.add(lab_cohesion, BorderLayout.NORTH);
213 scroll_cohesion=new JScrollPane();
214 pan_cohesion.add(scroll_cohesion, BorderLayout.CENTER);
215 pan_tableau.add(pan_cohesion);
216 //classification
217 pan_classification.setLayout(new BorderLayout());
218 pan_centre_classification=new JPanel();
219 pan_classement_classification=new JPanel();
220 scroll_couple_classification=new JScrollPane();
221 text_classification=new JTextArea("");
222 text_couple_classification=new JTextArea("");
223 text_significatif_classification=new JEditorPane("text/html", "");
224 //text_vpi_classification=new JEditorPane("text/html", "");
225 pan_centre_classification.setLayout(new GridLayout(1,2));
226 scroll_couple_classification.setViewportViewView(text_couple_classification);
227 pan_centre_classification.add(scroll_couple_classification);
228 pan_classement_classification.setLayout(new GridLayout(2,1));
229 scroll_classification=new JScrollPane();
230 scroll_classification.setViewportViewView(text_classification);
231 pan_classement_classification.add(scroll_classification);
232 scroll_significatif_classification=new JScrollPane();
233 scroll_significatif_classification.setViewportViewView(
    text_significatif_classification);
234 pan_classement_classification.add(scroll_significatif_classification);
235 pan_centre_classification.add(pan_classement_classification);
236 //scroll_vpi_classification=new JScrollPane();
```

```

237 //scroll_vpi_classification.setViewportView(text_vpi_classification);
238 //pan_classement_classification.add(scroll_vpi_classification);
239 pan_classification.add(pan_centre_classification, BorderLayout.CENTER);
240 pan_bouton_classification=new JPanel();
241 bouton_classifier=new JButton("Extraction des regles");
242 bouton_classifier.addActionListener(this);
243 bouton_dendogramme=new JButton("Affichage dendrogramme");
244 bouton_dendogramme.addActionListener(this);
245 bouton_exporter=new JButton("Exportation des resultats dans Excel");
246 bouton_exporter.addActionListener(this);
247 bouton_vpi=new JButton("VPI");
248 bouton_vpi.addActionListener(this);
249 pan_bouton_classification.setLayout(new GridLayout(1,4));
250 pan_bouton_classification.add(bouton_classifier);
251 pan_bouton_classification.add(bouton_dendogramme);
252 pan_bouton_classification.add(bouton_vpi);
253 pan_bouton_classification.add(bouton_exporter);
254 pan_classification.add(pan_bouton_classification, BorderLayout.SOUTH);
255 //gestion fermeture de la fenetre
256 this.addWindowListener(new WindowAdapter() {
257     public void windowClosing(WindowEvent e){
258         int reponse=JOptionPane.showConfirmDialog(null, "Voulez-vous quitter l'
                application?","fermeture de l'application", JOptionPane.YES_OPTION);
259         if(reponse==JOptionPane.YES_OPTION){
260             if(!(sortie==null)){
261                 sortie.close();
262             }
263             System.exit(0);
264         }
265     }
266 });
267 this.setLocationRelativeTo(null);
268 this.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
269 this.setVisible(true);
270 }
271 //main
272 public static void main(String[] args) {
273     new Fenetre_Principal();
274 }
275 //les couples formes a partir des variables retires dans le fichier csv

```

```
276 private ArrayList<Couple> couple_organise=new ArrayList<Couple>();
277 //recuperation des donnees du fichier excel au format .csv
278 //organisation des couples de variables
279 public ArrayList<Couple> organiser_couple(ArrayList<Variable> liste_variables){
280 //sortie couples
281     ArrayList<Couple> val=new ArrayList<Couple>();
282 //les variables qui vont former les couples
283     ArrayList<Variable> temp=new ArrayList<Variable>();
284 //initialiser avec la liste de variables recuperes du fichier excel
285     temp.addAll(liste_variables);
286 int i=0;
287 int j=0;
288 while(i<liste_variables.size()){
289 j=i+1;
290 while(j<liste_variables.size()){
291 //formation des couples
292 val.add(new Couple(temp.get(i), temp.get(j)));
293 j++;
294 }
295 i++;
296 }
297 return val;
298 }
299 //affichage des couples dans le JTextarea
300 public void afficher_couple(ArrayList<Couple> liste_couples,JTextArea affichage){
301 affichage.setText("");
302 Iterator<Couple> it=liste_couples.iterator();
303 sortie.println("* * * * *");
304 sortie.println("          VALEURS DES COUPLES          ");
305 sortie.println("* * * * *");
306 while(it.hasNext()){
307 Couple temp=it.next();
308 String str_temp="\n \n";
309 String V1=temp.variable1.nom;
310 String V2=temp.variable2.nom;
311 str_temp=str_temp.concat("("+V1+", "+V2+")\n");
312 str_temp=str_temp.concat(V1+"-"+V2+"="+temp.couple("11")+"\n");
313 str_temp=str_temp.concat(V1+"_non-"+V2+"="+temp.couple("01")+"\n");
314 str_temp=str_temp.concat(V1+"-"+V2+"_non="+temp.couple("10")+"\n");
315 str_temp=str_temp.concat(V1+"_non-"+V2+"_non="+temp.couple("00")+"\n");
```

```

316 sortie.println(";"+V2+";"+"non_"+V2+";"+";"+"khi2;" +temp.khi2()+";
317 "+"support("+V1+");"+temp.supp_x());
318 sortie.println(V1+";"+temp.couple("11")+";"+temp.couple("10")+";"+";"+
319 +"support("+V2+");"+temp.supp_y()+";similarite;" +temp.similarite());
320 sortie.println("non_"+V1+";"+temp.couple("01")+";"+temp.couple("00")+";
321 "+";"+"confiance("+V1+", "+V2+");"+ temp.confiance()+";correlation;" +temp.
    correlation());
322 sortie.println("*****");
323 affichage.append(str_temp);
324     }
325 }
326 ArrayList<Double> liste_suppx;
327 ArrayList< Double> liste_suppy;
328 ArrayList<ArrayList<String>> liste_nom_mgk;
329 //calcul MGK
330 public ArrayList<Double> calcul_MGK(ArrayList<Couple> liste_couples,double khi,
    boolean favorise){
331 /*
332 * liste_couples: liste des couples formes a partir des variables dans le fichier
    excel
333 * khi: khi theorique
334 * favorise: prend la valeur "vrai" si le calcul du mgk depend de confiance>support
    de y
335 */
336 liste_nom_mgk=new ArrayList<>();
337 ArrayList<Double> liste=new ArrayList<Double>();
338 liste_suppx=new ArrayList<Double>();
339 liste_suppy=new ArrayList<Double>();
340 Iterator<Couple> it=liste_couples.iterator();
341 sortie.println("*****");
342 sortie.println("VALEURS DE MGK ET MGK CRITIQUE "+"Khi2="+khi);
343 sortie.println("*****");
344 int d=1;
345 //System.out.println("N°"+" ;"+"MGK(X-->Y)"+" ;"+"MGK-critique"+" ;"
346 +"Khi2-Theorique"+" ;"+"conf(X-->Y)"+" ;"+"correlation"+" ;"+"supp(X-->Y)"+" ;"
347 +"supp(X)"+" ;"+"supp(Y)"+" ;"+"X favorise Y"+" ;"+"MGK(X-->non_Y)");
348 sortie.println("N°"+" ;"+"MGK(X-->Y)"+" ;"+"MGK-critique"+" ;"
349 +"Khi2-Theorique"+" ;"+"conf(X-->Y)"+" ;"+"correlation"+" ;"+"supp(X-->Y)"+" ;"
350 +"supp(n) (X-->Y)"+" ;"+"supp(X)"+" ;"+"supp(Y)"+" ;"+"X favorise Y"+" ;"+"MGK(X-->
    non_Y)");

```

```

351 while(it.hasNext()){
352 ArrayList<String> temp_nom_mgk=new ArrayList<String>();
353 Couple temp=it.next();
354 temp_nom_mgk.add(temp.variable1.nom);
355 temp_nom_mgk.add(temp.variable2.nom);
356 liste_nom_mgk.add(temp_nom_mgk);
357 double mgk=temp.mgk();
358 double supp_x=temp.supp_x();
359 double supp_y=temp.supp_y();
360 //si confiance > support de y
361 if(favorise){
362 if(!temp.favorise()){
363 mgk=0.0 ;
364 }
365 }
366 //calcul de mgk critique
367 double critique=temp.mgk_critique(khi);
368 //*****
369 //critique pour les regles negatives
370 double t;
371 // Impression Mgk Critique
372 if (critique != 0){
373 if (0<=temp.mgk_brn()){
374 t=temp.mgk_brn();
375 //System.out.println("Valeur_mgk+" ;+"Mgk_critique");
376 //System.out.println(mgk+" ;"+critique+" ;"+temp.khi2()+" ;"+temp.confiance()+" ;"
377 +temp.favorise()+" ;"+temp.mgk_brn());
378         }else
379         t=0 ;
380 //System.out.println(d+" ;"+mgk+" ;"+critique+" ;"+temp.khi2()+" ;"
381 +temp.confiance()+" ;"+temp.correlation()+" ;"
382 +temp.supp_x()*((1-temp.supp_y())*mgk+temp.supp_y())+" ;"
383 +(temp.supp_x()*((1-temp.supp_y())*mgk+temp.supp_y())
384 -(temp.supp_x()*temp.supp_y()))/(temp.supp_x()*(1-temp.supp_y()))+" ;"
385 +temp.supp_x()+" ;"+temp.supp_y()+" ;"+temp.favorise()+" ;"+t);
386 sortie.println(d+" ;"+mgk+" ;"+critique+" ;"+temp.khi2()+" ;"+temp.confiance()+" ;"
387 +temp.correlation()+" ;"+temp.supp_x()*((1-temp.supp_y())*mgk+temp.supp_y())+" ;"
388 +(temp.supp_x()*((1-temp.supp_y())*mgk+temp.supp_y())
389 -(temp.supp_x()*temp.supp_y()))/(temp.supp_x()*(1-temp.supp_y()))+" ;"
390 +temp.supp_x()+" ;"+temp.supp_y()+" ;"+temp.favorise()+" ;"+t);

```

```
391             d++;
392 // sortie.println();
393 // sortie.println();
394 }
395 //si on choisit aucun, la valeur de "critique" (mgk critique) est a 0
396 if(mgk>critique){
397 liste.add(mgk);
398 liste_suppx.add(supp_x);
399 liste_suppy.add(supp_y);
400 }
401 else{
402 liste.add(0.0);
403 liste_suppx.add(0.0);
404 liste_suppy.add(0.0);
405 }
406 //couple inverse (exemple si temp=(V1,V2) alors temp_2=(V2,V1))
407 Couple temp_2=new Couple(temp.variable2,temp.variable1);
408 temp_nom_mgk=new ArrayList<>();
409 temp_nom_mgk.add(temp_2.variable1.nom);
410 temp_nom_mgk.add(temp_2.variable2.nom);
411 liste_nom_mgk.add(temp_nom_mgk);
412 mgk=temp_2.mgk();
413 supp_x=temp_2.supp_x();
414 supp_y=temp_2.supp_y();
415 if(favorise){
416 if(!temp_2.favorise()){
417 mgk=0.0 ;
418 }
419 }
420 critique=temp_2.mgk_critique(khi);
421 if(mgk>critique){
422 liste.add(mgk);
423 liste_suppx.add(supp_x);
424 liste_suppy.add(supp_y);
425 }
426 else{
427 liste.add(0.0);
428 liste_suppx.add(0.0);
429 liste_suppy.add(0.0);
430 }
```

```
431 }
432 System.out.println("liste="+liste.size());
433 return liste;
434 }
435 //calcul MGK
436 public ArrayList<Double> calcul_MGK_BRN(ArrayList<Couple> liste_couples,double
    khi){
437 /*
438 * voir les commentaires calcul_MGK
439 */
440 ArrayList<Double> liste=new ArrayList<Double>();
441 Iterator<Couple> it=liste_couples.iterator();
442 while(it.hasNext()){
443 Couple temp=it.next();
444 double mgk=0.0;
445 //double old=0.0;
446 if(!temp.favorise()){
447 mgk=temp.mgk_brn();
448 }
449 double critique=temp.mgk_critique(khi);
450 if(mgk>critique){
451 liste.add(mgk);
452 }
453 else{
454 liste.add(0.0);
455 }
456 Couple temp_2=new Couple(temp.variable2,temp.variable1);
457 mgk=0.0;
458 if(!temp_2.favorise()){
459 mgk=temp_2.mgk();
460 }
461 critique=temp_2.mgk_critique(khi);
462 if(mgk>critique){
463 liste.add(mgk);
464 }
465 else{
466 liste.add(0.0);
467 }
468 }
469 return liste;
```

```
470 }
471 //calcul entropie
472 public double calcul_entropy(double p){
473 /*
474 * p=mgk ou p=supp_mgk
475 */
476 double entr=0.0;
477 //*****
478 /*
479     if(p!=1.0 && p>0.5){
480         // if(p>=0.5){
481 entr=(-p*Math.log(p)-(1-p)*Math.log(1-p))/Math.log(2);
482         }
483 //else if(p=0.5){
484 else entr=1;
485     //}
486 return entr;
487 */
488 //*****
489     if(p==1)
490     {
491         return entr =0;
492     }
493     else if(p>0.5)
494     {
495         return entr = (-p*Math.log(p)-(1-p)*Math.log(1-p))/Math.log(2);
496     }
497     else return entr = 1;
498 }
499 //calcul cohesion
500 public double calcul_cohesion(double entropie){
501 double cohesion=0.0;
502 cohesion=Math.sqrt(1-Math.pow(entropie, 2));
503 return cohesion;
504 }
505 /*
506 * nom des variables pour l'affichage dans le tableau ou dans la fenetre du
507     dendrogramme
508 * on ajoute "_non" apres le nom de variable pour les cas de mgk brn
509 */
```

```
509 private String[] nom_variables(ArrayList<Variable> liste_variables, Boolean non){
510 String[] str=new String[liste_variables.size()+1];
511 str[0]="";
512 int i=1;
513 Iterator<Variable> it=liste_variables.iterator();
514 while (it.hasNext()){
515 Variable temp=it.next();
516 if(non){
517 str[i]=temp.nom+"_non";
518 }
519 else{
520 str[i]=temp.nom;
521 }
522 i++;
523 }
524 return str;
525 }
526 /*pour la recherche d'un ArrayList<String> dans un ArrayList<ArrayList<String>>*/
527 //intervertit l'ordre de couple
528 public static void changer(ArrayList<String> couple){
529 String temp="";
530 temp=couple.get(0);
531 couple.remove(0);
532 couple.add(temp);
533 }
534 //retourne l'index de l'element "chercher" dans "liste"
535 public static Integer index(ArrayList<ArrayList<String>> liste,ArrayList<String>
    chercher){
536 return liste.indexOf(chercher);
537 }
538 //retourne le rang de "chercher" dans "liste"
539 public static int rang(ArrayList<ArrayList<String>> liste,ArrayList<String>
    chercher){
540 int i=index(liste,chercher);
541 if(i<0){
542 changer(chercher);
543 i=index(liste,chercher);
544 }
545 return i;
546 }
```

```

547 /*
548 * pour le calcul de l'implication, on recherche la valeur de mgk a partir des
      noms de couple de variables
549 */
550 public double chercher_mgk(ArrayList<ArrayList<String>> temp_nom_mgk,ArrayList<
      Double> temp_valeur_mgk,String gauche,String droite){
551 /*
552 * temp_nom_mgk: contient les noms des couples de variables formes
553 * temp_valeur_mgk: contient les valeurs de mgk de couples dans temp_nom_mgk
554 * (gauche,droite): represente le couple dont on souhaite connaître la valeur de
      mgk
555 */
556         double mgk=0.0 ;
557         ArrayList<String> temp=new ArrayList<>();
558         temp.add(gauche);
559         temp.add(droite);
560         int index=rang(temp_nom_mgk,temp);
561         if(index>-1){
562             mgk=temp_valeur_mgk.get(index);
563         }
564 /*for(int i=0;i<temp_nom_mgk.size();i++){
565 if(temp_nom_mgk.get(i).contains(gauche) && temp_nom_mgk.get(i).contains(droite)){
566 mgk=temp_valeur_mgk.get(i);
567 break;
568 }
569 }*/
570 return mgk;
571 }
572 //recherche la valeur de cohesions a partir des noms de couples de variables
573 public double chercher_cohesion(ArrayList<ArrayList<String>> temp_nom_cohesion,
      ArrayList<Double> temp_valeur_cohesion,String gauche,String droite){
574 /*
575     * voir explications dans cherher_mgk
576     */
577         double retour=0.0 ;
578         ArrayList<String> temp=new ArrayList<>();
579         temp.add(gauche);
580         temp.add(droite);
581         int index=rang(temp_nom_cohesion,temp);
582         if(index>-1){

```

```
583     retour=temp_valeur_cohesion.get(index);
584     }
585     /*for(int i=0;i<temp_nom_cohesion.size();i++){
586     if(temp_nom_cohesion.get(i).contains(gauche) && temp_nom_cohesion.get(i).contains
        (droite)){
587     retour=temp_valeur_cohesion.get(i);
588     break;
589     }
590     }*/
591     return retour;
592 }
593 //maximum des mgk
594 public double max_mgk(ArrayList<ArrayList<String>> temp_nom_mgk,ArrayList<Double>
        temp_valeur_mgk,ArrayList<String> temp_gauche,ArrayList<String> temp_droite)
        {
595 /*
596 * recherche de la valeur maximum de mgk pour un ensemble de variables
597 * temp_gauche:ai avec i={1, .... ,r}
598 * temp_droite:b_nonj avec j={1, ....., s}
599 */
600     double max=0.0;
601     for (int i=0;i<temp_gauche.size();i++){
602     for(int j=0;j<temp_droite.size();j++){
603     double temp=chercher_mgk(temp_nom_mgk, temp_valeur_mgk, temp_gauche.get(i),
        temp_droite.get(j));
604     if(temp>max){
605     max=temp;
606     }
607     }
608     }
609     /*
610 * max=sup[mgk(ai,b_nonj)]
611 */
612     return max;
613     }
614 //ajout produit cohesion
615 public double complement_cohesion(ArrayList<ArrayList<String>> temp_nom_cohesion,
        ArrayList<Double> temp_valeur_cohesion,ArrayList<String> temp_gauche,
        ArrayList<String> temp_droite,String gauche,String droite){
616 /*
```

```
617 * recherche des complements des valeurs de cohesion contenues dans une classe
618 * exemple pour (V1,V2) et (V2,V3) donne (V1,(V2,V3))
619 * cette fonction donnera produit=max_cohesion(V1,V3)*max_cohesion(V2,V3)
620 * max_cohesion(V1,V2) deja trouve dans la fonction generer_classification
621 */
622 double produit=1.0 ;
623 double temp=1 ;
624 if(temp_gauche.size(>0){
625 for(int j=0 ;j<temp_gauche.size();j++){
626 if(temp_droite.size(>0){
627 for(int k=0;k<temp_droite.size();k++){
628 if(!(temp_gauche.get(j).equals(gauche) && temp_droite.get(k).equals(droite))){
629 temp=chercher_cohesion(temp_nom_cohesion, temp_valeur_cohesion, temp_gauche.get(j
        ), temp_droite.get(k));
630 if(temp>0){
631 produit=produit*temp;
632 }
633 else{
634 produit=0.0 ;
635 break;
636 }
637 }
638 }
639 }
640 else{
641 if(!temp_gauche.get(j).equals(gauche)){
642 temp=chercher_cohesion(temp_nom_cohesion, temp_valeur_cohesion, temp_gauche.get(j
        ), droite);
643 if(temp>0){
644 produit=produit*temp;
645 }
646 else{
647 produit=0.0 ;
648 break;
649 }
650 }
651 }
652 }
653 }
654 else{
```

```
655 if(temp_droite.size(>1){
656 for(int i=0;i<temp_droite.size();i++){
657 if(!temp_droite.get(i).equals(droite)){
658 temp=chercher_cohesion(temp_nom_cohesion, temp_valeur_cohesion, gauche,
        temp_droite.get(i));
659 produit=produit*temp;
660 }
661 }
662 }
663 }
664 return produit;
665 }
666 ArrayList<Classification> classifications=null;
667 ArrayList<Point> liste_abscisse;
668 ArrayList<ArrayList<ArrayList<String>>> classification_vpi=new ArrayList<>();
669 //premiere generation de classification
670 private ArrayList<Classification> generer_classification(ArrayList<ArrayList<
        String>> nom_variables,ArrayList<Double> liste_cohesion,ArrayList<ArrayList<
        Integer>> index){
671 /*
672 * cette fonction retourne une liste de classifications sans noeuds c'est a dire
        sans composants graphiques
673 * nom_variables:liste de couples de variables a partir du tableau de cohesion
674 * liste_cohesion: liste des valeurs de cohesion correspondantes de "nom_variables
        "
675 * index:comprendra une liste de couples d'index
676 * exemple: pour une classification (V1,(V2,V3)), le couple d'index (-1,2)
        signifie que:
677 * -1 :pas de classification a renvoyer (correspond a V1)
678 * 2 : correspond a l'index de "classifications" dans lequel se trouve (V2,V3)
679 * cette liste d'index est indispensable dans la fonction "generer_dendogramme"
680 */
681 ArrayList<Classification> classifications=new ArrayList<Classification>();
682 int i=0;
683 int nombre=0;
684 classification_vpi=new ArrayList<>();
685 do{
686 String nom="";
687 String gauche="";
688 String droite="";
```

