Extraction de données symboliques et cartes topologiques: application aux données ayant une structure complexe

Aïcha El Golli

Université Paris IX Dauphine

Thèse

présentée par

Aïcha El Golli

pour obtenir le grade de docteur de l'Université Paris IX Dauphine

Discipline : Informatique

Extraction de données symboliques et cartes topologiques : Application aux données ayant une structure complexe

Soutenue le 1 Juin 2004 devant le jury composé de :

Diday	Edwin	Directeur de thèse
Lechevallier	Yves	Co-Directeur de thèse
Bouzeghoub	Mokrane	Rapporteur
Saporta	Gilbert	Rapporteur
Hébrail	Georges	Examinateur
Rossi	Fabrice	Examinateur
Cazes	Pierre	Professeur Invité

Thèse préparée au sein du Centre de Recherche de Mathématiques de la Décision (Université Paris-IX Dauphine) et de l'INRIA (Projet AXIS, Centre de Rocquencourt)

Résumé

Un des objectifs de l'analyse de données symboliques est de permettre une meilleure modélisation des variations et des imprécisions des données réelles. Ces données expriment en effet, un niveau de connaissance plus élevé, la modélisation doit donc offrir un formalisme plus riche que dans le cadre de l'analyse de données classiques. Un ensemble d'opérateurs de généralisation symbolique existent et permettent une synthèse et représentation des données par le formalisme des assertions, formalisme défini en analyse de données symboliques. Cette généralisation étant supervisée, est souvent sensible aux observations aberrantes. Lorsque les données que l'on souhaite généraliser sont hétérogènes, certaines assertions incluent des observations virtuelles. Face à ce nouveau formalisme et donc cette extension d'ordre sémantique que l'analyse de données symbolique a apporté, une nouvelle approche de traitement et d'interprétation s'impose. Notre objectif au cours de ce travail est d'améliorer tout d'abord cette généralisation et de proposer ensuite une méthode de traitement de ces données.

Les contributions originales de cette thèse portent sur de nouvelles approches de représentation et de classification des données à structure complexe. Nous proposons donc une décomposition permettant d'améliorer la généralisation tout en offrant le formalisme symbolique. Cette décomposition est basée sur un algorithme divisif de classification. Nous avons aussi proposé une méthode de généralisation symbolique non supervisée basée sur l'algorithme des cartes topologiques de Kohonen. L'avantage de cette méthode est de réduire les données d'une manière non supervisée et de modéliser les groupes homogènes obtenus par des données symboliques. Notre seconde contribution porte sur l'élaboration d'une méthode de classification traitant les données à structure complexe. Cette méthode est une adaptation de la version batch de l'algorithme des cartes topologiques de Kohonen aux tableaux de dissimilarités. En effet, seule la définition d'une mesure de dissimilarité adéquate, est nécessaire pour le bon déroulement de la méthode.

Mot-clés : Analyse de données, analyse de données symboliques, base de données relationnelle, généralisation, classification non supervisée, algorithme divisif, cartes topologiques de Kohonen, dissimilarité.

Abstract

The aim of symbolic data analysis is to provide a better representation of the variations and imprecision contained in real data. As such data express a higher level of knowledge; the representation must offer a richer formalism than that provided by classical data analysis. A generalization process exists that allows data to be synthesized and represented by means of an assertion formalism that was defined in symbolic data analysis. This generalization process is supervised and often sensitive to virtual and atypical individuals. When the data to be generalized is heterogeneous, some assertions include virtual individuals. Faced with this new formalism and the resulting semantic extension that symbolic data analysis offers, a new approach to processing and interpreting data is required.

The original contributions of our work concern new approaches to representing and clustering complex data.

First, we propose a decomposition step, based on a divisive clustering algorithm, that improves the generalization process while offering the symbolic formalism. We also propose a unsupervised generalization process based on the self-organizing map. The advantage of this method is that it enables the data to be reduced in an unsupervised way and allows the resulting homogeneous clusters to be represented by symbolic formalism.

The second contribution of our work is a development of a clustering method to handle complex data. The method is an adaptation of the batch version of the self-organizing map to dissimilarity tables. Only the definition of an adequate dissimilarity is required for the method to operate efficiently.

Key-words:

Data analysis, symbolic data analysis, relational database, generalization, unsupervised clustering, divisive algorithm, self-organizing map, dissimilarity

Remerciements

Tout au long de ce travail, et plus généralement de mon parcours réalisé ces dernières années, j'ai pu bénéficier de soutien, des conseils ou encore des encouragements d'un très grand nombre de personnes auxquelles je tiens ici à exprimer toutes ma reconnaissance.