```

689 ArrayList<String> nom_vars=new ArrayList<String>();
690 ArrayList<String> temp_gauche=new ArrayList<String>();
691 ArrayList<String> temp_droite=new ArrayList<String>();
692 ArrayList<ArrayList<String>> temp_vpi=new ArrayList<>();
693 double res_complement=0.00;
694 ArrayList<Integer> temp_index=new ArrayList<Integer>();
695 double produit_cohesion=0.0;
696 double g=1;
697 double d=1;
698 if(classifications.isEmpty()){
699 gauche=nom_variables.get(i).get(0);
700 droite=nom_variables.get(i).get(1);
701 nom=nom_classification(gauche, droite);
702 nom_vars=nom_variables.get(i);
703 produit_cohesion=liste_cohesion.get(i);
704 temp_index.add(-1);
705 temp_index.add(-1);
706 //temp_vpi.add(nom_variables.get(i));
707 //classification_vpi.add(temp_vpi);
708 //System.out.println("premiere insertion("+nom_variables.get(i).get(0)+","+
       nom_variables.get(i).get(1)+")");
709 }
710 else{
711 //temp_vpi=new ArrayList<>();
712 //temp_vpi.add(nom_variables.get(i));
713 gauche=nom_variables.get(i).get(0);
714 droite=nom_variables.get(i).get(1);
715 produit_cohesion=liste_cohesion.get(i);
716 int k=classifications.size()-1;
717     do{
718         if(k>-1){
719             if(contient(classifications.get(k).nom_variables,gauche)){
720                 temp_gauche=classifications.get(k).nom_variables;
721                 double prod=classifications.get(k).produit_cohesion;
722                 produit_cohesion=produit_cohesion*prod;
723                 gauche=classifications.get(k).nom;
724                 temp_index.add(k);
725                 temp_vpi.addAll(classification_vpi.get(k));
726             }
727             else{

```

```
728     k--;
729     }
730     }
731     else{
732     temp_gauche.add(gauche);
733     temp_index.add(-1);
734     }
735     }while(temp_gauche.isEmpty());
736     if(produit_cohesion>0){
737     k=classifications.size()-1;
738     do{
739     if(k>-1){
740     if(contient(classifications.get(k).nom_variables,droite)){
741     temp_droite=classifications.get(k).nom_variables;
742     if(!temp_droite.equals(temp_gauche)){
743     double prod=classifications.get(k).produit_cohesion;
744     produit_cohesion=produit_cohesion*prod;
745     droite=classifications.get(k).nom;
746     temp_index.add(k);
747     temp_vpi.addAll(classification_vpi.get(k));
748     }
749     else{
750     temp_droite=new ArrayList<String>();
751     temp_droite.add(droite);
752     temp_index.add(0);
753     produit_cohesion=0.0;
754     }
755     }
756     else{
757     k--;
758     }
759     }
760     else{
761     temp_droite.add(droite);
762     temp_index.add(-1);
763     }
764     }while(temp_droite.isEmpty());
765     nom=nom_classification(gauche, droite);
766     nom_vars=new ArrayList<String>();
767     nom_vars.addAll(temp_gauche);
```

```

768     nom_vars.addAll(temp_droite);
769     res_complement=complement_cohesion(nom_variables,liste_cohesion, temp_gauche
, temp_droite,nom_variables.get(i).get(0), nom_variables.get(i).get(1));
770 // System.out.println("gauche="+nom_variables.get(i).get(0)+" , droite="+
nom_variables.get(i).get(1));
771     for(int j=0;j<temp_gauche.size();j++){
772         for(int l=0;l<temp_droite.size();l++){
773 // if(!(temp_gauche.get(j).equals(gauche)&&temp_droite.get(l).equals(droite))){
774         ArrayList<String> temp_cpl=new ArrayList<>();
775         temp_cpl.add(temp_gauche.get(j));
776         temp_cpl.add(temp_droite.get(l));
777         if(!(nom_variables.get(i).equals(temp_cpl))){
778             if(rang(temp_vpi,temp_cpl)<0){
779                 int ind=rang(nom_variables,temp_cpl);
780                 if(ind>-1){
781                     temp_vpi.add(nom_variables.get(ind));
782                 }
783             }
784         }
785     }
786 }
787     produit_cohesion=produit_cohesion*res_complement;
788 }
789 }
790 if(produit_cohesion>0){
791     if(classifications.size()>0){
792         Classification temp_class_tri=new Classification(nom, nom_vars,produit_cohesion,
null,0);
793         if(temp_class_tri.moyenne_geometrique()>classifications.get(classifications.size
()-1).
794 moyenne_geometrique()){
795             double temp_c=liste_cohesion.get(i-1);
796             ArrayList<String> temp_n=nom_variables.get(i-1);
797             liste_cohesion.remove(i-1);
798             nom_variables.remove(i-1);
799             classification_vpi.remove(classification_vpi.size()-1);
800             liste_cohesion.add(i,temp_c);
801             nom_variables.add(i,temp_n);
802 //System.out.println("remaniement("+nom_variables.get(i).get(0)+" , "
803 +nom_variables.get(i).get(1)+")");

```

```
804 classifications.remove(classifications.size()-1);
805 index.remove(index.size()-1);
806 i--;
807 temp_class_tri=null;
808 produit_cohesion=0.0;
809 }
810 else{
811 Classification temp_class=new Classification(nom, nom_vars,produit_cohesion,
      null,0);
812 classifications.add(temp_class);
813 temp_vpi.add(nom_variables.get(i));
814 classification_vpi.add(temp_vpi);
815 // System.out.println("insertion "+ i+"("+nom_variables.get(i).get(0)+", "+
      nom_variables.get(i).get(1)+")");
816 index.add(temp_index);
817 temp_class=null;
818     produit_cohesion=0.0;
819     i++;
820 }
821 }
822 else{
823     Classification temp_class=new Classification(nom, nom_vars,produit_cohesion,
      null,0);
824     classifications.add(temp_class);
825     index.add(temp_index);
826     temp_class=null;
827     produit_cohesion=0.0;
828     temp_vpi.add(nom_variables.get(i));
829     classification_vpi.add(temp_vpi);
830     // System.out.println("insertion "+ i+"("+nom_variables.get(i).get(0)+", "+
      nom_variables.get(i).get(1)+")");
831     i++;
832 }
833 }
834 else{
835 double temp_c=liste_cohesion.get(i);
836 ArrayList<String> temp_n=nom_variables.get(i);
837 //ArrayList<ArrayList<String>> temp_vpi=classification_vpi.get(i);
838 liste_cohesion.remove(i);
839 nom_variables.remove(i);
```

```

840 // classification_vpi.remove(i);
841 liste_cohesion.add(liste_cohesion.size(),temp_c);
842 nom_variables.add(nom_variables.size(),temp_n);
843 //classification_vpi.add(classification_vpi.size(),temp_vpi);
844 // System.out.println("relegation "+ i+"("+nom_variables.get(i).get(0)+","+
      nom_variables.get(i).get(1)+")");
845     nombre++;
846     int test=liste_cohesion.size()-nombre;
847     if(i>=test){
848         i=liste_cohesion.size();
849     }
850     temp_c=0.0;
851 }
852 }while(i<liste_cohesion.size());
853 return classifications;
854 }
855 public ArrayList<ArrayList<String>> creer_couple_string(ArrayList<String>
      nom_variables){
856 ArrayList<ArrayList<String>> couples_variables=new ArrayList<>();
857 int i=0;
858 int j=0;
859 while(i<nom_variables.size()){
860 j=i+1;
861 while(j<nom_variables.size()){
862 ArrayList<String> temp=new ArrayList<String>();
863 temp.add(nom_variables.get(i));
864 temp.add(nom_variables.get(j));
865 couples_variables.add(temp);
866 j++;
867 }
868 i++;
869 }
870 return couples_variables;
871 }
872 //tableau des valeurs d'index
873 public int[] index_chercher(ArrayList<ArrayList<String>> liste_nom_cohesion,
      ArrayList<ArrayList<String>> couples){
874 int[] indexs=new int[couples.size()];
875 for(int i=0;i<couples.size();i++){
876 indexs[i]=rang(liste_nom_cohesion,couples.get(i));

```

```
877 }
878 return indexes;
879 }
880 //pour rechercher les variables qui se trouvent a gauche et qu'il ne faut pas
      compter
881 public int nb_gauche(int[] vbles,int index){
882 int nb=0;
883 int temp=vbles[index];
884 for(int i=0;i<vbles.length;i++){
885 if(temp<vbles[i]){
886 nb++;
887 }
888 }
889 return nb;
890 }
891 //somme a gauche d'un couple
892 public int somme_gauche(ArrayList<ArrayList<String>> liste_nom_cohesion,ArrayList
      <ArrayList<String>> couples,int card_total){
893 int[] vbles=index_chercher(liste_nom_cohesion, couples);
894 int[] sommes=new int[vbles.length];
895 int somme=0;
896 for(int i=0;i<vbles.length;i++){
897 int nb_g=nb_gauche(vbles,i);
898 sommes[i]=card_total-1-nb_g;
899 somme+=somes[i];
900 }
901 return somme;
902 }
903 //calcul des nombres de couples formes
904 public static int nb_couples(int nb_variables){
905 return(nb_variables*(nb_variables-1));
906 }
907 /*calcul du nombre de couples reunis
908 * classfictions: la liste des classifications
909 * index: index de la classification courante
910 */
911 public int rk(ArrayList<Classification> classfictions,int ind){
912 int nb=0;
913 nb=nb_couples(classfictions.get(ind).nom_variables.size())/2;
914 int[] mark=new int[ind+1];
```

```

915 for (int i=0;i<mark.length;i++){
916 mark[i]=1;
917 }
918 if(index.get(ind).get(0)>-1){
919 mark[index.get(ind).get(0)]=0;
920 }
921 if(index.get(ind).get(1)>-1){
922 mark[index.get(ind).get(1)]=0;
923 }
924 for(int j=ind-1;j>-1;j--){
925 if(mark[j]>0){
926 nb+=nb_couples(classfictions.get(j).nom_variables.size())/2;
927 }
928 if(index.get(j).get(0)>-1){
929 mark[index.get(j).get(0)]=0;
930 }
931 if(index.get(j).get(1)>-1){
932 mark[index.get(j).get(1)]=0;
933 }
934 }
935 return nb;
936 }
937 //calcul niveau significatif
938 public double niveau_significatif(ArrayList<Classification> classfictions,
          ArrayList<ArrayList<String>> liste_nom_cohesion,ArrayList<String>
          nom_variables,int nb_vbles_total,int index_class){
939 double retour=0.0;
940 int card_total=nb_couples(nb_vbles_total);
941 int r_k=rk(classfictions,index_class);
942 int s_k=card_total-r_k;
943 int skrk=r_k*s_k;
944 int s_gauche=somme_gauche(liste_nom_cohesion, creer_couple_string(nom_variables),
          card_total);
945 retour=(s_gauche-((skrk)/2))/((skrk*(card_total+1))/Math.sqrt(12));
946 //sortie.println(retour);
947 return retour;
948 }
949 ArrayList<ArrayList<Integer>> index;
950 //generer l'affichage de dendrogramme

```

```
951 public ArrayList<Classification> generer_dendogramme(ArrayList<ArrayList<String>>
    nom_variables,ArrayList<Classification> classifications,ArrayList<ArrayList<
    String>> nom_mgk,ArrayList<Double> liste_mgk){
952 /*
953 * cette fonction ajoute le "noeud" (composant graphique) a chaque classification
954 * elle calcule aussi la valeur de l'implication qui est affichee au dessus de
    chaque noeud dans le dendrogramme
955 */
956 ArrayList<ArrayList<String>> new_nom_variables=new ArrayList<ArrayList<String>>()
    ;
957 int nb_vbles_total=collecte_donnees.size();
958 for (int i=0;i<classifications.size();i++){
959 new_nom_variables.add(nom_variables.get(i));
960 }
961 ArrayList<String> liste_ordre=ordre_variables(new_nom_variables);
962 ArrayList<Point> points=axe_abscisse(liste_ordre);
963 liste_abscisse=points;
964 Point depart=null;
965 Point arrivee=null;
966         double implication=0.0;
967         int z=0;
968 //maximum du significatif
969 double max_sign=0;
970 int ind=0;
971 sortie.println(";"+" Regle "+";"+"cohesion inter-classe"+";"+"implication inter-
    classe;"+"valeur significative");
972 for(int j=0;j<classifications.size();j++){
973 int gauche=index.get(j).get(0);
974 int droite=index.get(j).get(1);
975 ArrayList<String> temp_gauche=new ArrayList<>();
976 ArrayList<String> temp_droite=new ArrayList<>();
977 double cohesion_gauche=0.0;
978 double cohesion_droite=0.0;
979 if(gauche!=-1){
980 ind=liste_ordre.indexOf(nom_variables.get(j).get(0));
981 temp_gauche.add(nom_variables.get(j).get(0));
982 depart=points.get(ind);
983 cohesion_gauche=1;
984 }
985 else{
```

```

986 depart=new Point("gauche_"+j,classifications.get(gauche).
987 noeud.milieu()),(int)Math.round(700*classifications.get(gauche).
    moyenne_geometrique()));
988 temp_gauche=classifications.get(gauche).nom_variables;
989 cohesion_gauche=classifications.get(gauche).moyenne_geometrique();
990 }
991 if(droite==-1){
992 ind=liste_ordre.indexOf(nom_variables.get(j).get(1));
993 temp_droite.add(nom_variables.get(j).get(1));
994 arrivee=points.get(ind);
995 cohesion_droite=1;
996 }
997 else{
998 arrivee=new Point("droite_"+j,classifications.get(droite).
999 noeud.milieu()),(int)Math.round(700*classifications.get(droite).
    moyenne_geometrique()));
1000 temp_droite=classifications.get(droite).nom_variables;
1001 cohesion_droite=classifications.get(droite).moyenne_geometrique();
1002 }
1003 implication=max_mgk(nom_mgk, liste_mgk, temp_gauche, temp_droite);
1004 implication=Math.pow(implication, temp_gauche.size()*temp_droite.size());
1005 double cohes=Math.sqrt(cohesion_droite*cohesion_gauche);
1006 implication=implication*cohes;
1007 z++;
1008 //ajout valeur significative
1009 double significatif=niveau_significatif(classifications, nom_variables,
    classifications.get(j).nom_variables, nb_vbles_total, j);
1010 classifications.get(j).set_valeur_significatif(significatif);
1011 if(max_sign<significatif){
1012 max_sign=significatif;
1013 }
1014 else{
1015 classifications.get(j-1).noeud.set_significatif();
1016 max_sign=significatif;
1017 }
1018 DecimalFormat d=new DecimalFormat();
1019 d.setMaximumFractionDigits(3);
1020 //ajout noeud
1021 Noeud noeud=new Noeud(d.format(implication), depart, arrivee,false);
1022 classifications.get(j).set_noeud(noeud);

```

```
1023 depart=null;
1024 arrivee=null;
1025 sortie.println("R("+z+")="+";"+classifications.get(j).nom+";"
1026 +classifications.get(j).moyenne_geometrique()+";"+implication+";"
1027 +classifications.get(j).valeur_significatif);
1028 }
1029 return classifications;
1030 }
1031 //creation nom des classifications
1032 private String nom_classification(String gauche,String droite){
1033 /*
1034 * exemple:
1035 * gauche=V1
1036 * droite=V2_non
1037 * retour=(V1,V2_non)
1038 */
1039 String retour="";
1040 retour="("+gauche+", "+droite+")";
1041 return retour;
1042 }
1043 //creations des points d'abscisses
1044 private ArrayList<Point> axe_abscisse(ArrayList<String> variables){
1045 /*
1046 * variables:tableau des noms de variables a afficher dans l'abscisse du
1047 * dendogramme
1048 */
1049 ArrayList<Point> points=new ArrayList<Point>();
1050 int width=(int) java.awt.Toolkit.getDefaultToolkit().getScreenSize().getWidth();
1051 width=width-100;
1052 //int pas=(int)Math.round((width-105)/variables.size());
1053 int pas=150;
1054 for(int i=0;i<variables.size();i++){
1055 Point temp=new Point(variables.get(i),40+((i+1)*pas),700);
1056 points.add(temp);
1057 }
1058 return points;
1059 }
1060 //tester si une variable appartient a un niveau
1061 public static boolean contient(ArrayList<String> tableau,String valeur){
1062 return tableau.contains(valeur);
1063 }
```

```
1062 }
1063 //identifier le niveau d'une variable
1064 public static ArrayList<String> ajouter_variable(ArrayList<ArrayList<String>>
    tableau,String valeur){
1065 ArrayList<String> retour=new ArrayList<String>();
1066 int k=0;
1067 do{
1068 ArrayList<String> temp=tableau.get(k);
1069 if(contient(temp,valeur)){
1070 retour=temp;
1071 tableau.remove(k);
1072 k=tableau.size()+1;
1073 }
1074 else{
1075 k++;
1076 }
1077 }while(k<tableau.size());
1078 if(retour.isEmpty()){
1079 retour.add(valeur);
1080 }
1081 return retour;
1082 }
1083 //arranger en ordre les variables pour le dendogramme
1084 private ArrayList<String> ordre_variables(ArrayList<ArrayList<String>>
    temp_classification){
1085 ArrayList<String> retour=new ArrayList<String>();
1086 ArrayList<ArrayList<String>> temp_tableau=new ArrayList<ArrayList<String>>();
1087 for(int i=0;i<temp_classification.size();i++){
1088 ArrayList<String> temp_gauche=new ArrayList<String>();
1089 ArrayList<String> temp_droite=new ArrayList<String>();
1090 if(temp_tableau.isEmpty()){
1091 temp_gauche.add(temp_classification.get(i).get(0));
1092 temp_droite.add(temp_classification.get(i).get(1));
1093 }
1094 else{
1095 temp_gauche=ajouter_variable(temp_tableau, temp_classification.get(i).get(0));
1096 if(temp_tableau.isEmpty()){
1097 temp_droite.add(temp_classification.get(i).get(1));
1098 }
1099 else{
```

```
1100 temp_droite=ajouter_variable(temp_tableau, temp_classification.get(i).get(1));
1101 }
1102 }
1103 if(!temp_gauche.contains(temp_classification.get(i).get(1))){
1104 temp_gauche.addAll(temp_droite);
1105 }
1106 temp_tableau.add(temp_gauche);
1107 temp_gauche=null;
1108 temp_droite=null;
1109 }
1110 if(temp_tableau.size(>1){
1111 Iterator<ArrayList<String>> it=temp_tableau.iterator();
1112 while(it.hasNext()){
1113 retour.addAll(it.next());
1114 }
1115 }
1116 else{
1117 retour=temp_tableau.get(0);
1118 }
1119 temp_tableau=null;
1120 return retour;
1121 }
1122 private ArrayList<Double> liste_cohesion_triee=null;
1123 //trier par ordre decroissant les couples a partir de la valeur de leur cohesion
1124 private void trier_cohesion(ArrayList<Double> liste_cohesion,ArrayList<ArrayList<
    String>> liste_nom){
1125 for (int i=0;i<liste_cohesion.size()-1;i++){
1126 if(liste_cohesion.get(i)==0.0){
1127 liste_cohesion.remove(i);
1128 liste_nom.remove(i);
1129 }
1130 }
1131 Boolean permut=false;
1132 do{
1133 permut=false;
1134 for (int i=0;i<liste_cohesion.size()-1;i++){
1135 if(liste_cohesion.get(i)<liste_cohesion.get(i+1)){
1136 Double temp_cohesion=liste_cohesion.get(i);
1137 ArrayList<String> nom=liste_nom.get(i);
1138 liste_cohesion.remove(i);
```

```

1139 liste_nom.remove(i);
1140 liste_cohesion.add(i+1,temp_cohesion);
1141 liste_nom.add(i+1,nom);
1142 permut=true;
1143 }
1144 }
1145 }while(permut==true);
1146 permut=null;
1147 }
1148 ArrayList<ArrayList<String>> nom_couples_tries;
1149 //liste des cohesions obtenues a partir du tableau genere data_cohesion
1150 private ArrayList<Double> liste_cohesion(Object[] [] temp_data_cohesion,String[]
        nom_variables){
1151 nom_couples_tries=new ArrayList<ArrayList<String>>();
1152 Object[] [] data_cohesion=temp_data_cohesion;
1153 ArrayList<Double> liste=new ArrayList<Double>();
1154 int length=data_cohesion.length;
1155 for(int i=0;i<length;i++){
1156 for(int j=1;j<length+1;j++){
1157 if(j!=(i+1)){
1158 if(!data_cohesion[i][j].equals("")){
1159 double temp=Double.parseDouble(((String)data_cohesion[i][j]).replace(",","."));
1160 data_cohesion[i][j]="";
1161 double temp_2=0.0;
1162 if(!data_cohesion[j-1][i+1].equals("")){
1163 temp_2=Double.parseDouble(((String)data_cohesion[j-1][i+1]).replace(",","."));
1164 }
1165 data_cohesion[j-1][i+1]="";
1166 ArrayList<String> temp_nom=new ArrayList<String>();
1167 if(temp>temp_2){
1168 if(temp>0){
1169 liste.add(temp);
1170 temp_nom.add((String)data_cohesion[i][0]);
1171 temp_nom.add(nom_variables[j]);
1172 nom_couples_tries.add(temp_nom);
1173 }
1174 }
1175 else{
1176 if(temp_2>0){
1177 liste.add(temp_2);

```