En premier lieu, je voudrais exprimer ici ma reconnaissance envers les membres du jury. Avoir pu réunir à cette occasion des chercheurs d'un tel niveau au sein de disciplines aussi diverses a été pour moi un véritable honneur et une marque d'encouragement à la conduite de recherches interdisciplinaires.

Je tiens à leur exprimer toute ma gratitude pour leur disponibilité et la qualité des remarques dont ils m'ont fait part au cours de la soutenance, et, en tout premier chef, Mokrane BOUZEGHOUB, Professeur à l'Université de Versailles, et Gilbert SAPORTA, professeur au Conservatoire National des Arts et Metiers, qui ont accepté de juger ce travail et d'en être les rapporteurs. Je les remercie pour l'application avec laquelle ils ont lu mon manuscrit et tous les problèmes, riches d'intérêt, qu'ils ont pu soulevés et qui sont encore vifs dans mon esprit.

Je remercie égalemenr Monsieur Edwin DIDAY, Professeur à l'Université de Paris Dauphine, de m'avoir proposé d'effectuer une thèse sous sa direction. J'ai particulièrement apprécié l'enthousiasme dont il fait preuve dans son travail.

Je suis très sensible à la présence dans ce jury de Georges HEBRAIL, professeur à l'École National de Télécommunication de Paris, et Pierre CAZES, Professeur à l'Université de Paris Dauphine, avec lesquels j'ai eu l'occasion de discuter dans le cadre des différents Workshops et conférences. Je les remercie pour l'intérêt qu'ils ont porté à ce travail et pour leurs suggestions constructives. Un grand merci à Monsieur Georges Hébrail pour m'avoir fait l'honneur de présider ce jury.

Ce document est finalement le fruit d'un travail collectif dont je suis l'heureuse dépositaire. À ce titre, je voudrais remercier les personnes dont la compétence, la patience et les conseils avisés m'ont permis de mûrir ce projet de recherche et de le mener à terme. Je les remercie également tout naturellement d'avoir accepté de faire partie de

mon jury. Mes premiers mots vont à Yves LECHEVALLIER, Directeur de Recherches à l'INRIA, qui m'a ouvert les portes du projet *AxIS* et qui a contribué grandement à l'élaboration de cette thèse. Sa grande compétence scientifique, ses précieux conseils et encouragements m'ont permis de mener à bien ce travail. Ses qualités humaines m'ont aidé à mener cette recherche dans la bonne humeur et à surmonter toutes les difficultés rencontrées pendant ces trois années de thèse. Je le remercie également sincèrement de m'avoir permis de réaliser ce travail dans d'excellentes conditions. Je lui en suis très reconnaissante.

Je me dois finalement d'exprimer ma profonde gratitude aux deux personnes qui ont compté aussi pour l'achèvement de cette thèse, celles qui m'ont accordé leur confiance et prodigué leur aide au cours de ces années. Je remercie donc Fabrice ROSSI, matre de conférence à l'Université de Paris Dauphine, et Brieuc CONAN-GUEZ, Docteur en Mathématiques Appliquées. Tous deux ont su m'inssufler leur enthousiasme, leur curiosité et leur engouement scientifique. Merci à Fabrice d'avoir incité ce travail et suivi avec attention son évolution, d'avoir su orienter mes recherches aux bons moments. Merci également d'avoir anticipé et provoqué les réajustements nécessaires avec la préoccupation constante de me donner les moyens de poursuivre des objectifs réalistes. Son ouverture et ses compétences scientifiques, son enthousiasme très communicatif m'ont également été très précieux. Merci à Brieuc qui a probablement été la personne la plus déterminante dans l'aboutissement de ce travail. Les conseils qu'il m'a prodigués ont toujours été clairs et riches, me facilitant grandement le travail, et me permettant d'aboutir à la production de cette thèse. Ses compétences, sa rigueur, sa disponibilité de tous les instants, ses critiques et ses encouragements y sont évidemment pour beaucoup. merci cher ami pour ton écoute, ta présence et tes encouragements!

Mes remerciements vont encore à tous les membres du projet *AxIS*, qui m'ont permis de passer des années très agréables et enrichissantes. J'exprime toute ma gratitude envers tous ceux qui m'ont aidé et encouragé tout au long de mon séjour au sein du projet.