```
1178 temp_nom.add((String)data_cohesion[j-1][0]);
1179 temp_nom.add(nom_variables[i+1]);
1180 nom_couples_tries.add(temp_nom);
1181 }
1182 }
1183 }
1184 }
1185 }
1186 }
1187 return liste;
1188 }
1189 ArrayList<Double> liste_MGK;
1190 ArrayList<Double> liste_mgk;
1191     ArrayList<ArrayList<String>> nom_mgk;
1192     ArrayList<ArrayList> result_mgk;
1193 //liste MGK pour le dendrogramme
1194 private ArrayList<ArrayList> liste_mgk_triee(ArrayList<Double> liste_mgk_initial)
    {
1195 ArrayList<ArrayList> liste=new ArrayList<ArrayList>();
1196 ArrayList<ArrayList<String>> nom_couples=new ArrayList<ArrayList<String>>();
1197     ArrayList<Double> liste_mgk=new ArrayList<Double>();
1198     Iterator<Couple> it=couple_organise.iterator();
1199     for(int i=0;i<liste_mgk_initial.size()-1;i=i+2){
1200         Couple temp=it.next();
1201         ArrayList<String> temp_couples=new ArrayList<String>();
1202         temp_couples.add(temp.variable1.nom);
1203         temp_couples.add(temp.variable2.nom);
1204         nom_couples.add(temp_couples);
1205         if(liste_mgk_initial.get(i)>liste_mgk_initial.get(i+1)){
1206             liste_mgk.add(liste_mgk_initial.get(i));
1207         }
1208         else{
1209             liste_mgk.add(liste_mgk_initial.get(i+1));
1210         }
1211     }
1212     liste.add(nom_couples);
1213     liste.add(liste_mgk);
1214 return liste;
1215 }
1216 //tableau supp_mgk
```

```

1217 private Object[] [] tableau_supp_mgk(Object[] [] data_mgk,String[] nom_variables){
1218 Object[] [] tab_supp=data_mgk;
1219 Object[] [] tab_suppx=tableau_mgk(liste_suppx, nom_variables);
1220 Object[] [] tab_suppy=tableau_mgk(liste_suppy, nom_variables);
1221 DecimalFormat format=new DecimalFormat();
1222 format.setMaximumFractionDigits(3);
1223 //System.out.println("X"+" ;"+"Y"+" ;"+"supp(X) "+" ;"+"supp(Y) "+"MGK"+" ;"
1224 +"supp(X-->Y) "+" ;"+"supp(n) (X-->Y) "+" ;"+"conf (X-->Y)");
1225 sortie.println("X"+" ;"+"Y"+" ;"+"supp(X) "+" ;"+"supp(Y) "+" ;"+"MGK(X-->Y)-Valides"+"
        ;"
1226 +"supp(X-->Y) "+" ;"+"supp(n) (X-->Y) "+" ;"+"conf (X-->Y)");
1227 for(int i=0 ;i<data_mgk.length;i++){
1228 int j=1 ;
1229 while(j<data_mgk.length+1){
1230 double mgk=0.0 ;
1231 double suppx=0.0 ;
1232 double suppy=0.0 ;
1233 double supp=0.0 ;
1234 double supp_n=0.0 ;
1235 if( !data_mgk[i] [j] .equals("")){
1236 mgk=Double.parseDouble(((String) data_mgk[i] [j]).replace(",","."));
1237 suppx=Double.parseDouble(((String) tab_suppx[i] [j]).replace(",","."));
1238 suppy=Double.parseDouble(((String) tab_suppy[i] [j]).replace(",","."));
1239 supp=suppx*((1-suppy)*mgk+suppy);
1240 supp_n=(supp-suppx*suppy)/(suppx*(1-suppy));
1241 tab_supp[i] [j]=format.format(supp);
1242 //i+1 a remplacer par nom couple
1243 sortie.println(data_mgk[i] [0]+" ;"+nom_variables[j]+" ;"+suppx+" ;"+suppy+" ;"
1244 +data_mgk[i] [j]+" ;"+supp+" ;"+supp_n+" ;"+(supp/suppx));
1245 }
1246 j++;
1247 }
1248 }
1249 return tab_supp;
1250 }
1251 //tableau de cohesion
1252 private Object[] [] tableau_cohesion(Object[] [] data_MGK_t){
1253 Object[] [] sortie=data_MGK_t;
1254 DecimalFormat format=new DecimalFormat();
1255 format.setMaximumFractionDigits(3);

```

```
1256 for(int i=0;i<data_MGK_t.length;i++){
1257     int j=1;
1258     while(j<data_MGK_t.length+1){
1259         if(!data_MGK_t[i][j].equals("")){
1260             double cohesion=Double.parseDouble(((String) data_MGK_t[i][j]).replace(",","."))
1261             ;
1262             sortie[i][j]=format.format(calcul_cohesion(calcul_entropy(cohesion)));
1263         }
1264         j++;
1265     }
1266     return sortie;
1267 }
1268 //tableau MGK
1269 public Object [][] tableau_mgk(ArrayList<Double> liste_mgk,String[] nom_variables)
1270     {
1271     Object [][] retour=new Object[nom_variables.length-1][nom_variables.length];
1272     ArrayList<Double> liste_temp_mgk=liste_mgk;
1273     Iterator<Double> it=liste_temp_mgk.iterator();
1274     for(int ligne=0;ligne<nom_variables.length-1;ligne++){
1275         retour[ligne][0]=nom_variables[ligne+1];
1276         retour[ligne][ligne+1]="";
1277         for(int colonne=1;colonne<nom_variables.length;colonne++){
1278             DecimalFormat format=new DecimalFormat();
1279             format.setMaximumFractionDigits(3);
1280             String str="";
1281             double temp=0.0;
1282             if(retour[ligne][colonne]==null){
1283                 if(it.hasNext()){
1284                     temp=it.next();
1285                 }
1286                 if(temp>0){
1287                     str=format.format(temp);
1288                 }
1289                 else{
1290                     str="";
1291                 }
1292                 retour[ligne][colonne]=str;
1293                 if(it.hasNext()){
1294                     temp=it.next();
1295                 }
1296             }
1297         }
1298     }
1299 }
```

```

1294 }
1295 if(temp>0){
1296 str=format.format(temp);
1297 }
1298 else{
1299 str="";
1300 }
1301 retour[colonne-1][ligne+1]=str;
1302 }
1303 }
1304 }
1305 return retour;
1306 }
1307 //différents calculs concernant les couples
1308 private void afficher_calcul(ArrayList<Couple> liste_couples, JTextArea affichage)
    {
1309 affichage.setText("");
1310 ArrayList<Double> liste_MGK=calcul_MGK(liste_couples,0,false);
1311 DecimalFormat txt=new DecimalFormat();
1312 txt.setMaximumFractionDigits(3);
1313 String t="";
1314 // sortie.println(" * * * * * ");
1315 //sortie.println(" CONFIANCE - SUPPORT - CORRELATION - SIMILARITE - khi2 - MGK -
        ENTROPIE - COHESION DE CHAQUE COUPLE DE VARIABLES ");
1316 //sortie.println(" * * * * * ");
1317 for(int i=0;i<liste_couples.size();i++)
1318 {
1319 Couple temp_couple=liste_couples.get(i);
1320 double temp_mgk=liste_MGK.get(2*i);
1321 double temp_mgk_2=liste_MGK.get(2*i+1);
1322 affichage.append("\n \n");
1323 //sortie.println("(" + temp_couple.variable1.nom + ", " + temp_couple.variable2.nom + ")")
        ;
1324 affichage.append("(" + temp_couple.variable1.nom + ", " + temp_couple.variable2.nom + ") \
        n");
1325 t=txt.format(temp_couple.confiance());
1326 //sortie.println("confiance;" + t);
1327 affichage.append("confiance=" + t + "\n");
1328 t=txt.format(temp_couple.suppl_y());
1329 //sortie.println("support;" + t);

```

```
1330 affichage.append("support="+t+"\n");
1331 t=txt.format(temp_couple.correlation());
1332 //sortie.println("correlation;"+t);
1333 affichage.append("correlation="+t+"\n");
1334 t=txt.format(temp_couple.similarite());
1335 //sortie.println("similarite;"+t);
1336 affichage.append("similarite="+t+"\n");
1337 t=txt.format(temp_couple.khi2());
1338 //sortie.println("khi2;"+t);
1339 affichage.append("khi2="+t+"\n");
1340 t=txt.format(temp_mgk);
1341 //sortie.println("MGK("+temp_couple.variable1.nom+", "+temp_couple.variable2.nom
    +");"+t);
1342 affichage.append("MGK("+temp_couple.variable1.nom+", "+temp_couple.variable2.nom+"
    )="+t+"\n");
1343 t=txt.format(temp_mgk_2);
1344 //sortie.println("MGK("+temp_couple.variable2.nom+", "+temp_couple.variable1.nom
    +");"+t);
1345 affichage.append("MGK("+temp_couple.variable2.nom+", "+temp_couple.variable1.nom+"
    )="+t+"\n");
1346 double temp_entropy=calcul_entropy(temp_mgk);
1347 double temp_entropy_2=calcul_entropy(temp_mgk_2);
1348 t=txt.format(temp_entropy);
1349 //sortie.println("entropie("+temp_couple.variable1.nom+", "
1350 +temp_couple.variable2.nom+");"+t);
1351 //affichage.append("entropie("+temp_couple.variable1.nom+", "
1352 +temp_couple.variable2.nom+)"="+t+"\n");
1353 t=txt.format(temp_entropy_2);
1354 //sortie.println("entropie("+temp_couple.variable2.nom+", "
1355 +temp_couple.variable1.nom+");"+t);
1356 //affichage.append("entropie("+temp_couple.variable2.nom+", "
1357 +temp_couple.variable1.nom+)"="+t+"\n");
1358 t=txt.format(calcul_cohesion(temp_entropy));
1359 //sortie.println("cohesion("+temp_couple.variable1.nom+", "
1360 +temp_couple.variable2.nom+");"+t);
1361 //affichage.append("cohesion("+temp_couple.variable1.nom+", "
1362 +temp_couple.variable2.nom+)"="+t+"\n");
1363 t=txt.format(calcul_cohesion(temp_entropy_2));
1364 //sortie.println("entropie("+temp_couple.variable2.nom+", "
1365 +temp_couple.variable1.nom+");"+t);
```

```

1366 //affichage.append("cohesion("+temp_couple.variable2.nom+", "
1367 +temp_couple.variable1.nom+")="+t+"\n");
1368 //sortie.println();
1369 }
1370 }
1371 private ArrayList<Variable> collecte_donnees=new ArrayList<Variable>();
1372 private ArrayList<Individu> collecte_individu=new ArrayList<Individu>();
1373 private String[] entete_valeur={};
1374 private Object[] [] data_valeur={};
1375 private String[] entete_variable={"", "occurence", "moyenne", "ecart-type"};
1376 private Object[] [] data_variable={};
1377 private Object[] [] data_individu;
1378 private Object[] entete_individu;
1379 @Override
1380 public void actionPerformed(ActionEvent e) {
1381 // TODO Auto-generated method stub
1382 //affichage du contenu de l'excel contenant les valeurs des variables
1383 if(e.getSource()==bouton_valeur){
1384 card.show(card_pan, "valeur");
1385 }
1386 //pour les individus
1387 else if(e.getSource()==bouton_individus){
1388 if(!collecte_individu.isEmpty()){
1389 data_individu=new Object[collecte_individu.size()][];
1390 for(int i=0;i<collecte_individu.size();i++){
1391 ArrayList<Couple_Individu> temp_cpl=new ArrayList<>();
1392 temp_cpl=collecte_individu.get(i).contribution();
1393 if(i==0){
1394 entete_individu=new Object[temp_cpl.size()+1];
1395 entete_individu[0]="";
1396 for(int k=0;k<temp_cpl.size();k++){
1397 entete_individu[k+1]=temp_cpl.get(k).nom();
1398 }
1399 }
1400 Object[] temp=new Object[temp_cpl.size()+1];
1401 temp[0]=collecte_individu.get(i).nom;
1402 for(int j=0;j<temp_cpl.size();j++){
1403 temp[j+1]=temp_cpl.get(j).valeur;
1404 }
1405 data_individu[i]=temp;

```

```
1406 }
1407 tab_individus=new JTable(new DefaultTableModel(data_individu,entete_individu));
1408 tab_individus.setRowHeight(30);
1409 tab_individus.setPreferredSize(new Dimension(tab_individus.getColumnCount()*163,
        tab_individus.getRowCount()*30));
1410 tab_individus.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
1411 scroll_individus.setViewportView(tab_individus);
1412 }
1413 card.show(card_pan, "individus");
1414 }
1415 //affichage du panel variables
1416 else if(e.getSource()==bouton_variables){
1417 if(!collecte_donnees.isEmpty()){
1418 //organisation dans un tableau de l'occurrence, de la moyenne et de l'ecart-type
        de chaque variable
1419 data_variable=new Object[entete_valeur.length-1] [] ;
1420 Iterator< Variable> it=collecte_donnees.iterator();
1421 int i=0 ;
1422 while(it.hasNext()){
1423 Variable var=it.next();
1424 Object [] obj={var.mon_nom(),var.occurrence(),var.moyenne(),var.ecart_type()};
1425 data_variable[i]=obj;
1426 i++;
1427 }
1428 tab_variables=new JTable(new DefaultTableModel(data_variable,entete_variable));
1429 tab_variables.setRowHeight(30);
1430 tab_variables.setPreferredSize(new Dimension(tab_variables.getColumnCount()*163,
        tab_variables.getRowCount()*30));
1431 tab_variables.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
1432 scroll_variables.setViewportView(tab_variables);
1433 }
1434 card.show(card_pan, "variables");
1435 }
1436 //affichage du panel couples
1437 else if(e.getSource()==bouton_couples){
1438 if(couple_organise.isEmpty()){
1439 if(!collecte_donnees.isEmpty()){
1440 couple_organise=organiser_couple(collecte_donnees);
1441 }
1442 }
```

```

1443 card.show(card_pan, "couples");
1444 }
1445 //affichage du panel tableau
1446 else if(e.getSource()==bouton_tableau){
1447 card.show(card_pan, "tableau");
1448 }
1449 //affichage du panel classification
1450 else if(e.getSource()==bouton_classification){
1451 card.show(card_pan, "classification");
1452 }
1453 //affichage des VPI
1454 else if(e.getSource()==bouton_vpi){
1455 //edit va contenir le document html
1456 String edit="";
1457 for(int i=0;i<classification_vpi.size();i++){
1458 //temp_vpi: les couples du niveau i
1459 ArrayList<ArrayList<String>> temp_vpi=classification_vpi.get(i);
1460 String n_vpi="";
1461 DecimalFormat d=new DecimalFormat();
1462 d.setMaximumFractionDigits(3);
1463 for(int n=0;n<temp_vpi.size();n++){
1464 n_vpi=n_vpi+"("+temp_vpi.get(n).get(0)+","+temp_vpi.get(n).get(1)+") ";
1465 }
1466 edit=edit+"<br>niveau "+(i+1)+" : "+n_vpi;
1467 sortie.println("niveau="+","+(i+1)+","n_vpi);
1468 //entete tableau VPI
1469 edit=edit+"<br><table><thead><th>individu</th><th>phi</th><th>distance empirique
      </th><th>distance ideale theorique</th><th>contribution</th></thead>";
1470 sortie.println("individu"+","+"phi"+","+"distance empirique"+","+"distance ideale
      theorique"+","+"contribution");
1471 double contr_groupe=0.0;
1472 int card_groupe=0;
1473 //affichage VPI
1474 for(int k=0;k<collecte_individu.size();k++){
1475 String temp="(";
1476 double d_emp=0.0;
1477 double d_th=0.0;
1478 double contr=0.0;
1479 //stockage des valeurs de couples dans temp_cpl et temp_cppl va contenir les noms
1480 ArrayList<Couple_Individu> temp_cpl=collecte_individu.get(k).contribution();

```

```
1481 ArrayList<ArrayList<String>> temp_cppl=new ArrayList<>();
1482 for(int l=0;l<temp_cpl.size();l++){
1483 temp_cppl.add(temp_cpl.get(l).nom);
1484 }
1485 //recherche des contributions de chaque tem_vpi
1486 //exemple pour (V1,(V2,V3)) on peut avoir (1,1,1) si tous les couples contribuent
1487 for(int m=0;m<temp_vpi.size();m++){
1488 int ind=rang(temp_cppl,temp_vpi.get(m));
1489 if(temp_cpl.get(ind).valeur==1.0){
1490 temp=temp+"1,";
1491 }
1492 int index=liste_nom_mgk.indexOf(temp_vpi.get(m));
1493 double mgk=liste_MGK.get(index);
1494 d_emp=d_emp+temp_cpl.get(ind).distance_empirique(mgk);
1495 d_th=d_th+temp_cpl.get(ind).distance_ideal(mgk);
1496 }
1497 if(d_th!=0){
1498 contr=d_emp/d_th;
1499 }
1500 if(temp.length()<2){
1501 temp="";
1502 }
1503 else{
1504 temp=temp.substring(0,temp.length()-1);
1505 temp=temp+"";
1506 contr_groupe=contr_groupe+contr;
1507 card_groupe++;
1508 }
1509 edit=edit+"<tr><td>"collecte_individu.get(k).nom+"</td><td>"temp+"</td><td>"
1510 +d.format(d_emp)+"</td><td>"d.format(d_th)+"</td><td>"d.format(contr)+"</td></
    tr>";
1511 sortie.println(collecte_individu.get(k).nom+";"temp+";"d.format(d_emp)+";"
1512 +d.format(d_th)+";"d.format(contr));
1513 }
1514 edit=edit+"</table>";
1515 edit=edit+"<br> contribution du groupe="+d.format(contr_groupe/card_groupe);
1516 sortie.println("groupe optimal"+";"d.format(card_groupe));
1517 sortie.println("contribution du groupe"+";"d.format(contr_groupe/card_groupe));
1518 }
1519 JFrame frame_vpi=new JFrame("affichage VPI");
```

```
1520 frame_vpi.setSize(800,600);
1521 frame_vpi.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
1522 JEditorPane pan_vpi=new JEditorPane("text/html", "");
1523 pan_vpi.setText(edit);
1524 JScrollPane scroll_vpi=new JScrollPane();
1525 scroll_vpi.setViewportViewView(pan_vpi);
1526 frame_vpi.getContentPane().add(scroll_vpi);
1527 frame_vpi.setLocationRelativeTo(null);
1528 frame_vpi.setVisible(true);
1529 }
1530 //chargement des donnees
1531 else if(e.getSource()==bouton_charger){
1532 //fenetre de dialogue pour le choix du fichier excel contenant les variables a
    traiter
1533 JFileChooser chargement=new JFileChooser();
1534 chargement.setDialogTitle("chargement des donnees");
1535 FileNameExtensionFilter filtre=new FileNameExtensionFilter("fichiers CSV","csv");
1536 chargement.addChoosableFileFilter(filtre);
1537 chargement.setAcceptAllFileFilterUsed(false);
1538 File fichier=null;
1539 BufferedReader buff=null;
1540 //ouverture de la boîte de dialogue de chargement de donnees
1541 int open=chargement.showOpenDialog(this);
1542 if(open==JFileChooser.APPROVE_OPTION){
1543 //lecture flux
1544 fichier=chargement.getSelectedFile();
1545 path = fichier.getPath().substring(0, fichier.getPath().lastIndexOf("\\")+ "\\");
1546 File nv_fichier = CreerFichier(path);
1547 try {
1548 sortie = new PrintWriter(new FileWriter(nv_fichier.getPath(), true));
1549 } catch (IOException e2) {
1550 e2.printStackTrace();
1551 }
1552 try {
1553 buff=new BufferedReader(new FileReader(fichier));
1554 } catch (FileNotFoundException e1) {
1555 e1.printStackTrace();
1556 }
1557 //collecte_donnees contient la liste de variables
1558 collecte_donnees=new ArrayList<Variable>();
```