J'ai eu l'occasion de côtoyer encore bien d'autres personnes au cours de ma thèse, stagiaires, thésards, chercheurs et assistantes qui ont contribué au bon déroulement de

cette thèse aussi bien sur le plan humain que scientifique. Je remercie donc les équipes du bâtiment 18, des projets *Mirages*, *Merlins* et *Reflecs*.

Je voudrais remercier tous mes "amis" et connaissances. J'adresse tout d'abord mes remerciements aux thétards et autres chercheurs qui m'ont accompagné pendant des mois et des mois, que j'ai croisés *quasi*-quotidiennement et avec qui j'ai eu l'occasion et le plaisir de partager des pauses café inoubliables: Brieuc Conan Guez, Jacopo Grazzini, mon cher Julien Fauqueur (Un grand merci aux trois pour leur aide dans la correction et la rédaction des chapitres), Antonio Turiel, Alexis Paljic, Karine Blin, Franois Perron, Richard Roussel, Noelly Grondin, Vincent Lucquiaut, Sandrine Fauqueux, Hatem Charfi. Dans la quête de l'improductif, j'assume le temps passé avec eux sur les vertes pelouses, sur les terrains de squash ou de tennis, sur les terrasses des cafés ou encore dans les soirées couscous et autres. Merci pour votre présence, votre soutiens moral et votre joie de vivre.

Merci à tous mes amis de très longue date qui m'ont aidé et soutenue tout au long de ce travail et particulièrement: Chiraz, selima, Senda, Méniar, Sarra, Haifa, Mehdi, Jawhar et Jacques. Merci pour toute votre affection et amitié dévouée qui m'ont longuement et profondement soutenue.

Ces trois années resteront ancrées comme une période enrichissante de ma vie. Cet aboutissement au niveau des études doit être finement associé au soutien des personnes proches de moi, celles qui ont contribué, moins directement, à l'aboutissement de mes travaux. Je pense à la famille, une valeur à laquelle j'attache énormément d'importance, et particulièrement à "ma famille". Une maman attentive, affectueuse, courageuse et admirable. "Rabbi yfadhlek" رَبِّ يُفضَلِك *Maman*, merci pour ton soutiens et ton dévouement. Mes deux frères et ma *belle*-sœur (belle certes mais surtout MA sœur) formidables. Je tiens à vous dire merci d'avoir été toujours là, pour la compréhension, la patience et l'affection que vous avez manifestées durant ces années passées et pour votre aide morale et financière :). Je remercie mon petit "Garmouche" قَرمُوشِي adoré, mon adorable neuveu *Iyed* qui par sa bonne humeur a su me détendre et m'apporter beaucoup de joie.

Enfin je dédie cette thèse à mon cher père qui m'a dit au revoir le 10 Mars 2003 mais reste bien présent dans mon coeur. Papa, tu t'es beaucoup investi dans la qualité de mes précedents documents mais malheureusement t'as pas connu celui là. Papa, toi qui m'a tout appris avec bonne humeur et grands fous rires et qui m'a permis d'arriver à ce niveau d'études, que dieu te garde au paradis et "*allah yarhmek*" ٱللَّهِ يَرحَمِك.

Table des matières

Introduction Générale

Contexte et objectifs

L'objectif de l'analyse de données symboliques, développée par E. Diday [DK91], est de permettre une meilleure modélisation des données réelles en offrant un formalisme plus riche que dans le cadre classique. Ainsi, les individus traités en analyse de données symboliques se distinguent des individus classiquement traités en analyse de données, au niveau de leur description. Cette description doit permettre de prendre en compte l'imperfection de l'information, fréquemment rencontrée en pratique dans les traitements de données. Par exemple, une variable qualitative peut être décrite par plusieurs modalités. Une variable quantitative peut être décrite par un intervalle de valeurs traduisant la variation d'un individu ou d'un groupe d'individus. Dans les deux cas, on ne transforme pas ces descriptions en une modalité unique, afin de ne pas perdre l'information contenue dans ces descriptions.