```
1559 collecte_individu=new ArrayList<Individu>();
1560 sortie.println("* * * * * * * * * * * * * * * * * * * * * * *");
1561 sortie.println("                TABLEAU DES VALEURS                ");
1562 sortie.println("* * * * * * * * * * * * * * * * * * * * * * *");
1563 String value="";
1564 Object[] values={};
1565 Vector<Object []> temp=new Vector<Object []>();
1566 Vector<Object []> temp_individu=new Vector<>();
1567 int compt=0;
1568 try {
1569 while((value=buff.readLine())!=null){
1570 if(compt>0){
1571 values=value.split(";");
1572 temp.add(values);
1573 values=value.substring(value.indexOf(";")).split(";");
1574 temp_individu.add(values);
1575 }
1576 else{
1577 entete_valeur=value.split(";");
1578 }
1579 compt++;
1580 sortie.println(value);
1581 buff.close();
1582 values=temp.toArray();
1583 data_valeur=new Object[values.length] [];
1584 for(int i=0;i<values.length;i++){
1585 data_valeur[i]=(Object[]) values[i];
1586 }
1587 //collecte des variables
1588 for(int i=1;i<entete_valeur.length;i++){
1589 Integer[] reponse=new Integer[data_valeur.length];
1590 for(int j=0;j<data_valeur.length;j++){
1591 if(data_valeur[j][i].equals("")){
1592 reponse[j]=0;
1593 }
1594 else{
1595 reponse[j]=Integer.parseInt((String) data_valeur[j][i]);
1596 }
1597 }
1598 Variable var=new Variable(reponse, entete_valeur[i]);
```

```

1599 collecte_donnees.add(var);
1600 }
1601 //collecte des individus
1602 Object[] temp_2_individu=temp_individu.toArray();
1603 String[] temp_ent_ind=new String[entete_valeur.length-1];
1604 for (int i=1;i<entete_valeur.length; i++){
1605 temp_ent_ind[i-1]=entete_valeur[i];
1606 }
1607 for(int i=0;i<temp_2_individu.length;i++){
1608 Object[] temp_r=(Object[])temp_2_individu[i];
1609 Object[] reponse=new Object[temp_r.length-1];
1610 for(int j=1;j<temp_r.length;j++){
1611 reponse[j-1]=(Object)temp_r[j];
1612 }
1613 //System.out.println("valeur 1="+reponse[0]);
1614 collecte_individu.add(new Individu((String) data_valeur[i][0],temp_ent_ind,
    reponse));
1615 //collecte_individu.add(new Individu((String) data_valeur[i][0],entete_valeur,
    reponse));
1616 }
1617 tab_valeur=new JTable(new DefaultTableModel(data_valeur,entete_valeur));
1618 tab_valeur.setRowHeight(30);
1619 tab_valeur.setPreferredSize(new Dimension(tab_valeur.getColumnCount()*100,
    tab_valeur.getRowCount()*30));
1620 tab_valeur.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
1621 scroll_valeur.setViewportViewView(tab_valeur);
1622 }
1623 catch (IOException e1) {
1624 e1.printStackTrace();
1625 }
1626 //reinitialisation des variables globales
1627 classification_vpi=new ArrayList<>();
1628 classifications=new ArrayList<>();
1629 couple_organise=new ArrayList<Couple>();
1630 liste_cohesion_triee=new ArrayList<Double>();
1631 liste_MGK=new ArrayList<>();
1632 liste_nom_mgk=new ArrayList<>();
1633 liste_mgk=new ArrayList<>();
1634 nom_mgk=new ArrayList<ArrayList<String>>();
1635 nom_couples_tries=new ArrayList<ArrayList<String>>();

```

```
1636 text_calcul.setText("");
1637 text_couple.setText("");
1638 text_classification.setText("");
1639 text_couple_classification.setText("");
1640 text_significatif_classification.setText("");
1641 }
1642 }
1643 /*
1644 * exemple affichage:
1645 * V1V2:12
1646 * V1V2_non:3
1647 * V1_nonV2:4
1648 * V1_nonV2_non:9
1649 */
1650 else if(e.getSource()==bouton_generer_couple){
1651 if(!couple_organise.isEmpty()){
1652 afficher_couple(couple_organise,text_couple);
1653 }
1654 }
1655 //affichage des mgk,similarite,cohesion, entropie, ....
1656 else if(e.getSource()==bouton_generer_calcul){
1657 if(!couple_organise.isEmpty()){
1658 afficher_calcul(couple_organise, text_calcul);
1659 }
1660 }
1661 //generation des tableaux mgk et cohesion
1662 else if(e.getSource()==bouton_generer_tableau){
1663 text_couple_classification.setText("");
1664 text_classification.setText("");
1665 text_significatif_classification.setText("");
1666 if(couple_organise.isEmpty()){
1667 couple_organise=organiser_couple(collecte_donnees);
1668 }
1669 String[] nom_variables=nom_variables(collecte_donnees,false);
1670 String[] entete=nom_variables;
1671 double min=0.0;
1672 //recuperation du choix selectionne pour le facteur "risque"
1673 switch(combo_risk.getSelectedIndex())
1674 {
1675 case 0 :
```

```

1676 min=0.0;
1677 break;
1678 case 1 :
1679 min=10.83;
1680 break;
1681 case 2 :
1682 min=6.63;
1683 break;
1684 case 3 :
1685 min=3.84;
1686 break;
1687 case 4 :
1688 min=2.7;
1689 break;
1690 }
1691 liste_MGK=new ArrayList<Double>();
1692 String regle="";
1693 //choix pour la regle (confiance>support y)
1694 switch (combo_favorise.getSelectedIndex()){
1695 case 0 :
1696 liste_MGK=calcul_MGK(couple_organise,min,false);
1697 regle=favorise[0];
1698 break;
1699 case 1 :
1700 liste_MGK=calcul_MGK(couple_organise,min,true);
1701 regle=favorise[1];
1702 break;
1703 case 2 :
1704 liste_MGK=calcul_MGK_BRN(couple_organise, min);
1705 entete=nom_variables(collecte_donnees, true);
1706 regle=favorise[2];
1707 break;
1708 }
1709 sortie.println("risque;" + min + " / ; regle: " + regle);
1710 sortie.println(" * * * * *");
1711 sortie.println("          TABLEAU SUPP(n)MGK          ");
1712 sortie.println(" * * * * *");
1713 //liste de mgk pour le dendogramme
1714 result_mgk=liste_mgk_triee(liste_MGK);
1715 nom_mgk=result_mgk.get(0);

```

```
1716 liste_mgk=result_mgk.get(1);
1717 //affichage du tableau de mgk
1718 Object [][] data_MGK=tableau_mgk(liste_MGK, nom_variables);
1719 ecrire_tableau(entete, sortie);
1720 ecrire_data(data_MGK, sortie);
1721 tab_MGK=new JTable(new DefaultTableModel(data_MGK,entete));
1722 tab_MGK.setRowHeight(30);
1723 tab_MGK.setPreferredSize(new Dimension(tab_MGK.getColumnCount()*163,tab_MGK.
    getRowCount()*30));
1724 tab_MGK.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
1725 scroll_MGK.setViewportView(tab_MGK);
1726 System.out.println("tab_mgk"+liste_MGK.size());
1727 //affichage du tableau de cohesion
1728 Object [][] data_cohesion=data_MGK;
1729 switch(combo_entropy.getSelectedIndex()){
1730 case 0 :
1731 //pour tableau cohesion avec p=mgk
1732 data_cohesion=tableau_cohesion(data_cohesion);
1733 break;
1734 case 1 :
1735 //pour tableau cohesion avec p=supp_mgk
1736 Object [][] data_supp=tableau_supp_mgk(data_MGK,nom_variables);
1737 sortie.println("* * * * * * * * * * * * * * * * * * * * * *");
1738 sortie.println("          TABLEAU SUPP MGK          ");
1739 sortie.println("* * * * * * * * * * * * * * * * * * * * * *");
1740 ecrire_tableau(entete, sortie);
1741 ecrire_data(data_supp, sortie);
1742 data_cohesion=tableau_cohesion(data_supp);
1743 break;
1744 }
1745 sortie.println("* * * * * * * * * * * * * * * * * * * * * *");
1746 sortie.println("          TABLEAU COHESION_MGK          ");
1747 sortie.println("* * * * * * * * * * * * * * * * * * * * * *");
1748 ecrire_tableau(entete, sortie);
1749 ecrire_data(data_cohesion, sortie);
1750 tab_cohesion=new JTable(new DefaultTableModel(data_cohesion,entete));
1751 tab_cohesion.setRowHeight(30);
1752 tab_cohesion.setPreferredSize(new Dimension(tab_cohesion.getColumnCount()*163,
    tab_cohesion.getRowCount()*30));
1753 tab_cohesion.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
```

```

1754 scroll_cohesion.setViewportView(tab_cohesion);
1755 //liste de cohesion pour la classification
1756 liste_cohesion_triee=new ArrayList<Double>();
1757 liste_cohesion_triee=liste_cohesion(data_cohesion,entete);
1758 }
1759 //classification des couples
1760 else if(bouton_classifier==e.getSource()){
1761 DecimalFormat d=new DecimalFormat();
1762 d.setMaximumFractionDigits(3);
1763 String edit="<table><thead><th>niveau</th><th>s(&#937 ;,k)</th><th>v(&#937 ;,k)</
    th>";
1764 text_couple_classification.setText("");
1765 text_classification.setText("");
1766 text_significatif_classification.setText("");
1767 if(liste_cohesion_triee!=null && liste_cohesion_triee.size(>0){
1768 //trier de la liste de cohesion en ordre decroissante
1769 trier_cohesion(liste_cohesion_triee, nom_couples_tries);
1770 /affichage
1771 sortie.println(" * * * * *");
1772 sortie.println("          PREORDRE GLOBALE          ");
1773 sortie.println(" * * * * *");
1774 for(int i=0;i<liste_cohesion_triee.size();i++){
1775 text_couple_classification.append("cohesion("+nom_couples_tries.get(i).get(0)+", "
    +nom_couples_tries.get(i).get(1)+")"+" = "+liste_cohesion_triee.get(i)+"\n");
1776 sortie.println("cohesion("+nom_couples_tries.get(i).get(0)+", "+nom_couples_tries.
    get(i).get(1)+")"+" = "+";"+liste_cohesion_triee.get(i));
1777 }
1778 classifications=new ArrayList<Classification>();
1779 index =new ArrayList<ArrayList<Integer>>();
1780 //generation des classifications sans noeuds
1781 classifications=generer_classification(nom_couples_tries,liste_cohesion_triee,
    index);
1782 //generation des classification pour le dendrogramme
1783 classifications=generer_dendogramme(nom_couples_tries, classifications,nom_mgk,
    liste_mgk);
1784 //classement de classifications par niveau
1785 ArrayList<Classification> temp_classifications=new ArrayList<Classification>();
1786 text_classification.setText("");
1787 sortie.println(" * * * * *");
1788 sortie.println("          NIVEAU CLASSIFICATION          ");

```

```
1789 sortie.println("* * * * *");
1790 int niveau=1;
1791 for(int j=0;j<classifications.size();j++){
1792 if(j==0){
1793 text_classification.append("niveau - "+niveau+" "+"R("+niveau+" ) : ");
1794 text_classification.append(classifications.get(j).nom+" = "+classifications.get(j)
    ).moyenne_geometrique()+"\n");
1795 //text_classification.append("niveau significatif="+classifications.get(j).
    valeur_significatif+"\n");
1796 edit=edit+"<tr><td>1</td><td>" + d.format(classifications.get(j).
    valeur_significatif)+"</td><td>" + d.format(classifications.get(j).
    valeur_significatif)+"</td></tr>";
1797 sortie.println("niveau "+niveau+";"+classifications.get(j).nom+";"+
    classifications.get(j).moyenne_geometrique());
1798 niveau++;
1799 temp_classifications.add(classifications.get(j));
1800 }
1801 else{
1802 text_classification.append("niveau - "+niveau+" "+"R("+niveau+" ) : ");
1803 text_classification.append(classifications.get(j).nom+" = "+classifications.get(j)
    ).moyenne_geometrique()+"\n");
1804 //text_classification.append("niveau significatif="+classifications.get(j).
    valeur_significatif+"\n");
1805 double diff=classifications.get(j).valeur_significatif-classifications.get(j-1).
    valeur_significatif;
1806 edit=edit+"<tr><td>"+(j+1)+"</td><td>" + d.format(classifications.get(j).
    valeur_significatif)+"</td><td>" + d.format(diff)+"</td></tr>";
1807 sortie.println("niveau "+niveau+";"+classifications.get(j).nom+";"+
    classifications.get(j).moyenne_geometrique()+";"+diff);
1808 niveau++;
1809 temp_classifications.add(classifications.get(j));
1810 }
1811 }
1812 text_significatif_classification.setText(edit);
1813 }
1814 }
1815 //affichage du dendrogramme
1816 else if(e.getSource()==bouton_dendogramme){
1817 if(classifications!=null && classifications.size()>0){
1818 Dimension dimension=java.awt.Toolkit.getDefaultToolkit().getScreenSize();
```

```

1819 int height = (int)dimension.getHeight();
1820 int width = (int)dimension.getWidth();
1821 DendrogramPanel dendo=new DendrogramPanel(classifications,liste_abscisse);
1822 JFrame frame=new JFrame();
1823 frame.setSize(width-40,height);
1824 frame.getContentPane().setLayout(new BorderLayout());
1825 JScrollPane scroll=new JScrollPane(dendo);
1826 scroll.getViewPort().setPreferredSize(new Dimension(200+liste_abscisse.size()
    *160, height));
1827 frame.add(scroll,BorderLayout.CENTER);
1828 JButton but_den=new JButton("exporter");
1829 but_den.addActionListener(new ActionListener() {
1830 @Override
1831 public void actionPerformed(ActionEvent arg0) {
1832 // TODO Auto-generated method stub
1833 Image img=dendo.axe();
1834 try {
1835 ImageIO.write((RenderedImage)img, "png",CreerImage(path) );
1836 JOptionPane.showMessageDialog(null, "exportation effectuee!");
1837 } catch (IOException e) {
1838 // TODO Auto-generated catch block
1839 e.printStackTrace();
1840 }
1841 }
1842 });
1843 frame.add(but_den,BorderLayout.SOUTH);
1844 frame.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
1845 frame.setLocationRelativeTo(null);
1846 frame.setVisible(true);
1847 }
1848 }
1849 //fermeture du fichier de sortie excel
1850 else if(e.getSource()==bouton_exporter){
1851 sortie.close();
1852 JOptionPane.showMessageDialog(this,"Exportation effectuee!!");
1853 }
1854 }
1855 //tableau vers un tableau de string pour le fichier de sortie excel
1856 private void ecrire_data(Object[] [] data,PrintWriter write){
1857 for(int ligne=0 ;ligne<data.length;ligne++){

```

```
1858 Object[] obj=data[ligne];
1859 ecrire_tableau(obj, write);
1860 }
1861 }
1862 //tableau a deux dimensions en tableau de string pour le fichier de sortie excel
1863 private void ecrire_tableau(Object[] data,PrintWriter write){
1864 for(int colonne=0;colonne<data.length;colonne++){
1865 write.print(data[colonne]+";");
1866 }
1867 write.println();
1868 }
1869 //creation du fichier de sortie excel
1870 private static File CreerFichier(String path){
1871 int i = 0;
1872 String name = "SORTIE";
1873 boolean isOK = false;
1874 while(!isOK)
1875 {
1876     String name1 = name;
1877     if (i != 0) name1 = name+i;
1878     File f = new File(path+name1+".csv");
1879     if(f.exists()) i+=1;
1880     else
1881     {
1882         isOK = true;
1883         name = name1;
1884     }
1885 }
1886 File f = new File(path+name+".csv");
1887 try {
1888     f.createNewFile();
1889 } catch (IOException ex) {
1890     JOptionPane.showMessageDialog(null, "Impossible de creer un nouveau fichier")
1891     ;
1892 }
1893 return f;
1894 }
1895 private static File CreerImage(String path) {
1896 int i = 0;
1897 String name = "SORTIE";
```

```
1897 boolean isOK = false;
1898 while(!isOK)
1899 {
1900     String name1 = name;
1901     if (i != 0) name1 = name+i;
1902     File f = new File(path+name1+".png");
1903     if(f.exists()) i+=1;
1904     else
1905     {
1906         isOK = true;
1907         name = name1;
1908     }
1909 }
1910 File f = new File(path+name+".png");
1911 try {
1912     f.createNewFile();
1913 } catch (IOException ex) {
1914     JOptionPane.showMessageDialog(null, "Impossible d'exporter le dendogramme");
1915 }
1916 return f;
1917 }
1918 }
```

Listing B.1 – Classe *Fenetre_principale.java*

B.3 Classe utilisant pour les calculs en fonction de couple de variables

```
1 public class Couple {
2 public Variable variable1;
3 public Variable variable2;
4 /*
5 * couple de variables: (variable1,variable2)
6 */
7 public Couple(Variable variable1,Variable variable2){
8 this.variable1=variable1;
9 this.variable2=variable2;
10 }
11 //mgk du couple
12 public double mgk(){
13 double mgk=0.0;
14 double n_V1_oui=this.couple("11")+this.couple("10");
15 double n_V2_non=this.couple("10")+this.couple("00");
16 double n_V1_non=this.couple("01")+this.couple("00");
17 double n=n_V1_oui+n_V1_non;
18 mgk=1-(this.couple("10")/(n_V1_oui*n_V2_non))*n;
19 return mgk;
20 }
21 //mgk brn du couple
22 public double mgk_brn(){
23 double mgk=0.0;
24 int cple_a_b=this.couple("11");
25 int cple_a_b_=this.couple("10");
26 double n_a=cple_a_b+cple_a_b_;
27 int cple_a__b=this.couple("01");
28 double n_b=cple_a_b+cple_a__b;
29 double n_a_=cple_a_b+cple_a__b;
30 double n=n_a+n_a_;
31 mgk=1-(cple_a_b/(n_a*n_b))*n;
32 return mgk;
33 }
34 public int couple(String val){
35 //val=0 pour variable1=0 et variable2=0
36 //val=1 pour variable1=0 et variable2=1
```

```

37 //val=2 pour variable1=1 et variable2=0
38 //val=3 pour variable1=1 et variable2=1
39 int nb=0;
40 Integer[] tab_1=this.variable1.reponse;
41 Integer[] tab_2=this.variable2.reponse;
42 char a=val.charAt(0);
43 char b=val.charAt(1);
44 for(Integer i=0;i<tab_1.length;i++){
45 if(tab_1[i]==Character.getNumericValue(a) && tab_2[i]==Character.getNumericValue(
    b)){
46 nb++;
47 }
48 }
49     return nb;
50 }
51 //khi2 du couple
52 public double khi2(){
53 double k=0.0;
54 double n_v1=this.couple("11")+this.couple("10");
55 double n_v2=this.couple("11")+this.couple("01");
56 double n_v1_=this.couple("01")+this.couple("00");
57 double n_v2_=this.couple("10")+this.couple("00");
58 double n=n_v1+n_v1_;
59 k=n*Math.pow((((this.couple("11")*this.couple("00"))-(this.couple("01")*this.
    couple("10"))),2)/(n_v1*n_v1_*n_v2*n_v2_);
60 return k;
61 }
62 //correlation du couple
63 public double correlation(){
64 double cor=0.0;
65 double n_v1=this.couple("11")+this.couple("10");
66 double n_v2=this.couple("11")+this.couple("01");
67 double n_v1_=this.couple("01")+this.couple("00");
68 double n_v2_=this.couple("10")+this.couple("00");
69 double n=n_v1+n_v1_;
70 cor=((n*this.couple("11"))-(n_v1*n_v2))/Math.sqrt(n_v1*n_v1_*n_v2*n_v2_);
71 return cor;
72 }
73 //similarite du couple
74 public double similarite(){

```

```
75 double sim=0.0;
76 double n_v1=this.couple("11")+this.couple("10");
77 double n_v2=this.couple("11")+this.couple("01");
78 double n_v2_=this.couple("10")+this.couple("00");
79 double n=n_v2+n_v2_;
80 sim=(this.couple("11")-((n_v1*n_v2)/n))/Math.sqrt((n_v1*n_v2)/n);
81 return sim;
82 }
83 //confiance du couple
84 public double confiance(){
85 double conf=0.0;
86 double nx=(this.couple("11")+this.couple("10"));
87 if(nx>0){
88 conf=this.couple("11")/nx;
89 }
90 return conf;
91 }
92 //support de variable2
93 public double supp_y(){
94 double sup=0.0;
95 double ny=this.couple("11")+this.couple("01");
96 double ny_=this.couple("10")+this.couple("00");
97 double n=ny+ny_;
98 sup=ny/n;
99 return sup;
100 }
101 //support de variable1
102 public double supp_x(){
103 double sup=0.0;
104 double nx=this.couple("11")+this.couple("10");
105 double nx_=this.couple("01")+this.couple("00");
106 double n=nx+nx_;
107 sup=nx/n;
108 return sup;
109 }
110 //test de confiance>support y
111 public boolean favorise(){
112 return this.confiance()>this.supp_y();
113 }
114 //mgk critique du couple
```

```
115 public double mgk_critique(double khi_theorique){
116     double mgk=0.0;
117     double n_v1=this.couple("11")+this.couple("10");
118     double n_v2=this.couple("11")+this.couple("01");
119     double n_v2_=this.couple("10")+this.couple("00");
120     double n=n_v2+n_v2_;
121     mgk=Math.sqrt((khi_theorique*(n-n_v1)*n_v2)/(n*n_v1*(n-n_v2)));
122     return mgk;
123 }
124 }
```

Listing B.2 – Classe *Couple_variable.java*

B.4 Classe utilisant pour les calculs occurrence, moyenne et ecart-type

```
1 public class Variable {
2     public Integer[] reponse;
3     public String nom;
4     public Variable(Integer[] reponse,String nom){
5         this.reponse=reponse;
6         this.nom=nom;}
7     public Integer occurrence(){
8         Integer nb=0;
9         Integer index=0;
10        while (index<this.reponse.length) {
11            if(reponse[index]==1){
12                nb++;
13            }
14            index ++;
15        }
16        return nb;}
17    public Double moyenne(){
18        Double moy=0.0;
19        moy= (double)this.occurrence()/this.reponse.length;
20        return moy;}
21    public Double ecart_type(){
22        Double ecart=0.0;
23        Double moy=this.moyenne();
24        Double occ=(double)this.occurrence();
25        ecart=Math.sqrt((occ*Math.pow((1-moy),2)+(this.reponse.length-occ)*Math.pow(moy
26            , 2))/(this.reponse.length-1));
27        return ecart;}
28    public String mon_nom(){
29        return this.nom;}
30 }
```

Listing B.3 – Classe *Variable.java*

B.5 Classe de la classification

```

1 import java.awt.Graphics;
2 import java.util.ArrayList;
3 public class Classification {
4     String nom; //exemple (V1,(V2,V3))
5     ArrayList<String> nom_variables; //exemple:{V1,V2,V3}
6     double produit_cohesion;//exemple:max_cohesion(V1,V2)*max_cohesion(V1,V3)*
7         max_cohesion(V2,V3)
8     Noeud noeud;//composant graphique a représenter dans le dendrogramme
9     double valeur_significatif;
10    boolean significatif;
11    String vpi;
12    Classification(String nom,ArrayList<String> nom_variables,double
13        produit_cohesion,Noeud noeud,double valeur_significatif){
14        this.nom=nom;
15        this.nom_variables=nom_variables;
16        this.produit_cohesion=produit_cohesion;
17        this.noeud=noeud;
18        this.valeur_significatif=valeur_significatif;
19    }
20    //affichage dans le dendrogramme
21    public void tracer(Graphics g,double cohesion,int max){
22        noeud.tracer(g,(int)Math.round(cohesion*max));
23    }
24    //moyenne geometrique
25    public double moyenne_geometrique(){
26        double m=0.0;
27        double den=(double)2/(nom_variables.size()*(nom_variables.size()-1));
28        m=Math.pow(produit_cohesion, den);
29        return m;
30    }
31    //changement de la valeur du noeud
32    public void set_noeud(Noeud noeud){
33        this.noeud=noeud;
34    }
35    //changement de la valeur significative
36    public void set_valeur_significative(double sign){

```

```
37     this.valeur_significatif=sign;
38 }
39
40 //vecteur puissance implicative
41 public void set_vpi(String vpi){
42     this.vpi=vpi;
43 }
44 }
```

Listing B.4 – Classe *Classification.java*

B.6 Classe pour tracer le dendrogramme

```

1 import java.awt.Color;
2 import java.awt.Dimension;
3 import java.awt.Graphics;
4 import java.awt.Image;
5 import java.awt.Rectangle;
6 import java.awt.image.BufferedImage;
7 import java.util.ArrayList;
8 import java.util.Iterator;
9 import javax.swing.JPanel;
10 import javax.swing.Scrollable;
11 class DendrogramPanel extends JPanel implements Scrollable
12 {
13     private static final long serialVersionUID = 1L;
14     ArrayList<Classification> dendogramme;
15     ArrayList<Point> variables;
16     Dimension dim= java.awt.Toolkit.getDefaultToolkit().getScreenSize();
17     public DendrogramPanel(ArrayList<Classification> dendogramme,ArrayList<
18     Point> variables){
19         super();
20         this.dendogramme=dendogramme;
21         this.variables=variables;
22         this.setPreferredSize(new Dimension(60+variables.size()*151,(int) dim.
23         getHeight()));
24     }
25     //affichage dans le panel
26     public void paint(Graphics g){
27         super.paintComponent(g);
28         //g.setColor(Color.BLACK);
29         Image img=axe();
30         g.drawImage(img,10,10,this);
31     }
32     //axe du dendogramme
33     public Image axe(){
34         BufferedImage buff=new BufferedImage(60+variables.size()*151, (int) dim.
35         getHeight()+10,BufferedImage.TYPE_INT_ARGB);
36         Graphics g=buff.getGraphics();
37         ordonnee(g);
38         abscisse(g);

```

```
36     tracer(g);
37     return buff;
38 }
39 //axe des ordonnees
40 public void ordonnee(Graphics g){
41     g.setColor(Color.black);
42     g.drawLine(50, 50, 50,700 );
43     double pas=(700-60)/4;
44     double max=0.75;
45     for(int i=1;i<=4;i++){
46         Point p=new Point(Double.toString(max*(5-i)/4), 50, (int) Math.round(700-
47         pas*i));
48         p.dessiner_nom(g,p.x-40,p.y);
49         p.dessiner_point(g);
50     }
51 }
52 //axe des abscisses
53 public void abscisse(Graphics g){
54     Iterator<Point> it=this.variables.iterator();
55     while(it.hasNext()){
56         Point temp=it.next();
57         temp.dessiner_point(g);
58         temp.dessiner_nom(g, temp.x-1, temp.y+20);
59     }
60 }
61 //tracage du dendrogramme
62 public void tracer(Graphics g){
63     Iterator<Classification> it=this.dendogramme.iterator();
64     while(it.hasNext()){
65         Classification temp=it.next();
66         double cohesion=temp.moyenne_geometrique();
67         temp.tracer(g,cohesion,700);
68     }
69 }
70 @Override
71 public Dimension getPreferredSize() {
72     return getPreferredSize();
73 }
74 public int getScrollableUnitIncrement(Rectangle visibleRect,
```

```
75     int orientation, int direction) {
76     return variables.size();
77 }
78 public int getScrollableBlockIncrement(Rectangle visibleRect,
79     int orientation, int direction) {
80     return variables.size()*100;
81 }
82 public boolean getScrollableTracksViewportWidth() {
83     return false;
84 }
85 public boolean getScrollableTracksViewportHeight() {
86     return false;
87 }
88 }
```

Listing B.5 – Classe *DendrogrammePanel.java*

B.7 Classe pour dessiner les points du dendrogramme

```
1 import java.awt.Graphics;
2 public class Point {
3     int x;
4     int y;
5     String nom;
6     public Point(String nom,int x,int y){
7         this.x=x;
8         this.y=y;
9         this.nom=nom;
10    }
11    public void dessiner_nom(Graphics g,int x1,int y1){
12        g.drawString(this.nom, x1, y1);
13    }
14    public void dessiner_point(Graphics g){
15        g.drawRect(this.x-1, this.y-1, 2, 2);
16    }
17    public void set_y(int y){
18        this.y=y;
19    }
20 }
```

Listing B.6 – Classe *Point.java*

B.8 Classe pour tracer les noeuds du dendrogramme

```
1 import java.awt.Color;
2 import java.awt.Graphics;
3 public class Noeud {
4     Point debut;
5     Point fin;
6     String implication;
7     boolean significatif;
8     public Noeud(String implication,Point debut,Point fin,boolean significatif){
9         this.implication=implication;
10        this.debut=debut;
11        this.fin=fin;
12        this.significatif=significatif;
13    }
14    public void tracer(Graphics g,int y_dest){
15        String nom1="temp";
16        String nom2="temp2";
17            boolean gauche=true;
18            g.setColor(Color.black);
19        if(debut.x<fin.x){
20            nom1=implication;
21        }
22        else{
23            nom2=implication;
24            gauche=false;
25        }
26        Point temp=new Point(nom1, debut.x, y_dest);
27        Point temp2=new Point(nom2+fin.nom, fin.x, y_dest);
28        debut.dessiner_point(g);
29        g.drawLine(debut.x, debut.y,temp.x, temp.y);
30        fleche(g,temp,temp2);
31        g.setColor(Color.black);
32        g.drawLine(temp2.x,temp2.y,fin.x,fin.y);
33        fin.dessiner_point(g);
34            if(gauche){
35                temp.dessiner_nom(g, temp.x, temp.y-2);
36            }
37            else{
38                temp2.dessiner_nom(g, temp.x, temp.y-2);
```

```
39     }
40     if(this.significatif){
41         g.setColor(Color.blue);
42         etoile(g, new Point("etoile",milieu(),temp.y));
43     }
44     g.setColor(Color.black);
45 }
46 public int milieu(){
47     int abscisse=0;
48     int pas=Math.abs(debut.x-fin.x);
49     pas=pas/2;
50     if(debut.x<fin.x){
51         abscisse=debut.x+pas;
52     }
53     else{
54         abscisse=fin.x+pas;
55     }
56     return abscisse;
57 }
58 public void set_depart(Point depart){
59     this.debut=depart;
60 }
61 public void set_arrivee(Point arrivee){
62     this.fin=arrivee;
63 }
64 public void gauche(Graphics g,Point point){
65     g.drawLine(point.x,point.y, point.x+10, point.y-5);
66     g.drawLine(point.x,point.y, point.x+10, point.y+5);
67 }
68 public void droite(Graphics g,Point point){
69     g.drawLine(point.x,point.y, point.x-10, point.y-5);
70     g.drawLine(point.x,point.y, point.x-10, point.y+5);
71 }
72 public void ligne(Graphics g,Point depart,Point arrivee){
73     g.drawLine(depart.x, depart.y, arrivee.x, arrivee.y);
74 }
75 public void fleche(Graphics g,Point depart,Point arrivee){
76     ligne(g, depart, arrivee);
77     if(depart.x<arrivee.x){
78         g.setColor(Color.red);
```

```
79     droite(g,arrivee);
80 }
81 else{
82     g.setColor(Color.red);
83     gauche(g,arrivee);
84 }
85 }
86 public void etoile(Graphics g,Point reference){
87     g.drawLine(reference.x+5, reference.y+5, reference.x-5, reference.y-5);
88     g.drawLine(reference.x-5, reference.y+5, reference.x+5, reference.y-5);
89     g.drawLine(reference.x, reference.y+5, reference.x, reference.y-5);
90 }
91 public void set_significatif(){
92     this.significatif=true;
93 }
94 }
```

Listing B.7 – Classe *Noeud.java*

B.9 Organisation des individus

```
1 import java.util.ArrayList;
2 public class Individu {
3     String nom;
4     String[] variables;
5     Object[] reponse;
6     public Individu(String nom,String[] variables,Object[] reponse) {
7         this.nom=nom;
8         this.variables=variables;
9         this.reponse=reponse;
10    }
11    public Integer occurrence(){
12        Integer nb=0 ;
13        Integer index=0 ;
14        while (index<this.reponse.length) {
15            if((Integer)reponse[index]==1){
16                nb++;
17            }
18            index ++;
19        }
20        return nb;
21    }
22    public double participation(){
23        return this.occurrence()/this.reponse.length;
24    }
25    public ArrayList<Couple_Individu> contribution(){
26        ArrayList<Couple_Individu> res=new ArrayList<Couple_Individu>();
27        for(int i=0 ; i<reponse.length;i++){
28            for(int j=0 ;j<reponse.length;j++){
29                if(i!=j){
30                    double valeur=0 ;
31                    if(Integer.valueOf((String)reponse[j])==1){
32                        valeur=1 ;
33                    }
34                    else if(Integer.valueOf((String)reponse[i])==0){
35                        valeur=0.5 ;
36                    }
37                    ArrayList<String> temp_nom=new ArrayList<String>();
38                    temp_nom.add(this.variables[i]);
```

```
39         temp_nom.add(this.variables[j]);
40 Couple_Individu temp=new Couple_Individu(temp_nom, valeur);
41         res.add(temp);
42     }
43 }
44 }
45         return res;
46 }
47
48 }
```

Listing B.8 – Classe *Individu.java*

B.10 Calcul concernant les individus

```
1 import java.util.ArrayList;
2 public class Couple_Individu {
3     ArrayList<String> nom;
4     double valeur;
5     //double mgk=0.0;
6     public Couple_Individu(ArrayList<String> nom, double valeur){
7         this.nom=nom;
8         this.valeur=valeur;
9     //this.mgk=mgk;
10    }
11    public String nom(){
12        return "("+nom.get(0)+","+nom.get(1)+")";
13    }
14    /*public void setMGK(double mgk){
15        this.mgk=mgk;
16    }*/
17    //distance empirique
18    public double distance_empirique(double mgk){
19        double d=0.0;
20        double num=0.0;
21        double den=0.0;
22        if(mgk!=1){
23            //formule
24            //d(i,couple)=racine_carre((MGK-valeur)^2/(1-MGK))
25            num=Math.pow(mgk-valeur,2);
26            den=1-mgk;
27            d=Math.sqrt(num/den);
28        }
29        return d;
30    }
31    //distance ideale theorique
32    public double distance_ideal(double mgk){
33        double d=0.0;
34        double num=0.0;
35        double den=0.0;
36        if(mgk!=1){
37            //formule
38            //d(i,couple)=racine_carre((MGK-valeur)^2/(1-MGK))
```