L'avancée dans les domaines des bases de données permet à présent de gérer une connaissance plus riche et plus complexe. L'analyse de grands volumes d'information constitue l'un des problèmes majeurs auxquels sont confrontés les entreprises. La synthèse d'information apporte un premier élément de réponse à l'exploration de grands volumes de données. Elle s'avère être à la fois un outil descriptif pour l'utilisateur mais elle peut aussi être vue comme une étape intermédiaire pour d'autres analyses exploratoires. La synthèse de grands volumes de données, à partir de bases de données relationnelles, en descriptions se fait dans le cadre de l'analyse de données symboliques par une méthode de généralisation [Ste98], [SHL00] comprenant la capacité de réduire le volume

des données et celle de le recouvrir. La qualité de généralisation est alors fondée sur le choix d'un bon compromis entre la réduction du volume et la perte d'information qu'elle induit. Parmi les types de données symboliques définis par E. Diday [DB89] [Did98], le formalisme des assertions est utilisé pour modéliser les résultats de cette généralisation. Ce formalisme permet des représentations en intervalles, en ensemble de valeurs ou encore des distributions de fréquences. L'avantage de telles représentations est de prendre en compte la notion de variabilité dans la généralisation. Donc, cette généralisation constitue une première et importante étape de l'analyse de données symboliques car elle permet d'associer aux groupes d'individus une description de nature symbolique.

La généralisation proposée dans le cadre de l'analyse de données symboliques, étant supervisée, est sensible aux observations aberrantes. En effet, lorsque les données que l'on souhaite généraliser sont très hétérogènes, certaines descriptions incluent des individus dont la probabilité d'être éléments des groupes de départ est faible, éléments appelés atypiques ou virtuels. Une idée de réduction a déjà été développée dans le cadre des travaux de la thèse de Véronique Stéphan [Ste98]. Cette réduction a permis surtout d'éliminer les individus atypiques mais sans pour autant améliorer l'homogénéité de certaines descriptions. **Notre premier objectif sera donc d'améliorer le procédé de généralisation en décomposant les groupes généralisés**. Il s'agit d'une part de définir la méthode de décomposition. D'autre part, il convient de définir un bon formalisme qui puisse représenter et modéliser les données obtenues par cette décomposition. **Nous proposons aussi une méthode de généralisation basée sur un algorithme de classification automatique offrant une modélisation symbolique des groupes homogènes obtenus**.

On oppose parfois l'analyse de données symboliques, liée à une approche statistique du traitement des données, à l'apprentissage automatique lié à une approche symbolique du traitement des données. Cette deuxième approche a été développée essentiellement dans le domaine de l'Intelligence Artificielle. On distingue l'apprentissage non supervisé de l'apprentissage supervisé, correspondant respectivement aux problèmatiques de la classification et de la discrimination. Les méthodes d'apprentissage non supervisé encore appelées méthodes de classification conceptuelle ou méthodes d'apprentissage à partir

d'observations, remontent essentiellement aux travaux de [MDS82] et [MS83]. Ces méthodes de classification ont la particularité de chercher simultanément une structuration en classes d'un ensemble d'objets et une description conceptuelle de chacune de ces classes.

Les cartes auto-organisatrices de Kohonen [Koh82a][Koh82b][Koh97] sont parmi les réseaux neuronaux artificiels les plus utilisés dans le cadre de la classification non supervisée. Leur capacité d'accomplir la réduction de dimension ainsi qu'une organisation topologique, leur ont permis d'être employées dans de nombreux domaines d'applications. Une nouvelle application de ces cartes comme méthode de classification et d'analyse de données a été introduite. En effet, l'avantage des cartes auto-organisatrices de Kohonen comme méthode de classification et d'analyse de données par rapport aux méthodes traditionnelles de classification, c'est qu'elles peuvent englober des distributions de données plus générales, nécessitent moins de connaissances a priori et peuvent aussi être utilisées comme un outil général pour l'analyse des données multidimentionnelles et de grandes dimensions.

L'analyse de données symboliques s'inscrit dans le cadre d'une nouvelle approche numérique/symbolique du traitement des connaissances [KD91]. Cette nouvelle approche tente de rassembler l'analyse de données d'une part et l'apprentissage automatique d'autre part. Le rapprochement de ces deux disciplines est fondé sur la notion de données, sur des problèmes communs et sur une méthodologie commune [GG90]. **La deuxième partie de nos recherches est axée sur une nouvelle méthode de classification en analyse de données symboliques basée sur l'algorithme des cartes auto-organisatrices de Kohonen**. Pour cela, nous proposerons une adaptation des cartes topologiques de Kohonen aux données symboliques et plus généralement aux données ayant une structure complexe. En effet, les cartes topologiques de Kohonen sont basées sur la notion de centre de gravité et malheureusement ce concept n'est pas applicable aux données complexes. Notre but est de modifier l'algorithme des cartes topologiques afin de permettre son application aux mesures de dissimilarités. Cette approche permet un traitement aisé des différents types de données, car seule la définition d'une mesure de dissimilarité est nécessaire au déroulement de la méthode.