```
39     num=Math.pow(mgk-1,2);
40     den=1-mgk;
41     d=Math.sqrt(num/den);
42
43 }
44 return d;
45 }
46 }
```

Listing B.9 – Classe *Couple_individu.java*

B.11 Résultats 1 : §4.4.1, tab 4.3, tab 4.4

V1	V2	non_V2	khi2	4.47441966702028	support(V1)	0.566666666666667		
non_V1	8	9	support(V2)	0.633333333333333	similarite	-0.8431720539054295		
			confiance(V1,V2)	0.4705882329411764	correlation	-0.3861969258768566		
V1	V3	non_V3	khi2	3.32579185520362	support(V1)	0.566666666666667		
non_V1	2	11	support(V3)	0.333333333333333	similarite	0.9801960588196067		
			confiance(V1,V3)	0.4705882329411764	correlation	0.33295604590814587		
V1	V4	non_V4	khi2	3.32579185520362	support(V1)	0.566666666666667		
non_V1	9	8	support(V4)	0.666666666666667	similarite	-0.6931032800836722		
			confiance(V1,V4)	0.5294117647058824	correlation	-0.33295604590814587		
V1	V5	non_V5	khi2	3.2850678733031673	support(V1)	0.566666666666667		
non_V1	16	1	support(V5)	0.833333333333334	similarite	0.4878081870470256		
			confiance(V1,V5)	0.9411764705882353	correlation	0.330911260679028		
V1	V6	non_V6	khi2	4.692954104718811	support(V1)	0.566666666666667		
non_V1	5	12	support(V6)	0.466666666666667	similarite	-1.01148353028532		
			confiance(V1,V6)	0.29411764705882354	correlation	-0.39551460591314075		
V1	V7	non_V7	khi2	1.330316742081448	support(V1)	0.566666666666667		
non_V1	13	4	support(V7)	0.833333333333334	similarite	-0.309965209930332		
			confiance(V1,V7)	0.7647058823294117	correlation	-0.21057989315866538		
V1	V8	non_V8	khi2	19.02714932126697	support(V1)	0.566666666666667		
non_V1	16	1	support(V8)	0.6	similarite	1.816050441469338		
			confiance(V1,V8)	0.9411764705882353	correlation	0.79639080272528		
V1	V9	non_V9	khi2	0.889172746757886	support(V1)	0.566666666666667		
non_V1	12	5	support(V9)	0.633333333333333	similarite	0.3758718794518176		
			confiance(V1,V9)	0.705882329411765	correlation	0.1721600739088879		
V1	V10	non_V10	khi2	0.2714932126698324	support(V1)	0.566666666666667		
non_V1	12	5	support(V10)	0.666666666666667	similarite	0.19802905503333466		
			confiance(V1,V10)	0.705882329411765	correlation	0.0951302988308882		
V1	V11	non_V11	khi2	0.28708183971291866	support(V2)	0.633333333333333		
non_V1	8	5	support(V11)	0.785882329411765	similarite	0.633333333333333		
			confiance(V2,V11)	0.705882329411765	correlation	0.097823197690969		
V2	V3	non_V3	khi2	0.333333333333333	support(V3)	0.333333333333333		
non_V2	7	12	support(V3)	0.333333333333333	similarite	0.2649064714130089		
			confiance(V2,V3)	0.3684210526315789	correlation	0.097823197690969		
V2	V4	non_V4	khi2	1.794258372057416	support(V2)	0.633333333333333		
non_V2	11	8	support(V4)	0.666666666666667	similarite	-0.468292957908408		
			confiance(V2,V4)	0.5789473684210527	correlation	0.2145579040225922		
V2	V5	non_V5	khi2	3.473684210526316	support(V2)	0.633333333333333		
non_V2	14	5	support(V5)	0.833333333333334	similarite	-0.460739299120852		
			confiance(V2,V5)	0.7368421052631579	correlation	-0.3402785236899603		
V2	V6	non_V6	khi2	0.740772855092276	support(V2)	0.633333333333333		
non_V2	10	9	support(V6)	0.466666666666667	similarite	0.386007613436981		
			confiance(V2,V6)	0.5263157894736842	correlation	0.157181966623882		
V2	V7	non_V7	khi2	4.851067641148326	support(V2)	0.633333333333333		
non_V2	18	1	support(V7)	0.833333333333334	similarite	0.54510080779204		
			confiance(V2,V7)	0.9473684210526315	correlation	0.4021473461783486		
V2	V8	non_V8	khi2	1.172248308277512	support(V2)	0.633333333333333		
non_V2	10	9	support(V8)	0.6	similarite	-0.414642144336473		
			confiance(V2,V8)	0.5263157894736842	correlation	-0.19767387315371676		
V2	V9	non_V9	khi2	5.440122707813466	support(V2)	0.633333333333333		
non_V2	15	4	support(V9)	0.633333333333333	similarite	0.573392891896255		
			confiance(V2,V9)	0.7894736842105263	correlation	0.428537320571627		
V2	V10	non_V10	khi2	0.28708183971291866	support(V2)	0.633333333333333		
non_V2	12	7	support(V10)	0.666666666666667	similarite	-0.1873171623163863		
			confiance(V2,V10)	0.631578947368421	correlation	-0.09782319769089039		
V3	V4	non_V4	khi2	4.8	support(V3)	0.333333333333333		
non_V3	4	6	support(V4)	0.666666666666667	similarite	0.5773502091896255		
			confiance(V3,V4)	0.4	correlation	-0.4		
V3	V5	non_V5	khi2	3.0	support(V3)	0.333333333333333		
non_V3	15	5	support(V5)	0.833333333333333	similarite	0.573392891896255		
			confiance(V3,V5)	1.0	correlation	0.3162277660168379		
V3	V6	non_V6	khi2	0.069696428571428571	support(V3)	0.333333333333333		
non_V3	5	5	support(V6)	0.666666666666667	similarite	0.1543034996209175		
			confiance(V3,V6)	0.5	correlation	0.0472455912615341		
V3	V7	non_V7	khi2	3.0	support(V3)	0.333333333333333		
non_V3	10	0	support(V7)	0.833333333333334	similarite	0.5773502091896255		
			confiance(V3,V7)	1.0	correlation	0.3162277660168379		
V3	V8	non_V8	khi2	0.0	support(V3)	0.333333333333333		
non_V3	10	0	support(V8)	1.0	similarite	1.629931618554523		
			confiance(V3,V8)	1.0	correlation	0.573392891896257		
V3	V9	non_V9	khi2	4.5930014354066986	support(V3)	0.333333333333333		
non_V3	9	1	support(V9)	0.633333333333333	similarite	1.05962585652053		
			confiance(V3,V9)	0.9	correlation	0.39129279043561477		
V3	V10	non_V10	khi2	0.075	support(V3)	0.333333333333333		
non_V3	7	3	support(V10)	0.666666666666667	similarite	0.12909044487358044		
			confiance(V3,V10)	0.7	correlation	0.05		
V4	V5	non_V5	khi2	3.0	support(V4)	0.666666666666667		
non_V4	15	5	support(V5)	0.833333333333334	similarite	-0.408248290458633		
			confiance(V4,V5)	0.75	correlation	-0.3162277660168379		
V4	V6	non_V6	khi2	0.26785714285714285	support(V4)	0.666666666666667		
non_V4	10	10	support(V6)	0.466666666666667	similarite	0.2182178923599217		
			confiance(V4,V6)	0.5	correlation	0.09449111825230681		
V4	V7	non_V7	khi2	0.48	support(V4)	0.666666666666667		
non_V4	16	4	support(V7)	0.833333333333334	similarite	-0.162993161855455		
			confiance(V4,V7)	0.8	correlation	-0.12649110640673517		
V4	V8	non_V8	khi2	5.625	support(V4)	0.666666666666667		
non_V4	9	11	support(V8)	0.6	similarite	-0.8660254037844387		
			confiance(V4,V8)	0.45	correlation	-0.433017201892193		
V4	V9	non_V9	khi2	4.5930014354066986	support(V4)	0.666666666666667		
non_V4	10	10	support(V9)	0.633333333333333	similarite	-0.74926864926355		
			confiance(V4,V9)	0.5	correlation	-0.39129279043561477		
V4	V10	non_V10	khi2	1.2	support(V4)	0.666666666666667		
non_V4	12	8	support(V10)	0.666666666666667	similarite	-0.3651483716701109		
			confiance(V4,V10)	0.6	correlation	-0.2		
V5	V6	non_V6	khi2	2.6785714285714284	support(V5)	0.833333333333334		
non_V5	4	1	support(V6)	0.466666666666667	similarite	0.4875003647426644		
			confiance(V5,V6)	0.4	correlation	-0.2988071523335984		
V5	V7	non_V7	khi2	1.2	support(V5)	0.833333333333334		
non_V5	20	5	support(V7)	0.333333333333334	similarite	-0.1825741858350551		
			confiance(V5,V7)	0.5	correlation	-0.2		
V5	V8	non_V8	khi2	4.0	support(V5)	0.833333333333334		
non_V5	17	8	support(V8)	0.6	similarite	0.516397794943222		
			confiance(V5,V8)	0.68	correlation	0.3651483716701107		
V5	V9	non_V9	khi2	0.028708133971291867	support(V5)	0.833333333333334		
non_V5	16	9	support(V9)	0.633333333333333	similarite	0.014885908291694		
			confiance(V5,V9)	0.64	correlation	0.60904141244482792		
V5	V10	non_V10	khi2	0.48	support(V5)	0.833333333333334		
non_V5	16	9	support(V10)	0.666666666666667	similarite	-0.162993161855455		
			confiance(V5,V10)	0.64	correlation	-0.12649110640673517		
V6	V7	non_V7	khi2	0.4285714285714285	support(V6)	0.466666666666667		
non_V6	11	3	support(V7)	0.833333333333334	similarite	-0.1951801458970646		
			confiance(V6,V7)	0.7857142857142857	correlation	-0.1195228609334936		
V6	V8	non_V8	khi2	3.2142857142857144	support(V6)	0.466666666666667		
non_V6	6	8	support(V8)	0.6	similarite	-0.8280786712108251		
			confiance(V6,V8)	0.4285714285714285	correlation	-0.3272308335398854		
V6	V9	non_V9	khi2	4.739405331510595	support(V6)	0.466666666666667		
non_V6	6	8	support(V9)	0.633333333333333	similarite	-0.627133751052969		
			confiance(V6,V9)	0.4285714285714285	correlation	-0.3974672033225129		
V6	V10	non_V10	khi2	4.285714285714286	support(V6)	0.466666666666667		
non_V6	12	2	support(V10)	0.666666666666667	similarite	0.8728156909439692		
			confiance(V6,V10)	0.8571428571428571	correlation	-0.3162277660168379		
V7	V8	non_V8	khi2	0.0	support(V7)	0.833333333333334		
non_V7	15	10	support(V8)	0.6	similarite	0.0		
			confiance(V7,V8)	0.6	correlation	0.0		
V7	V9	non_V9	khi2	1.406689564593014	support(V7)	0.833333333333334		
non_V7	17	8	support(V9)	0.633333333333333	similarite	0.2931973580418673		
			confiance(V7					

B.12 Résultats 2 : §4.4.2

N	MGR(X > Y)	MGR-critique	Chi2-Observable	conf(X > Y)	correlation	supp(X > Y)	supp(X < Y)	supp(0) [X > Y]	supp(X)	supp(Y)	X favorable Y	MGR(X > non_Y)
1	0.0	0.344785448282822	4.47441966702028	0.47058823529411764	-0.381967287608566	0.358888888888887	0.0	0.0	0.566666666666667	0.333333333333333	false	0.108591331203349
2	0.0	0.1850408272025268	3.32579185520362	0.47058823529411764	0.33295604500814587	0.236666666666666	0.0	0.2058235294117646	0.566666666666667	0.333333333333333	true	0.0
3	0.0	0.3710081654050536	3.32579185520362	0.5294117647058824	-0.33295604500814587	0.377777777777777	0.0	0.2058235294117646	0.566666666666667	0.333333333333333	true	0.0
4	0.6476888235294118	0.5866154106562821	3.2580678733031673	0.9411764705882534	0.3309112600779028	0.333333333333333	0.0	0.6470588235294112	0.566666666666667	0.333333333333333	true	0.0
5	0.0	0.24580883093247142	4.692954104718811	0.29411764705882534	-0.3955146059134075	0.264444444444444	0.0	0.0	0.566666666666667	0.466666666666667	false	0.3487394957983193
6	0.0	0.5866154106562821	3.330316742083448	0.7647058825294118	-0.210579489318066538	0.333333333333333	0.0	0.0	0.566666666666667	0.6	true	0.0
7	0.8529411764705882	0.321302490282527	3.889172746778786	0.9411764705882534	0.7963080275258	0.333333333333333	0.0	0.529411764705882	0.566666666666667	0.6	true	0.0
8	0.19780969626684193	0.344785448282822	0.889172746778786	0.7058823529411765	0.172100075890887	0.399999999999999	0.0	0.19780969626684185	0.566666666666667	0.633333333333333	true	0.0
9	0.11764705882352944	0.161082709534015	0.28708133971291886	0.3884210320315789	0.0978231970809369	0.399999999999999	0.0	0.1764705882352927	0.566666666666667	0.666666666666667	true	0.0
10	0.0526315789179836	0.3228165321098322	1.7942838752057416	0.378937085210527	-0.2445759940225922	0.2222222222222217	0.0	0.0263157891798356	0.633333333333333	0.666666666666667	false	0.0
11	0.0	0.3104177853340405	3.473683210293316	0.326315789179836	0.340278326830303	0.277777777777778	0.0	0.0	0.633333333333333	0.833333333333333	true	0.0
12	0.0	0.228165321098322	1.7942838752057416	0.378937085210527	-0.2445759940225922	0.2222222222222217	0.0	0.0	0.633333333333333	0.833333333333333	true	0.0
13	0.11184210526315798	0.2135250789435118	0.740772853092276	0.326315789179836	0.340278326830303	0.277777777777778	0.0	0.11184210526315798	0.633333333333333	0.833333333333333	true	0.0
14	0.68421052631579836	0.3104177853340405	4.53107464118326	0.947308120293316	0.4021176471833486	0.0	0.0	0.68421052631579836	0.633333333333333	0.833333333333333	true	0.0
15	0.0	0.296673396890004	1.729638328774312	0.326315789179836	0.340278326830303	0.277777777777778	0.0	0.0	0.633333333333333	0.833333333333333	true	0.0
16	0.0	0.296673396890004	1.729638328774312	0.326315789179836	0.340278326830303	0.277777777777778	0.0	0.0	0.633333333333333	0.833333333333333	true	0.0
17	0.0	0.296673396890004	1.729638328774312	0.326315789179836	0.340278326830303	0.277777777777778	0.0	0.0	0.633333333333333	0.833333333333333	true	0.0
18	0.0	0.296673396890004	1.729638328774312	0.326315789179836	0.340278326830303	0.277777777777778	0.0	0.0	0.633333333333333	0.833333333333333	true	0.0
19	0.0	0.9488328905138	3.0	0.4	-0.4	0.0	0.0	0.0	0.333333333333333	0.666666666666667	false	0.4
20	0.0025	0.9686269686886	0.6696428571428571	1.0	0.0	0.222222222222222	0.0	0.0	0.333333333333333	0.666666666666667	true	0.1428571428571429
21	0.0	0.9488328905138	3.0	0.5	0.0	0.222222222222222	0.0	0.0	0.333333333333333	0.666666666666667	true	0.0
22	0.0	0.519615242706632	10.0	1.0	0.0	0.333333333333333	0.0	0.0	0.333333333333333	0.666666666666667	true	0.0
23	0.7272727272727273	0.53759222863707511	4.933014354066986	0.9	0.0	0.333333333333333	0.0	0.7272727272727273	0.333333333333333	0.6	true	0.0
24	0.10000000000000009	0.6	0.075	0.7	0.05	0.3	0.0	0.0	0.666666666666667	0.666666666666667	true	0.0
25	0.0	0.474941649025269	3.0	0.75	-0.3162277660168379	0.233333333333333	0.0	0.0	0.666666666666667	0.666666666666667	true	0.0
26	0.0025	0.198431348239843	0.26785714285714285	0.8	0.04491182629081	0.333333333333333	0.0	0.0	0.666666666666667	0.666666666666667	false	0.0
27	0.0	0.474941649025269	4.8	0.8	-0.12649110640673517	0.333333333333333	0.0	0.0	0.666666666666667	0.666666666666667	true	0.0
28	0.0	0.259807621353316	5.025	0.45	-0.4330127018922193	0.399999999999999	0.0	0.0	0.666666666666667	0.666666666666667	false	0.0
29	0.0	0.27879611318537556	4.5933014354066986	0.5	-0.39129270043561477	0.2222222222222217	0.0	0.0	0.666666666666667	0.666666666666667	false	0.0
30	0.0	0.3	1.2	0.6	-0.2	0.444444444444444	0.0	0.0	0.666666666666667	0.666666666666667	false	0.0
31	0.0	0.1254900093901132	2.6785714285714284	0.4	-0.298807152335984	0.388888888888889	0.0	0.0	0.833333333333333	0.833333333333333	true	0.0
32	0.0	0.3	1.2	0.8	-0.2	0.444444444444444	0.0	0.0	0.833333333333333	0.833333333333333	true	0.0
33	0.19999999999999996	0.16416167625154984	0.028708133971291887	0.68	0.365148371071017	0.666666666666667	0.0	0.19999999999999993	0.833333333333333	0.6	true	0.0
34	0.01818181818181818	0.1762961440935776	0.028708133971291887	0.64	0.0309344112344487297	0.333333333333333	0.0	0.01818181818181815	0.833333333333333	0.6	true	0.0
35	0.0	0.18973665961010275	0.48	0.64	-0.12649110640673517	0.555555555555556	0.0	0.0	0.833333333333333	0.6	true	0.0
36	0.0	0.717371656006361	0.4285714285714285	0.64	-0.11932286093343086	0.555555555555556	0.0	0.0	0.833333333333333	0.666666666666667	false	0.0
37	0.0	0.3927922024247863	3.2142857142857144	0.4285714285714285	-0.3272685335308854	0.279999999999999	0.0	0.0	0.466666666666667	0.6	false	0.0
38	0.0	0.4215001089852533	4.739403331510595	0.4285714285714285	-0.397467203225129	0.295555555555556	0.0	0.0	0.466666666666667	0.6	false	0.0
39	0.5714285714285714	0.4535573676110727	4.285714285714286	0.8571428571428571	0.37796447039922725	0.399999999999999	0.0	0.3714285714285711	0.466666666666667	0.633333333333333	false	0.2380652890321814
40	0.0	0.1643167625154984	0.0	0.0	0.0	0.5	0.0	0.0	0.833333333333333	0.6	true	0.0
41	0.1762961440935776	1.406695645033014	1.006995645033014	0.68	0.2165627807114111	0.666666666666667	0.0	0.1762961440935776	0.833333333333333	0.6	true	0.0
42	0.0	0.18973665961010275	3.0	0.6	0.3162277660168379	0.555555555555556	0.0	0.0	0.833333333333333	0.666666666666667	false	0.0
43	0.24242424242424243	0.3219260219931959	1.531100478468995	0.7222222222222222	0.22519297089062	0.333333333333333	0.0	0.24242424242424246	0.6	0.6	true	0.0
44	0.16666666666666674	0.346101615137755	0.625	0.7222222222222222	0.1433576729740643	0.333333333333333	0.0	0.16666666666666663	0.6	0.6	true	0.0
45	0.0	0.3228165321098322	1.7942838752057416	0.578937085210527	-0.2445759940225922	0.2222222222222217	0.0	0.0	0.666666666666667	0.666666666666667	false	0.0

Tableau B.2 – Tableau représentant les caractéristiques de toutes les règles positives et négatives du contexte 4.2 donné

X	Y	supp(X)	supp(Y)	suppMGK(X->Y)-Valides	supp(X->Y)	supp(n)(X->Y)	conf(X->Y)
V1	V3	0.567	0.333	0,267	0.266717934	0.205999999999999996	0.47040200000000004
V1	V5	0.567	0.833	0,534	0.53357478299999999	0.64700000000000001	0.941049
V1	V8	0.567	0.6	0,534	0.53366039999999999	0.85299999999999999	0.94119999999999999
V2	V7	0.633	0.833	0,6	0.59959532399999999	0.68399999999999999	0.94722799999999998
V2	V9	0.633	0.633	0,5	0.499653486	0.42599999999999999	0.789342
V3	V1	0.333	0.567	0,266	0.26638468200000004	0.53800000000000001	0.799954
V3	V5	0.333	0.833	0,333	0.333	1.00000000000000000	1.0
V3	V7	0.333	0.833	0,333	0.333	1.00000000000000000	1.0
V3	V8	0.333	0.6	0,333	0.333	1.0	1.0
V3	V9	0.333	0.633	0,3	0.299636397	0.727	0.899809
V5	V1	0.833	0.567	0,533	0.533267441	0.16900000000000002	0.640177
V5	V3	0.833	0.333	0,333	0.3329501	0.10000000000000006	0.39970000000000006
V5	V8	0.833	0.6	0,566	0.56643999999999999	0.19999999999999993	0.67999999999999999
V6	V10	0.467	0.667	0,4	0.400285781	0.57100000000000001	0.857143
V7	V2	0.833	0.633	0,599	0.599436796	0.236	0.719612
V7	V3	0.833	0.333	0,333	0.3329501	0.10000000000000006	0.39970000000000006
V8	V1	0.6	0.567	0,533	0.53349119999999999	0.74399999999999999	0.88915199999999999
V8	V3	0.6	0.333	0,333	0.3330666	0.33299999999999996	0.555111
V8	V5	0.6	0.833	0,567	0.5666334	0.66699999999999997	0.94438899999999999
V9	V2	0.633	0.633	0,5	0.499653486	0.42599999999999999	0.789342
V9	V3	0.633	0.333	0,3	0.29987552100000003	0.21100000000000008	0.47373700000000001
V10	V6	0.667	0.467	0,4	0.40036675	0.24999999999999997	0.60025

Tableau B.3 – Tableau représentant les caractéristiques des règles M_{GK} -valides

B.13 Résultats 3 : §5.2.2

 PREORDRE GLOBALE

Couple	Cohesion
(V3,V5)	1.0
(V3,V7)	1.0
(V3,V8)	1.0
(V1,V8)	0.798
(V3,V9)	0.534
(V2,V7)	0.436
(V8,V5)	0.397
(V1,V5)	0.35
(V6,V10)	0.17
(V3,V1)	0.091

Tableau B.4 – Tableau des Paires Orientées triées

	Regle	cohésion inter-classe	implication inter-classe	valeur significative
R(1)=	(V3,V5)	1.0	1.0	0.019247385193381768
R(2)=	((V3,V5),V8)	0.7349596596552507	1.0	0.01954400304949093
R(3)=	(V1,((V3,V5),V8))	0.46485460666233813	0.5319722690003863	0.02016647329586997
R(4)=	(V2,V7)	0.436	0.6842105263157895	-0.013169496030767124
R(5)=	(V6,V10)	0.17	0.5714285714285714	-0.013868940733347683

Tableau B.5 – Tableau montrant les classes constituées, $coh_{supp(n)M_{GK}}(\underline{C})$ et $\psi(\underline{C}, \underline{C}')$

B.14 Résultats 4 : §5.3

niveau 1 : (V3,V5) ; groupe optimal 25 contribution du groupe=0				
individu	phi	distance empirique	distance idéale théorique	contribution
i1		0	0	0
i2		0	0	0
i3		0	0	0
i4		0	0	0
i5	(1)	0	0	0
i6	(1)	0	0	0
i7	(1)	0	0	0
i8	(1)	0	0	0
i9	(1)	0	0	0
i10	(1)	0	0	0
i11	(1)	0	0	0
i12	(1)	0	0	0
i13	(1)	0	0	0
i14	(1)	0	0	0
i15	(1)	0	0	0
i16	(1)	0	0	0
i17	(1)	0	0	0
i18	(1)	0	0	0
i19	(1)	0	0	0
i20	(1)	0	0	0
i21	(1)	0	0	0
i22	(1)	0	0	0
i23	(1)	0	0	0
i24	(1)	0	0	0
i25	(1)	0	0	0
i26	(1)	0	0	0
i27	(1)	0	0	0
i28	(1)	0	0	0
i29	(1)	0	0	0
i30		0	0	0

niveau 2 : (V3,V5) ,(V8,V5) ,(V3,V8); groupe optimal 26 contribution du groupe 1,038				
individu	phi	distance empirique	distance idéale théorique	contribution
i1		0,289	0,577	0,5
i2		0,289	0,577	0,5
i3		0,289	0,577	0,5
i4		0,289	0,577	0,5
i5	(1,1,1)	0,577	0,577	1
i6	(1,1,1)	0,577	0,577	1
i7	(1,1,1)	0,577	0,577	1
i8	(1,1,1)	0,577	0,577	1
i9	(1,1,1)	0,577	0,577	1
i10	(1,1,1)	0,577	0,577	1
i11	(1,1)	0,577	0,577	1
i12	(1,1)	0,577	0,577	1
i13	(1,1)	0,577	0,577	1
i14	(1,1)	0,577	0,577	1
i15	(1,1)	0,577	0,577	1
i16	(1,1)	0,577	0,577	1
i17	(1,1)	0,577	0,577	1
i18	(1,1)	0,577	0,577	1
i19	(1,1,1)	0,577	0,577	1
i20	(1,1,1)	0,577	0,577	1
i21	(1,1,1)	0,577	0,577	1
i22	(1,1,1)	0,577	0,577	1
i23	(1,1,1)	0,577	0,577	1
i24	(1,1,1)	0,577	0,577	1
i25	(1,1,1)	0,577	0,577	1
i26	(1,1,1)	0,577	0,577	1
i27	(1,1,1)	0,577	0,577	1
i28	(1,1,1)	0,577	0,577	1
i29	(1,1,1)	0,577	0,577	1
i30	(1)	1,155	0,577	2

Niveau 3 : (V3,V5) ,(V8,V5) ,(V3,V8) ,(V3,V1) ,(V1,V5) ,(V1,V8); groupe optimal 26 contribution du groupe 1,044				
individu	phi	distance empirique	distance idéale théorique	contribution
i1		1,513	2,234	0,677
i2		1,513	2,234	0,677
i3		1,513	2,234	0,677
i4		1,513	2,234	0,677
i5	(1,1,1,1,1)	2,348	2,234	1,051
i6	(1,1,1,1,1)	2,348	2,234	1,051
i7	(1,1,1,1,1)	2,234	2,234	1
i8	(1,1,1,1,1)	2,234	2,234	1
i9	(1,1,1,1,1)	2,234	2,234	1
i10	(1,1,1,1,1)	2,234	2,234	1
i11	(1,1,1)	2,148	2,234	0,962
i12	(1,1,1)	2,148	2,234	0,962
i13	(1,1,1)	2,148	2,234	0,962
i14	(1,1,1)	2,148	2,234	0,962
i15	(1,1,1)	2,148	2,234	0,962
i16	(1,1,1)	2,148	2,234	0,962
i17	(1,1,1)	2,148	2,234	0,962
i18	(1,1,1,1,1)	4,075	2,234	1,824
i19	(1,1,1,1,1)	2,234	2,234	1
i20	(1,1,1,1,1)	2,234	2,234	1
i21	(1,1,1,1,1)	2,234	2,234	1
i22	(1,1,1,1,1)	2,234	2,234	1
i23	(1,1,1,1,1)	2,234	2,234	1
i24	(1,1,1,1,1)	2,234	2,234	1
i25	(1,1,1,1,1)	2,234	2,234	1
i26	(1,1,1,1,1)	2,234	2,234	1
i27	(1,1,1,1,1)	2,234	2,234	1
i28	(1,1,1,1,1)	2,234	2,234	1
i29	(1,1,1,1,1)	2,234	2,234	1
i30	(1,1,1)	3,307	2,234	1,48

Niveau 4 : (V2,V7) ; groupe optimal 25 contribution du groupe 1				
individu	phi	distance empirique	distance idéale théorique	contribution
i1	(1)	0,562	0,562	1
i2	(1)	0,562	0,562	1
i3	(1)	0,562	0,562	1
i4	(1)	0,562	0,562	1
i5	(1)	0,562	0,562	1
i6	(1)	0,562	0,562	1
i7	(1)	0,562	0,562	1
i8	(1)	0,562	0,562	1
i9	(1)	0,562	0,562	1
i10	(1)	0,562	0,562	1
i11	(1)	0,562	0,562	1
i12	(1)	0,562	0,562	1
i13	(1)	0,562	0,562	1
i14	(1)	0,562	0,562	1
i15	(1)	0,562	0,562	1
i16	(1)	0,562	0,562	1
i17		0,328	0,562	0,583
i18		0,328	0,562	0,583
i19		0,328	0,562	0,583
i20	(1)	0,562	0,562	1
i21	(1)	0,562	0,562	1
i22	(1)	0,562	0,562	1
i23		0,328	0,562	0,583
i24		1,218	0,562	2,167
i25	(1)	0,562	0,562	1
i26	(1)	0,562	0,562	1
i27	(1)	0,562	0,562	1
i28	(1)	0,562	0,562	1
i29	(1)	0,562	0,562	1
i30	(1)	0,562	0,562	1

Niveau 5 : (V6,V10) ; groupe optimal 20 contribution du groupe 1				
individu	phi	distance empirique	distance idéale théorique	contribution
i1	(1)	0,655	0,655	1
i2	(1)	0,655	0,655	1
i3	(1)	0,655	0,655	1
i4	(1)	0,655	0,655	1
i5	(1)	0,655	0,655	1
i6	(1)	0,655	0,655	1
i7	(1)	0,655	0,655	1
i8	(1)	0,655	0,655	1
i9	(1)	0,655	0,655	1
i10	(1)	0,655	0,655	1
i11	(1)	0,655	0,655	1
i12		0,109	0,655	0,167
i13		0,109	0,655	0,167
i14		0,109	0,655	0,167
i15		0,873	0,655	1,333
i16		0,873	0,655	1,333
i17	(1)	0,655	0,655	1
i18	(1)	0,655	0,655	1
i19	(1)	0,655	0,655	1
i20	(1)	0,655	0,655	1
i21	(1)	0,655	0,655	1
i22	(1)	0,655	0,655	1
i23	(1)	0,655	0,655	1
i24	(1)	0,655	0,655	1
i25	(1)	0,655	0,655	1
i26		0,109	0,655	0,167
i27		0,109	0,655	0,167
i28		0,109	0,655	0,167
i29		0,109	0,655	0,167
i30		0,109	0,655	0,167

Tableau B.6 – Contributions des individus

Annexe C

Résultats de la stratégie de mise en œuvre pour l'intégration des TIC par la méthode AFM

C.1 Questionnaire

<p>QUESTIONNAIRE INDIVIDUEL</p> <p>NOM DU RÉPONDANT</p> <p>001 ENSEIGNANT/ETUDIANT/ELEVE(ENSEIGNANT=1,ETUDIANT =2, ÉLÈVE=3)</p> <p>002 SEXE (HOMME=1, FEMME=2)</p> <p>003M MOIS DE NAISSANCE</p> <p>003A ANNÉE DE NAISSANCE</p> <p>004 ÂGE</p>
<p>PROFIL DE L'ENSEIGNANT</p> <p>005 Matière enseignée ?</p> <p>006 Nombre d'années dans l'enseignement ?</p> <p>007 Nombre d'années dans L'Etablissement ?</p>
<p>PROFIL DE L'ÉLÈVE/ETUDIANT</p> <p>008 Classe fréquentée ? Année</p> <p>009 Série ? Filière</p> <p>010 (GÉNÉRAL=1, TERTIAIRE=2, INDUSTRIEL=3, TECHNIQUE=4, TECHNOLOGIQUE=5, PROFESSIONNEL=6)</p>
<p>SECTION 1 : CONNAISSANCES ET UTILISATION DE L'ORDINATEUR ET/OU DE L'INTERNET COMME COMPOSANTES TIC</p> <p>A. CONNAISSANCES ET UTILISATION DES TIC</p> <p>101 Savez-vous que le Lycée/Université dispose d'ordinateurs et d'une connexion Internet ? (OUI=1, NON=2, NE SAIT PAS=3)</p> <p>102 Avez-vous accès à un ordinateur ? (OUI= 1, NON=2 (Aller à 111))</p> <p>103 Où avez-vous accès à un ordinateur ? (AU LYCEE/UNIVERSITE = 1, DANS UN SERVICE PUBLIC/SERVICE PRIVÉ = 4, AUTRE _____ = 5 (préciser))</p> <p>104 Travaillez-vous avec l'ordinateur très fréquemment, souvent, quelquefois, rarement ou jamais ? (TRÈS FRÉQUEMMENT = 1, SOUVENT = 2, QUELQUEFOIS = 3 RAREMENT=4, JAMAIS= 5 (Aller à 106))</p> <p>105 Quelle utilisation faites-vous (ou quel genre de travail faites-vous) principalement de (sur) l'ordinateur ? (SAISIE COURS/LEÇON/TEXTE = 1, DISTRACTION (JEUX) = 2, PASSE-TEMPS = 3, AUTRE _____ 4 (préciser))</p> <p>106 Avez-vous accès à l'Internet ? (OUI=1, NON=2 (aller à 111))</p> <p>107 Où avez-vous accès à l'Internet ? (AU LYCEE/UNIVERSITE = 1, DANS LES CYBERCAFÉS = 2, À LA MAISON = 3, DANS UN SERVICE PUBLIC/SERVICE PRIVÉ = 4, AUTRE _____ = 5 (préciser))</p> <p>108 Utilisez-vous l'Internet tous les jours, quelquefois par semaine, rarement ou jamais ? (TOUS LES JOURS= 1, QUELQUEFOIS=2, RAREMENT=3,</p> <p>109 Quelle utilisation faites-vous (ou quel genre de travail faites-vous) principalement de (sur) l'Internet ? (RECHERCHES DANS LE CADRE DE MES ENSEIGNEMENTS = 1 RECHERCHES DANS LE CADRE DE MA FORMATION= 2 SUIVRE L'ACTUALITÉ DANS LE MONDE = 3 CORRESPONDRE AVEC LE COURRIER ÉLECTRONIQUE (E-MAIL) = 4 RECHERCHE D'EMPLOI= 5 PASSE-TEMPS=6 AUTRE _____=7 (préciser))</p> <p>110 ENSEIGNANT : Y trouvez-vous des ressources intéressantes pour votre discipline ? ÉLÈVE/ETUDIANT : Y trouvez-vous des informations intéressantes pour votre apprentissage ? (OUI= 1, NON=2)</p> <p>111 Selon vous, l'ordinateur et l'Internet peuvent-ils servir à apprendre efficacement, un peu... ? EFFICACEMENT= 1 UN PEU= 2 PAS DU TOUT=3 NE SAIT PAS=4</p> <p>112 VÉRIFIER 103 A ACCÈS À L'INTERNET = 1, A ACCÈS À L'INTERNET AILLEURS = 2 AU Lycée/Université (CODE '1' À 103) OU QUESTION NON POSÉE (aller à 115)</p> <p>113 Rencontrez-vous des problèmes dans l'utilisation d'Internet au Lycée/Université? (OUI=1, NON= 2 (aller à 116))</p> <p>114 Quel type de problème principalement avez-vous souvent rencontré ? DIFFICULTÉ DE CONNEXION AU RÉSEAU = 1 SALLE DES MACHINES PAS TOUJOURS DISPONIBLE = 4</p> <p>115 VÉRIFIER 106 ET 107 N'A PAS ACCÈS À L'INTERNET (CODE '2' À 106) OU A ACCÈS À L'INTERNET AILLEURS (CODES DIFFÉRENTS DE '1' À 107) Pourquoi n'avez-vous pas accès à l'Internet ? Pourquoi n'utilisez-vous pas l'Internet au Lycée/Université ? PAS INFORMÉ QU'ON POUVAIT S'ABONNER = 1, FRAIS D'ABONNEMENT TROP CHER = 2 SALLE DES MACHINES PAS TOUJOURS DISPONIBLE = 3, IGNORANCE DE SON EXISTENCE AU LYCEE/UNIVERSITE = 4 IGNORANCE DE SON FONCTIONNEMENT = 5, PAS DE TEMPS (EMPLOI DE TEMPS CHARGÉ = 6 HORAIRES D'ACCÈS NON-CONFORMES À MON EMPLOI DE TEMPS=7, NE M'INTÉRESSE PAS= 08, ÉLOIGNÉ=9, AUTRE=10 _____(préciser))</p> <p>116 Selon vous, qu'est-ce qu'il est possible de faire principalement au Lycée/Université en vue d'améliorer l'accès aux ordinateurs et au service d'Internet ? INFORMER DE LA DISPONIBILITÉ D'ORDINATEURS ET D'INTERNET AU LYCEE/UNIVERSITE = 1 OFFRIR DES SERVICES GRATUITS = 2 REVOIR LES FRAIS D'ABONNEMENT À LA BAISSE = 3 DONNER DES FORMATIONS SUR L'UTILISATION DES TIC=4 PERMETTRE L'ACCÈS À TOUT MOMENT = 5 INTÉGRER DES COURS D'INFORMATIQUE AU PROGRAMME DES ÉLÈVES/ETUDIANTS DE TOUTES LES CLASSES/FILIERES = 6 AUTRE _____ = 7 (préciser)</p>

<p>B. FORMATION EN INFORMATIQUE</p> <p>117 Avez-vous eu une formation en informatiques? (OUI=1, NON=2 (Aller à 121))</p> <p>118 Dans quels domaines Bureautiques=1, Programmation =2, Internet =3, Maintenance= 4, Autres _____ =5 (à préciser)</p> <p>119 Avez-vous tiré des avantages de la formation ? Enseignement = 1, Apprentissage = 2, Autres _____ = 3 (à préciser)</p> <p>120 Etes-vous intéressés par une formation en Informatiques? (OUI=1, NON=2 (Aller à 121))</p> <p>Dans quels domaines Bureautiques=1, Programmation =2, Internet =3, Maintenance = 4, Autres _____ =5 (à préciser)</p>	
<p>SECTION 2 : CONNAISSANCES ET UTILISATION DU TICE POUR L'ENSEIGNANT/ETUDIANT-ELEVE</p> <p>121 Savez-vous l'existence de vidéo projecteur au niveau de l'Établissement (OUI=1, NON=2 (Aller à 123 ou 124))</p> <p>122 Utilisez-vous un vidéo projecteur ? (OUI=1, NON=2 (Aller à 123 ou 124))</p> <p>123 Combien d'heures par semaine vous vous en servez ? _____</p> <p>124 Selon vous, le vidéo projecteur peut-il servir à apprendre efficacement, un peu ou pas du tout ? EFFICACEMENT= 1 UN PEU= 2 PAS DU TOUT=3 NE SAIT PAS=4</p> <p>125 Selon vous, le retroprojecteur peut-il servir à apprendre efficacement, un peu ou pas du tout ? EFFICACEMENT= 1 UN PEU= 2 PAS DU TOUT=3 NE SAIT PAS=4</p> <p>126 Selon vous, le Tableau numérique peut-il servir à apprendre efficacement, un peu ou pas du tout ? EFFICACEMENT= 1 UN PEU= 2 PAS DU TOUT=3 NE SAIT PAS=4</p> <p>127 Selon vous, le Tableau Noir peut-il servir à apprendre efficacement, un peu ou pas du tout ? EFFICACEMENT= 1 UN PEU= 2 PAS DU TOUT=3 NE SAIT PAS=4</p> <p>128 Quels supports utilisez-vous pendant votre étude personnelle/Préparation de votre cours ? Livres = 1, Internet = 2, Documents Manuscrits = 3, Autres = 4 (à préciser)</p> <p>129 Avez-vous une version numérique de votre cours ? Oui = 1, Non = 2 (Aller à 131)</p> <p>130 Pourcentage : 25% = 1, 50% = 2, 75% = 3, 100% = 4</p> <p>131 Problème de disponibilité = 1, Problème de Moyen = 2, Autres = 3 (à préciser)</p> <p>132 Avez-vous déjà entendu parler du sigle TIC ? Oui = 1, Non = 2</p> <p>133 Avez-vous déjà entendu parler du sigle TICE ? Oui = 1, Non = 2</p> <p>134 Avez-vous déjà entendu parler du projet EDUCMAD ? Oui = 1, Non = 2 (Aller à 136)</p> <p>135 Etes-vous satisfait de ce projet ? EFFICACEMENT= 1 UN PEU= 2 PAS DU TOUT=3</p> <p>136 Pouvez-vous citer deux ou trois problèmes relatifs à ce projet</p> <p>137 Avez-vous une suggestion pour améliorer l'utilisation de TIC dans l'éducation ?</p>	
<p>SECTION 3 : OPINION SUR LA PLACE ET LE ROLE DE L'INTERNET DANS L'ENSEIGNEMENT EN GENERAL ET CELUI DE L'APPRENTISSAGE DE L'INFORMATIQUE EN PARTICULIER</p> <p>201 Selon vous, l'ordinateur et l'Internet peuvent-ils servir à apprendre efficacement, ... l'informatique ? EFFICACEMENT= 1 UN PEU= 2 PAS DU TOUT=3 NE SAIT PAS=4</p> <p>202 Quel rôle joue principalement l'ordinateur et/ou l'Internet dans l'apprentissage de l'Informatique ? FAVORISE LES APPRENTISSAGES, LES RECHERCHES=1 ASSISTE LES ÉLÈVES FAIBLES OU PRÉSENTANT DES DIFFICULTÉS D'APPRENTISSAGE=2 OFFRE DES COURS ET EXERCICES= 3 OFFRE DES RÉPONSES À CERTAINES QUESTIONS DU COURS (ÉCLAIRCIS DES NOTIONS)/AIDE À CORRIGER LES FAUTES=4 APPORTE PLUS DE MOTIVATION= 5 AMÉLIORE LES CONTENUS DU COURS= 6 AMÉLIORE LES RÉSULTATS SCOLAIRES=7 PRÉPARE AU MARCHÉ DE L'EMPLOI= AUTRE _____9 (préciser)</p> <p>203 Trouvez-vous des ressources intéressantes sur l'Internet pour enrichir vos cours d'Informatique ? (OUI=1, NON=2, PAS DU TOUT=3)</p> <p>204 Selon vous, quelle est la raison principale qui vous fait dire que l'ordinateur et/ou l'Internet ne permettent pas d'apprendre l'Informatique ? REND PARESSEUX=1 OFFRE DES RESSOURCES PAS TOUJOURS FIABLES=3 CONTRIBUE À LA DÉPRÉVATION DES MŒURS=4 OFFRE DES CONNAISSANCES NON ORGANISÉES=5 DEMANDE BEAUCOUP DE TEMPS DE RECHERCHE= 6 MOYEN TROP COUTEUX=7</p> <p>205 Quel rôle joue principalement l'ordinateur et/ou l'Internet dans l'enseignement ? FAVORISE LES APPRENTISSAGES, LES RECHERCHES=1 ASSISTE LES ÉLÈVES FAIBLES OU PRÉSENTANT DES DIFFICULTÉS D'APPRENTISSAGE=2 OFFRE DES COURS ET EXERCICES= 3 OFFRE DES RÉPONSES À CERTAINES QUESTIONS DU COURS (ÉCLAIRCIS DES NOTIONS)/AIDE À CORRIGER LES FAUTES=4 APPORTE PLUS DE MOTIVATION= 5 AMÉLIORE LES CONTENUS DU COURS= 6 AMÉLIORE LES RÉSULTATS SCOLAIRES=7 PRÉPARE AU MARCHÉ DE L'EMPLOI= AUTRE _____9 (préciser)</p>	

Figure C.1 – Modèle de questionnaire pour l'étude sur l'intégration du TIC au Lycée

C.3 Les tables de rapport Classe/Modalité et Valeur-test

	Cla/Mod	Mod/Cla	Global	p.value	v.test
V001=Eleve	89.333333	95.714286	44.117647	0	12.131164
V009=V009_A	100	75.714286	31.176471	0	11.208974
V010=General	85.333333	91.428571	44.117647	0	10.88721
V008=V008_TERMINALE	100	54.285714	22.352941	0	8.797188
V002= Femme	84	60	40	0	7.638946
V134=V134_Oui	84	60	29.411765	0	7.290471
V008=V008_IERE	100	28.571429	11.764706	0	5.811241
V120=V120_Oui	57.009346	87.142857	62.941176	0	5.507777
V124=V124_Efficacement	68.852459	60	35.882353	0	5.350495
V113=NA	75	47.142857	25.882353	0	5.134046
V126=V126_Efficacement	69.811321	52.857143	31.176471	0.000001	4.947737
V128=V128_Livres	58.888889	75.714286	52.941176	0.000001	4.899646
V125=V125_Efficacement	76.470588	37.142857	20	0.000007	4.488679
V135=V135_Un_pou	64.285714	51.428571	32.941176	0.000037	4.123256
V116=Service_gratuit	60	60	41.176471	0.000057	4.025677
V204=V204_Asservit_homme	78.26087	25.714286	13.529412	0.000249	3.663288
V202=V202_Apport_motiv	83.333333	21.428571	10.588235	0.0003	3.615126
V115=V115_Pas_le_temps	76	27.142857	14.705882	0.000308	3.608498
V120A=V120A_Internat	72.413793	30	17.058824	0.000407	3.535816
V131=V131_Autre	86.666667	18.571429	8.823529	0.00044	3.514792
V133=NA	70.967742	31.428571	18.235294	0.000443	3.513291
V104=V104_Rarement	75	25.714286	14.117647	0.000666	3.403354
V132=NA	73.076923	27.142857	15.294118	0.000766	3.364772
V135=V135_Pas_du_tout	65.789474	35.714286	22.352941	0.00098	3.296252
V202=V202_Assist_Elev_faible	77.777778	20	10.588235	0.002053	3.082385
V205=V205_Favorise_Appren_Rech	51.020408	71.428571	57.647059	0.003626	2.990096
V109=V109_mail	73.684211	20	11.76471	0.005154	2.797246
V101=V101_Oui	48.305085	81.428571	69.411765	0.006579	2.717414
V201=V201_Efficacement	47.540984	82.857143	71.764706	0.010638	2.555035
V204=V204_Offre_Ressources_pas_fiable	68.181818	21.428571	12.941176	0.012036	2.511077
V123=NA	47.154472	82.857143	73.252941	0.015373	2.42346
V127=V127_Efficacement	49	70	58.823529	0.019656	2.332851
V009=V009_D	100	7.142857	2.941176	0.021709	2.295425
V102=V102_Non	56.521739	37.142857	27.058824	0.021977	2.290759
V117=V117_Non	53.225806	47.142857	36.470588	0.024286	2.252572
V007=NA	43.670886	98.571429	92.941176	0.026823	2.214079
V006=NA	43.670886	98.571429	92.941176	0.026823	2.214079
V122=V122_Non	48.484848	68.571429	58.235294	0.032474	2.138522
V130=50%	68.75	15.714286	9.411765	0.038006	2.074788
V107=V107_Maison	68.75	15.714286	9.411765	0.038006	2.074788
V005=NA	43.396226	98.571429	93.529412	0.043868	2.015347
V137=NA	42.944785	100	95.882353	0.044564	2.008743
V111=V111_Efficacement	47.222222	72.857143	63.529412	0.049587	1.963511

	Cla/Mod	Mod/Cla	Global	p.value	v.test
V109=V109_Ma_formation	20.833333	7.142857	14.117647	0.044691	-2.007554
V202=V202_Amelioration_cours	9.090909	1.428571	6.470588	0.043868	-2.015347
V125=NA	9.090909	1.428571	6.470588	0.043868	-2.015347
V109=V109_Passetemps	19.047619	5.714286	12.352941	0.043628	-2.017649
V134=NA	13.333333	2.857143	8.823529	0.035412	-2.103617
V124=V124_Ne_sait_pas	20	7.142857	14.705882	0.030364	-2.165307
V001=Enseignant	8.333333	1.428571	7.058824	0.026823	-2.214079
V009=V009_GM	0	0	4.705882	0.025426	-2.234863
V10=NA	20.689655	8.571429	17.058824	0.020773	-2.312085
V117=NA	0	0	5.294118	0.01444	-2.446136
V127=V127_Ne_sait_pas	0	0	5.294118	0.01444	-2.446136
V121=NA	0	0	5.294118	0.01444	-2.446136
V009=V009_GMI	0	0	5.294118	0.01444	-2.446136
V128=V128_Doc_manuscris	26	18.571429	29.411765	0.013975	-2.457915
V125=V125_Ne_sait_pas	26	18.571429	29.411765	0.013975	-2.457915
V201=V201_Un_pou	23.076923	12.857143	22.941176	0.013117	-2.480567
V204=NA	23.809524	14.285714	24.705882	0.012393	-2.500762
V204=V204_Exige_moyen_coiteux	14.285714	4.285714	12.352941	0.010928	-2.544991
V104=V104_Sonvent	19.354839	8.571429	18.235294	0.009223	-2.603663
V101=V101_Non	21.621622	11.428571	21.764706	0.009201	-2.604488
V120A=V120A_Programmation	6.666667	1.428571	8.823529	0.005775	-2.760265
V128=NA	6.666667	1.428571	8.823529	0.005775	-2.760265
V126=NA	0	0	6.470588	0.004591	-2.834422
V124=NA	0	0	6.470588	0.004591	-2.834422
V131=V131_Non-dispo	0	0	7.058824	0.00257	-3.015007
V130=25%	5.882353	1.428571	10	0.001984	-3.092685
V116=Acces_tout_moment	9.52381	2.857143	12.352941	0.001926	-3.10139
V114=V114_Dbit_très_lent	20.454545	12.857143	25.882353	0.001648	-3.147232
V127=NA	0	0	7.647059	0.001431	-3.188275
V10=Industriel	0	0	7.647059	0.001431	-3.188275
V202=V202_Offre_cours_exo	5.555556	1.428571	10.588235	0.001148	-3.251498