Plan de la thèse

Le chapitre 1 présente tout d'abord le formalisme et les méthodes de l'analyse de données classiques, nous détaillons celles utilisées dans nos recherches. On aborde alors les limites de l'analyse de données qui peuvent être franchies par l'analyse de données symboliques. C'est l'occasion d'introduire les concepts de base de l'analyse de données symboliques ainsi que les principales notions qui nous seront utiles dans le cadre de ce travail. Nous présentons le formalisme et les opérations élémentaires que l'on peut effectuer sur des descriptions symboliques. Nous finirons par présenter les travaux récents en analyse de données symboliques.

Le chapitre 2 définit le principe de sélection des informations à partir d'une base de données relationnelle. La définition d'un certain nombre d'opérateurs nous permet de construire une base de connaissances, décrite sous forme d'assertions, par généralisation des informations extraites de la base. En définissant les critères de qualité d'une description par rapport aux éléments qu'elle généralise, nous introduisons une méthode de spécialisation décrite dans [Ste98] permettant de traiter un des problèmes de cette généralisation.

Le chapitre 3 présente deux méthodes pour améliorer le procédé de généralisation, décrit au chapitre 2. La méthode de spécialisation proposée n'est pas toujours efficace. En effet, lorsque les données que l'on souhaite généraliser sont très hétérogènes, nous sommes amenés à les décomposer. Nous proposons donc une méthode de décomposition basée sur une méthode divisive de classification [Cha97]. La contrainte d'une telle décomposition est d'associer à chacun des groupes trouvés un objet symbolique généralisant les valeurs observées au sein du groupe. Nous proposons donc des modélisations permettant d'obtenir des descriptions. Nous nous intéressons dans un second temps à un cadre non supervisé de généralisation basé sur les méthodes de classification automatiques et plus

spécifiquement basé sur l'algorithme des cartes topologiques de Kohonen, connu pour sa capacité de réduction et de classification. Cette généralisation non supervisée sera complétée par la proposition de modélisations des groupes obtenus.

Dans le chapitre 4, nous proposons une extension de la version batch de l'algorithme des cartes topologiques, sur des tableaux de dissimilarités. On illustre par quelques applications l'intérêt de l'approche développée tant dans la classification classique que la classification symbolique.

Le chapitre 5 présente deux applications sur des données ayant une structure complexe. La première application porte sur les données fonctionnelles et la seconde sur les données Web.

En guise de conclusion, nous présentons l'intérêt et les limites de notre travail. Ces considérations inspirent les perspectives des travaux futurs dans la continuation de ceux présentés ici.

Première partie

Etat de l'art : Données symboliques et techniques d'extraction

Chapitre 1

Introduction aux données symboliques

1.1 Introduction

La première étape de l'analyse de données symboliques est un processus d'extraction des connaissances à partir des grands volumes de données contenues dans des bases de données. Face à ces grands volumes, une tâche de première importance est de réduire intelligemment les données en décrivant les concepts sous-jacents. Ces concepts sont modélisés par une structure plus complexe qu'on appelle *Objets Symboliques* [DK91]. Une nouvelle approche de traitement de ces données s'impose, c'est l'analyse de données symboliques [DK91]. L'objectif de l'analyse de données symboliques est d'étendre la problématique, les méthodes et les algorithmes de l'analyse de données aux données symboliques exprimant un niveau de connaissance plus élevé. Il peut s'agir de connaissances supplémentaires comme les dépendances entre certaines variables ou encore l'introduction de variables taxonomiques. Les descriptions symboliques mesurent l'imprécision, l'incertitude ou la variation des ensembles de données. Les données dans les tableaux de données symboliques sont décrites par un ensemble de valeurs ou par une distribution sur un ensemble de valeurs. On dira que chaque case du tableau contient une description symbolique.

Il y a deux grandes étapes dans l'analyse de données symboliques : une première étape qui consiste à extraire et à construire des données symboliques à partir de larges bases de données, en définissant un ensemble d'opérateurs adéquats. Une seconde étape

consiste à appliquer de nouveaux outils d'extraction de connaissances. L'analyse de données symboliques offre une nouvelle approche permettant donc le traitement des données complexes. Un de ses avantages est qu'elle offre aussi une interprétation facile des résultats pour les utilisateurs.