V008=NA	5.263158	1.428571	11.176471	0.000659	-3.400644
V008=V008_3_ANNEE	0	0	8.823529	0.000437	-3.516553
V008=V008_1_ANNEE	0	0	9.411765	0.00024	-3.627947
V008=V008_2_ANNEE	8	2.857143	14.705882	0.000227	-3.687324
V008=V008_4_ANNEE	0	0	10	0.000131	-3.824979
V132=V132_Oui	9.090909	4.285714	19.411765	0.000021	-4.250472
V113=V113_Oui	27.777778	42.857143	63.529412	0.000006	-4.53296
V120=NA	15.789474	12.857143	33.529412	0.000002	-4.777279
V126=V126_Ne_sait_pas	9.090909	5.714286	25.882353	0	-5.165813
V134=V134_Non	24.761959	37.142857	61.764706	0	-5.402082
V009=V009_GE	0	0	24.117647	0	-6.8181
V010=Technique	0	0	25.882353	0	-7.151542
V002=Homme	17.647059	25.714286	60	0	-7.638946
V135=NA	6.060606	5.714286	38.823529	0	-7.762476
V001=Etudiant	2.409639	2.857143	48.823529	0	-10.72982

Tableau C.3 – Tableau représentant les valeurs de Cla/Mod, p.value et v.test des variables constituant la classe 1+ à gauche et classe 1- à droite

	Cla/Mod	Mod/Cla	Global	p.value	v.test
V001=Enseignant	83.333333	100	7.058824	0	7.593583
V008=NA	52.631579	100	11.176471	0	6.591451
V010=NA	34.482759	100	17.058824	0	5.740427
V009=V009_PROF	100	50	2.941176	0	5.045641
V117=NA	66.666667	60	5.294118	0.000001	4.876554
V121=NA	66.666667	60	5.294118	0.000001	4.876554
V126=NA	54.545455	60	6.470588	0.000006	4.53768
V125=NA	54.545455	60	6.470588	0.000006	4.53768
V124=NA	54.545455	60	6.470588	0.000006	4.53768
V007=V007_10	71.428571	50	4.117647	0.000009	4.439479
V006=V006_30+	100	40	2.352941	0.000013	4.368584
V005=MATH	100	40	2.352941	0.000013	4.368584
V127=NA	46.153846	60	7.647059	0.00002	4.262362
V134=NA	40	60	8.823529	0.000056	4.027116
V128=NA	40	60	8.823529	0.000056	4.027116
V132=NA	26.923077	70	15.294118	0.000159	3.776661
V133=NA	22.580645	70	18.235294	0.000158	3.440782
V006=V006_10	75	30	2.352941	0.001156	3.249554
V119=NA	16.326531	80	28.823529	0.001665	3.144314
V122=NA	22.222222	60	15.882353	0.002542	3.018332
V120=NA	14.035088	80	33.529412	0.00543	2.780368
V007=V007_30+	100	20	1.176471	0.006265	2.733568
V007=V007_20	100	20	1.176471	0.006265	2.733568
V006=V006_30	100	20	1.176471	0.006265	2.733568
V129=NA	14	70	29.411765	0.01518	2.428061
V135=NA	12.121212	80	38.823529	0.016419	2.399477
V130=75%	66.666667	20	1.764706	0.018199	2.361544
V131=NA	11.940299	80	39.411765	0.018347	2.358533
V107=V107_Lycée/Université	18.518519	50	15.882353	0.020215	2.322341
V109=V109_Cadre_Enseignement	17.857143	50	16.470588	0.023922	2.258377
V105=V105_Saisie_cours	9.782609	90	54.117647	0.035349	2.104339

	Cla/Mod	Mod/Cla	Global	p.value	v.test
V131=V131_Pas_moyen	1.31579	10	44.705882	0.042377	-2.029795
V009=V009_A	0	0	31.176471	0.042055	-2.03298
V134=V134_Non	1.904762	20	61.764706	0.01466	-2.44066
V120=V120_Oui	1.869159	20	62.941176	0.01161	-2.523795
V001=Elève	0	0	44.117647	0.004764	-2.822551
V010=Général	0	0	44.117647	0.004764	-2.822551
V127=V127_Efficacement	1	10	58.823529	0.003253	-2.942735
V119=V119_Apprentissage	0	0	47.647059	0.002398	-3.035986
V001=Etudiant	0	0	48.823529	0.001886	-3.107575
V133=V133_Non	1.587302	20	74.117647	0.000701	-3.389055
V132=V132_Non	0.900901	10	65.294118	0.000687	-3.394618
V007=NA	0	0	92.941176	0	-7.593583
V006=NA	0	0	92.941176	0	-7.593583
V005=NA	0	0	93.529412	0	-7.822309

Tableau C.4 – Tableau représentant les valeurs de Cla/Mod, p.value et v.test des variables constituant la classe 2+ à gauche et classe 2- à droite

Annexe C. Résultats intégration TIC par la méthode AFM

	Cla/Mod	Mod/Cla	Global	p.value	v.test
V001=Etudiant	97,590361	90	48,823529	0	12,345851
V010=Technique	100	48,888889	25,882353	0	7,956828
V009=V009_GE	100	45,555556	24,117647	0	7,584088
V002=Homme	74,509804	84,444444	60	0	6,905875
V134=V134_Non	73,333333	85,555556	61,764706	0	6,770041
V126=V126_Ne_sait_pas	90,909091	44,444444	25,882353	0	6,019038
V135=NA	81,818182	60	38,823529	0	6,013701
V113=V113_Oui	65,740741	78,888889	63,529412	0,000018	4,28713
V008=V008_4_ANNEE	100	18,888889	10	0,000018	4,283751
V008=V008_2_ANNEE	92	25,555556	14,705882	0,000018	4,283706
V008=V008_1_ANNEE	100	17,777778	9,411765	0,000038	4,17842
V132=V132_Oui	84,848485	31,111111	19,411765	0,000052	4,048036
V008=V008_3_ANNEE	100	16,666667	8,823529	0,000079	3,947302
V133=V133_Non	61,904762	86,666667	74,117647	0,000133	3,821712
V202=V202_Offre_cours_exo	94,444444	18,888889	10,588235	0,000182	3,743152
V010=Industriel	100	14,444444	7,647059	0,000331	3,589913
V005=NA	56,603774	100	93,529412	0,00034	3,583032
V130=25%	94,117647	17,777778	10	0,000356	3,570638
V128=V128_Doc_manuscrits	74	41,111111	29,411765	0,000599	3,431912
V125=V125_Ne_sait_pas	74	41,111111	29,411765	0,000599	3,431912
V120A=V120A_Programmation	93,333333	15,555556	8,823529	0,001327	3,210788
V120=NA	70,175439	44,444444	33,529412	0,002191	3,063033
V007=NA	56,329114	98,888889	92,941176	0,002454	3,028919
V006=NA	56,329114	98,888889	92,941176	0,002454	3,028919
V101=V101_Non	75,675676	31,111111	21,764706	0,002713	2,998465
V201=V201_Un_peu	74,358974	32,222222	22,941176	0,003562	2,914538
V124=V124_Ne_sait_pas	80	22,222222	14,705882	0,00531	2,784613
V127=V127_Ne_sait_pas	100	10	5,294118	0,005361	2,784496
V009=V009_GMI	100	10	5,294118	0,005361	2,784496
V124=V124_Un_peu	66,666667	48,888889	38,823529	0,006657	2,713511
V131=V131_Non-dispo	91,666667	12,222222	7,058824	0,008813	2,619242
V204=V204_Excige_moyen_coteux	71,428571	33,333333	24,705882	0,008867	2,617146
V109=V109_Ma_formation	79,166667	21,111111	14,117647	0,008873	2,616918
V108=V108_Rarement	72,972973	30	21,764706	0,009106	2,608602
V116=Accs_tout_moment	80,952381	18,888889	12,352941	0,009862	2,580647
V109=V109_Passetemps	80,952381	18,888889	12,352941	0,009862	2,580647
V009=V009_GM	100	8,888889	4,705882	0,010591	2,558988
V104=V104_Souvent	74,193548	25,555556	18,235294	0,013937	2,458876
V202=V202_Amelioration_cours	90,909091	11,111111	6,470588	0,016183	2,404758
V131=V131_Pas_moyen	63,157895	53,333333	44,705882	0,024369	2,251261
V122=V122_Oui	68,181818	33,333333	25,882353	0,028319	2,192845
V114=V114_Dbit_trs_lent	68,181818	33,333333	25,882353	0,028319	2,192845
V205=V205_Amelioration_cours	90	10	5,882353	0,029336	2,17894
V116=Information_dispo_mat	76,190476	17,777778	12,352941	0,037537	2,079875
V105=V105_Distracton	76,190476	17,777778	12,352941	0,037537	2,079875
V120=V120_Non	100	6,666667	3,529412	0,04061	2,047494
V102=NA	100	6,666667	3,529412	0,04061	2,047494

	Cla/Mod	Mod/Cla	Global	p.value	v.test
V009=V009_PROF	0	0	2,941176	0,04312	-2,022548
V009=V009_D	0	0	2,941176	0,04312	-2,022548
V107=V107_Maison	25	4,444444	9,411765	0,035208	-2,105957
V137=NA	50,920245	92,222222	95,882353	0,0208	-2,311595
V203=V203_Oui	47,2	65,555556	73,529412	0,019078	-2,34401
V135=V135_Pas_du_tout	34,210526	14,444444	22,352941	0,014392	-2,447336
V202=V202_Assist_Eleve_faible	22,222222	4,444444	10,588235	0,010991	-2,542969
V104=V104_Rarement	25	6,666667	14,117647	0,005674	-2,76603
V205=V205_Favorise_Appren_Rech	42,857143	46,666667	57,647059	0,003351	-2,933562
V123=NA	45,528455	62,222222	72,352941	0,002723	-2,997405
V001=Enseignant	8,333333	1,111111	7,058824	0,002454	-3,028919
V131=V131_Autre	13,333333	2,222222	8,823529	0,002424	-3,032677
V201=V201_Efficacement	45,081967	61,111111	71,764706	0,001671	-3,143204
V101=V101_Oui	44,067797	57,777778	69,411765	0,000754	-3,369006
V115=V115_Pas_le_temps	20	5,555556	14,705882	0,000635	-3,416103
V204=V204_Asservit_homme	17,391304	4,444444	13,529412	0,000421	-3,526665
V116=Service_gratuit	35,714286	27,777778	41,176471	0,000287	-3,626899
V202=V202_Apport_motiv	11,111111	2,222222	10,588235	0,000272	-3,640833
V135=V135_Un_peu	32,142857	20	32,941176	0,000247	-3,665138
V120A=V120A_Internet	20,689655	6,666667	17,058824	0,000234	-3,678696
V109=V109_mail	10,526316	2,222222	11,176471	0,000127	-3,831943
V120=V120_Oui	41,121495	48,888889	62,941176	0,000091	-3,913597
V126=V126_Efficacement	28,301887	16,666667	31,176471	0,000026	-4,203554
V128=V128_Livres	36,666667	36,666667	52,941176	0,000011	-4,399978
V125=V125_Efficacement	17,647059	6,666667	20	0,000006	-4,521117
V113=NA	22,727273	11,111111	25,882353	0,000005	-4,556553
V124=V124_Efficacement	26,229088	17,777778	35,882353	0	-5,124449
V008=V008_1ERE	0	0	11,764706	0	-5,27135
V132=NA	6,451613	2,222222	18,235294	0	-5,856363
V132=NA	0	0	15,294118	0	-6,227409
V002=V002_Femme	20,588235	15,555556	40	0	-6,905875
V134=V134_Oui	12	6,666667	29,411765	0	-6,993781
V008=V008_TERMINALE	0	0	22,352941	0	-7,971329
V010=Général	14,666667	12,222222	44,117647	0	-9,110543
V009=V009_A	0	0	31,176471	0	-10,074111
V001=Elève	10,666667	8,888889	44,117647	0	-10,211705

Tableau C.5 – Tableau représentant les valeurs de Cla/Mod, p.value et v.test des variables constituant la classe 3+ à gauche et classe 3- à droite

Annexe D

Résultats d'évaluation des classes de seconde en algorithmique

D.1 Notes des étudiants en langues et maths-info

Note N°	/60 MALG	/40 FRS	/40 ANG	/60 MATHS	/20 INFO
1	24	24	20	40,5	13,5
2	33	31	20	33	12
3	27	20	18	21	9
4	36	25	22	12	9
5	19,5	20	18	21	6
6	33	15	14	18	7
7	46,5	21	18	33	12
8	21	14	20	16,5	12
9	51	6	16	21	9
10	27	23	20	15	11
11	15	28	30	15	10
12	37,5	25	29	21	13
13	40,5	31	26	24	13
14	39	20	22	24	13,5
15	30	29	18	6	8
16	45	23	18	21	11
17	51	38	38	49,5	13
18	18	26	18	13,5	8
19	34,5	21	16	33	6
20	13,5	20	16	15	9
21	37,5	22	28	30	11
22	36	20	22	16,5	11
23	12	10	20	18	11
24	24	13	16	12	9
25	34,5	33	35	21	11
26	18	9	8	9	6
27	9	9	10	10,5	9
28	36	12	14	24	8
29	18	15	22	10,5	11
30	27	17	22	10,5	9
31	9	26	12	21	11
32	48	30	30	30	10
33	30	17	16	27	12
34	39	18	16	15	9
35	45	27	31	21	15
36	10,5	5	10	15	11
37	48	33	30	15	11
38	34,5	15	16	21	13

Tableau D.1 – Résultats de malagasy, français, anglais, mathématiques et informatique de la classe de seconde I

N°	MALG faible	MALG moy	MALG fort	FRS faible	FRS moy	FRS fort	ANG faible	ANG moy	ANG fort	MATHS faible	MATHS moy	MATHS fort	INFO faible	INFO moy	INFO fort
1	1	0	0	0	1	0	0	1	0	0	0	1	0	0	1
2	1	1	0	0	0	1	0	1	0	0	1	0	0	1	0
3	1	0	0	0	1	0	1	0	0	1	0	0	1	0	0
4	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0
5	1	0	0	0	1	0	1	0	0	1	0	0	1	0	0
6	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0
7	0	0	1	0	1	0	1	0	0	0	1	0	0	1	0
8	1	0	0	1	0	0	1	1	0	1	0	0	0	1	0
9	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0
10	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0
11	1	0	0	0	1	0	0	1	0	1	0	0	0	1	0
12	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1
13	0	0	1	0	0	1	0	1	0	0	0	0	0	0	1
14	0	0	1	0	0	1	0	1	0	0	0	0	0	0	1
15	0	1	0	0	1	0	1	0	0	0	0	0	1	0	0
16	0	0	1	0	1	0	1	0	0	0	0	0	0	1	0
17	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
18	1	0	0	0	1	0	1	0	0	1	0	0	1	0	0
19	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0
20	1	0	0	0	1	0	1	0	0	1	0	0	1	0	0
21	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0
22	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0
23	1	0	0	1	0	0	0	1	0	1	0	0	0	1	0
24	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
25	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0
26	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
27	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
28	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0
29	1	0	0	1	0	0	0	1	0	1	0	0	0	1	0
30	1	0	0	1	0	0	0	1	0	1	0	0	1	0	0
31	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0
32	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0
33	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0
34	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0
35	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1
36	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0
37	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0
38	0	1	0	1	0	0	1	0	0	1	0	0	0	0	1

Tableau D.2 – Table binaire de la classe de seconde I, tirée du tableau D.1

Note N°	/60 MALG	/40 FRS	/40 ANG	/60 MATHS	/20 INFO
1	4,5	7	14	3	13
2	12	21	24	19,5	8
3	16,5	24	20	16,5	13
4	13,5	6	10	18	14
5	30	25	16	18	11
6	30	24	30	22,5	11
7	42	23	29	31,5	13
8	24	14	12	12	8,5
9	18	15	10	15	8
10	33	13	8	15	11
11	36	21	28	25,5	6
12	13,5	4	12	12	11
13	16,5	20	22	9	9
14	34,5	10	26	15	13
15	42	35	28	37,5	10
16	18	11	20	9	9
17	4,5	11	12	6	10
18	4,5	3	2	3	8
19	25,5	23	29	27	8,5
20	36	23	26	7,5	13
21	13,5	14	12	9	13
22	13,5	10	12	9	5
23	12	16	16	7,5	7
24	18	6	12	6	5
25	45	32	22	21	14
26	7,5	3	10	9	4
27	10,5	7	6	16,5	12
28	10,5	8	12	7,5	9
29	24	16	18	10,5	14
30	49,5	28	28	22,5	11
31	12	0	22	0	11
32	18	12	14	19,5	11
33	40,5	27	34	19,5	11
34	21	12	20	3	0
35	13,5	4	10	4,5	11
36	18	14	16	6	12
37	15	11	10	9	5
38	51	20	20	18	8

Tableau D.3 – Résultats de malagasy, français, anglais, mathématiques et informatique de la classe de seconde II

N°	MALG faible	MALG moy	MALG fort	FRS faible	FRS moy	FRS fort	ANG faible	ANG moy	ANG fort	MATHS faible	MATHS moy	MATHS fort	INFO faible	INFO moy	INFO fort
1	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1
2	1	0	0	0	1	0	0	1	0	1	0	0	1	0	0
3	1	0	0	0	1	0	0	1	0	1	0	0	0	0	1
4	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1
5	0	1	0	0	1	0	1	0	0	1	0	0	0	1	0
6	0	1	0	0	1	0	0	1	0	1	0	0	0	1	0
7	0	0	1	1	0	0	1	0	0	0	1	0	0	0	1
8	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
9	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
10	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0
11	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0
12	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
13	1	0	0	0	1	0	0	1	0	1	0	0	1	0	0
14	0	1	0	0	1	0	0	1	0	1	0	0	0	0	1
15	0	0	1	0	0	1	0	1	0	0	0	1	0	1	0
16	1	0	0	1	0	0	0	1	0	1	0	0	1	0	0
17	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
18	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
19	1	0	0	0	1	0	0	1	0	1	0	0	1	0	0
20	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0
21	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
22	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
23	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
24	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
25	0	0	1	0	0	1	0	1	0	1	0	0	0	0	1
26	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
27	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
28	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
29	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
30	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0
31	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
32	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
33	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0
34	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
35	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
36	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
37	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
38	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0

Tableau D.4 – Table binaire de la classe de seconde II, tirée du tableau D.3

Résumé

L'analyse statistique implicative (ASI) est un outil d'analyse de données dédié initialement pour les recherches en didactique de disciplines. Dans cette thèse nous réalisons l'implémentation sur un langage de programmation des nouvelles algorithmes de construction du dendrogramme représentant la classification hiérarchique cohésive et implicative selon la nouvelle cohésion basée sur M_{GK} . Ce travail comprend aussi l'utilisation de l'outil logiciel ainsi élaboré sur des données, nous permettant d'analyser des curriculum de l'informatique dans le modèle anglosaxon, comparés au curriculum de l'informatique de l'ENSET. L'analyse obtenue nous a permis de créer et d'adapter un nouveau curriculum de l'informatique, et plus particulièrement l'algorithmique et programmation en classe de seconde au lycée de Madagascar. Comme, la formation des jeunes exige une démarche de pensée nouvelle « *modélisante, algorithmique et organisationnelle* » qui est une pensée informatique liée à la science de traitement de l'information. Ce programme d'informatique est dans ce cas pris comme pré-requis des élèves qui voudraient poursuivre le cursus Informatique au niveau supérieur. Nous proposons aussi dans ce travail quelques méthodes d'apprentissage de l'algorithmique et programmation en utilisant le logiciel SCRATCH en tenant compte les deux composantes qui nous paraissent inévitables, « *Communauté et TIC* » dans le système pédagogique.

Mots-clés: Règles d'association, Mesure de qualité, Classification Hiérarchique Implicative et Cohésive, TIC, curriculum, algorithme.

Abstract

Statistical Implicative Analysis (SIA) is a data analysis tool initially dedicated to research in didactic disciplines. In this thesis we realize the implementation on a programming language of the new dendrogram construction algorithms representing the cohesive and implicative hierarchical classification according to the new cohesion based on M_{GK} . This work also includes the use of the software tool thus developed on data, allowing us to analyze curricula of computer science in the Anglo-Saxon model, compared to the computer science curriculum of ENSET. The analysis obtained allowed us to create and adapt a new curriculum of computer science, and more particularly the algorithmic and programming in second class at the College of Madagascar. As, youth training requires a new thinking approach « *modelling, algorithmic and organizational* » that is a computer thinking related to the science of information processing. This computer program is in this case taken as a prerequisite for students who would like to continue the computer science course at the higher level. We also propose in this work some algorithmic and programming learning methods using the SCRATCH software taking into account the two components that seem to us inevitable, « *Community and ICT* » in the pedagogical system.

Keys-words: Association rules - Quality of measures - Hierarchical Implicative and Cohesive Classification, ICT, curriculum, algorithm.