Dans un premier temps, ce chapitre présente brièvement l'analyse de données ainsi que certaines limites de son formalisme, ce qui permet de comprendre les motivations de l'analyse de données symboliques.

1.2 L'analyse de données classique

L'analyse de données est un ensemble de méthodes et d'outils statistiques qui permettent de recueillir, traiter et interpréter un tableau de données. Ce dernier correspond à des entités statistiques comparables et décrites par les mêmes variables. Ces entités appartiennent à une population ou à un échantillon. Deux aspects composent la démarche statistique de l'analyse de données :

- la statistique descriptive ou exploratoire, ayant pour but de synthétiser, structurer, visualiser l'information présente dans les données ;
- la statistique décisionnelle ou explicative, dont le but est d'étendre à la population les propriétés caractérisant un échantillon connu, ou encore de tester des hypothèses sur les propriétés des données.

Les méthodes explicatives permettent de répondre aux questions que se posent les utilisateurs sur les données récoltées et sont donc généralement employées après les méthodes exploratoires qui permettent de mieux cerner ces questions.

L'analyse de données vise à mettre en relief les relations entre les individus, entre les variables et entre les individus et les variables. Une des actions est de dégager une typologie des individus et une typologie des variables, puis de mesurer les relations entre ces deux typologies.

1.2.1 Contexte

Soit Ω l'ensemble des individus supposé fini et $\mathcal{O}$ un ensemble d'arrivée. Les individus sont décrits par des variables. Une variable sur $\mathcal{O}$ est définie par une application Y de

Ω dans $\mathcal{O}$, où $\mathcal{O}$ est muni d'une structure algébrique S. Suivant cette structure S et le cardinal de $\mathcal{O}$, on distingue deux grands types de variables : les variables quantitatives et les variables qualitatives

- les variables quantitatives

  Pour ces variables l'ensemble d'arrivée est $\mathbb{R}$. Dans la pratique on distingue :

  - quantitatif mesurable (poids, revenu,...)
  - quantitatif d'ordre (note, rang,...)
  - quantitatif de comptage (fréquence,...)
  - quantitatif binaire (succès/échec, présence/absence,...)

- les variables qualitatives

  L'ensemble d'arrivée $\mathcal{O}$ est fini. Les éléments de $\mathcal{O}$ sont appelés modalités de la variable. On distingue essentiellement les types suivants :

  - qualitatif nominal (lieu d'habitation, catégorie socio-professionnelle,...), on ne considère que la structure d'ensemble, la variable est définie par une relation d'équivalence sur Ω.
  - qualitatif ordinal (faible, moyen, fort,...), $\mathcal{O}$ est muni d'une structure d'ordre total.
  - qualitatif textuel (nom d'auteur, nom de film,...)

Les tableaux de données constituent les entrées des différentes méthodes d'analyse. Le cas le plus général est celui des tableaux individus $\times$ variables, avec les individus en lignes et les variables en colonnes. Selon le type des variables utilisées, on distingue différents types de tableaux : les tableaux quantitatifs, les tableaux qualitatifs, les tableaux binaires et les tableaux hétérogènes constitués de variables de types différents. Cependant d'autres types de tableaux peuvent être étudiés :

- les tableaux de dissimilarités entre individus (tableaux individus $\times$ individus) ;
- les tableaux de contingence obtenus à partir des tableaux individus $\times$ variables, en croisant les modalités de deux variables nominales et en comptant pour chaque paire de modalités le nombre d'individus présentant leur cooccurrence ;
- les tableaux de fréquences (tableaux variables $\times$ variables) : ils sont obtenus en rapportant les tableaux de contingence à leur marge totale ou aux marges en lignes

ou en colonnes.

1.2.2 Les méthodes de l'analyse de données

Les méthodes de la statistique exploratoire sont la visualisation et les méthodes de classification. Les méthodes de la statistique décisionnelle sont les méthodes de discrimination et les méthodes de régression.

1.2.2.1 Les méthodes de visualisation

Ces méthodes, basées sur l'algèbre linéaire, ont pour but la visualisation du nuage des observations. Cette visualisation se fait sur un espace de dimension réduite choisi de sorte à minimiser la déformation du nuage de points sur cet espace. Cela revient à chercher un petit nombre de variables synthétiques qui résument au mieux l'ensemble des variables et qui engendre un espace de projection conservant aux données le maximum de variation. Les méthodes sont différentes selon le type des tableaux utilisés :

- l'Analyse en Composante Principale (ACP), traite les tableaux de données quantitatives ;
- l'Analyse Factorielle d'un Tableau de Distances (AFTD), dont le but est de représenter les points à partir de leurs distances ;
- l'Analyse Factorielle des Correspondances (AFC), traite les tableaux de contingence et de fréquences ;
- l'Analyse des Correspondances Multiples (ACM), traite les tableaux de données qualitatives.

1.2.2.2 Les méthodes de classification

Les méthodes de classification visent à mettre en évidence une typologie des individus autrement dit une structuration des individus en classes homogènes. En intelligence artificielle, la classification automatique est considérée comme un procédé permettant à l'ordinateur de découvrir une information d'ordre sémantique qui n'était pas dans le tableau initial sous forme claire : on parle alors d'apprentissage sans professeur ou non supervisé. Plusieurs types de méthodes existent selon la structure classificatoire recherchée : partition, hiérarchie ou recouvrement. Toutes ces méthodes produisent en sortie des

regroupements d'individus homogènes qu'on appelle **classes**. La caractérisation d'une classe peut être constituée par l'énumération de ses éléments ou par une description représentant l'ensemble de ses éléments. En analyse de données, cette description s'appuie sur des indicateurs statistiques tels que les indicateurs centraux, de dispersion, de distribution, etc.

1.2.2.3 Les méthodes de discrimination et les méthodes de régression

Sous le nom d'analyse discriminante, on désigne toute une série de méthodes explicatives, descriptives et surtout prédictives destinées à étudier une population comportant k classes d'individus [DLPT82]. Chaque individu est caractérisé par un ensemble de q variables et une variable qualitative identifiant la classe à laquelle appartient cet individu. Quand la variable à expliquer est quantitative, on utilise les méthodes de regression. En fait, ces méthodes cherchent à expliquer les valeurs prises par les individus sur une variable, dite variable à expliquer, à partir des valeurs prises sur d'autres variables, dites variables explicatives.

1.2.3 Les méthodes utilisées

Dans cette section nous exposons les méthodes de l'analyse de données utilisées dans la suite de notre travail.

1.2.3.1 Analyse factorielle sur tableaux de distances

L'analyse en composantes principales est une technique de représentation d'un nuage de n points de l'espace $\mathbb{R}^p$, définis par leurs coordonnées sur les p axes, sur un sous espace de faible dimension $\mathbb{R}^q$, avec q $\leq$ p. Dans le cas où les données de départ sont les $\frac{n(n-1)}{2}$ distances ou dissimilarités entre individus, et non les variables les décrivant, on parle alors d'analyse factorielle sur tableaux de distances ou de dissimilarités. Le cas où l'on dispose d'une véritable distance euclidienne entre individus n'est qu'une version de l'ACP, le cas de dissimilarités conduit à des techniques originales [Sap90].

Soit Δ le tableau $n \times n$ des carrés des distances entre points :

$$d_{ij}^2 = d_{ji}^2 \quad et \quad d_{ii} = 0$$

Si d est euclidienne, chaque individu i, peut être représenté dans un espace de dimension q par un point e_i, tel que :

$$d_{ij}^2 = (e_i - e_j)'(e_i - e_j) \quad \text{où} \quad (e_i - e_j)' \quad \text{est le vecteur transposé}$$

Si on place l'origine au centre de gravité, les produits scalaires $w_{ij} = e_i' e_j$ sont alors entièrement déterminés par les d_{ij}^2.

Supposons $p_i = 1/n$ le poids de l'individu i, $\forall\ i \in \Omega$ et posons

$$d_{i.}^2 = \frac{1}{n}\sum_{j=1}^{n} d_{ij}^2 \ \text{ et } \ d_{..}^2 = \frac{1}{n}\sum_{i=1}^{n} d_{i.}^2 = 2I$$

I étant l'inertie.

On a alors la formule de Torgerson :

$$w_{ij} = -\frac{1}{2}(d_{ij}^2 - d_{i.}^2 - d_{.j}^2 + d_{..}^2)$$

En effet :

$$d_{ij}^2 = e_i' e_i + e_j' e_j - 2w_{ij} \quad \text{soit} \quad w_{ij} = \frac{1}{2}(-d_{ij}^2 + e_i' e_i + e_j' e_j)$$

d'où :

$$d_{i.}^2 = e_i' e_i + \frac{1}{n}\sum_j e_j' e_j \text{ car } \sum_j w_{ij} = e_i'(\sum_j e_j) = \mathbf{0}$$

car l'origine est au centre de gravité.

On a donc $d_{i.}^2 = e_i' e_i + I$ et de même $d_{.j}^2 = e_j' e_j + I$, d'où la formule par substitution. Matriciellement $W = -\frac{1}{2}\mathbf{A}\ \Delta\ \mathbf{A}$ où $\mathbf{A}$ est l'opérateur de centrage $\mathbf{A}=\mathbf{I} - \frac{\mathbf{11'}}{n}$: il y a donc double centrage en lignes et en colonnes de Δ, $\mathbf{I}$ est la matrice identité, $\mathbf{1}$ le vecteur de $\mathbb{R}^n$ dont toutes les composantes sont égales à 1.

On sait que les vecteurs propres de $\mathbf{WD}$, avec $\mathbf{D}$ la matrice diagonale des poids (ici $\frac{1}{n}\mathbf{W}$) sont les composantes principales du nuage des n points.

Connaissant uniquement les distances d_{ij}, on peut donc calculer les composantes principales et faire une représentation euclidienne de l'ensemble des points dans un espace de dimension fixée, car les composantes principales ne sont autres que des listes de coordonnées sur une base orthogonale. La dimension de l'espace est alors égale au rang de $\mathbf{W}$. Si $\mathbf{W}$ n'est pas positive, il n'existe pas de représentation euclidienne respectant les distances. On pourra obtenir une représentation approchée en se limitant au sous espace engendré par les vecteurs propres associés aux valeurs propres positives.

Une transformation permettant de passer d'une distance non euclidienne à une distance euclidienne [Sap90]

Si d n'est pas euclidienne, ce qui se produit quand $\mathbf{W}$ a des valeurs propres négatives, la méthode de la constante additive permet d'en déduire une distance euclidienne. Il existe en effet une constante c^2, telle que la distance δ_{ij}^2 définie par : $\delta_{ij}^2 = d_{ij}^2 + c^2$ avec $\delta_{ii} = 0$, soit euclidienne.

$$\mathbf{W}_\delta = \mathbf{W}_d + \mathbf{W}_c$$

$$\mathbf{W}_c = -\frac{1}{2}\mathbf{A}\begin{pmatrix} 0 & c^2 & c^2 & c^2 \\ c^2 & 0 & & \\ \vdots & \vdots & \vdots & \vdots \\ c^2 & & & 0 \end{pmatrix}\mathbf{A} = -\frac{1}{2}\mathbf{A}c^2\mathbf{(11'-I)A}$$

Comme $\mathbf{A}=\mathbf{I}-\frac{\mathbf{11'}}{n}$, $\mathbf{W}_c = -\frac{c^2\mathbf{A}}{2}((n-1)\mathbf{I}-n\mathbf{A})\mathbf{A} = -\frac{c^2}{2}((n-1)\mathbf{A}-n\mathbf{A})\mathbf{A} = \frac{c^2}{2}\mathbf{A}$, car $\mathbf{A}^2=\mathbf{A}$.

Les vecteurs propres associés à des valeurs propres non nulles de $\mathbf{W}_d$ sont centrés, comme $\mathbf{A}$ est l'opérateur de centrage, ils sont vecteurs propres de $\mathbf{W}_c$ avec pour valeur propre $c^2/2$.

Aux vecteurs propres de $\mathbf{W}_d$ correspondent les vecteurs propres de $\mathbf{W}_\delta$ avec pour valeurs propres $\lambda + c^2/2$. Il suffit donc de prendre $c^2 = 2|\lambda_n|$, où λ_n est la plus petite valeur propre de $\mathbf{W}_d$ (ici négative) pour que δ soit euclidienne.

Cette méthode permet aussi de transformer une dissimilarité directement en une distance euclidienne mais peut être au prix d'une déformation importante des données.

1.2.3.2 Les méthodes divisives de classification

Les méthodes divisives de classification sont des méthodes de classification hiérarchiques. Elles partent d'un ensemble d'individus Ω et procèdent par division successive des classes jusqu'à l'obtention de classes qui vérifient certaines règles d'arrêt. On les appelle les méthodes descendantes de classification hiérarchique pour les différencier des méthodes ascendantes qui partent des singletons et qui procèdent par agrégation. De plus, les complexités de ces deux familles de méthodes de classification hiérarchiques sont différentes. En effet, lors de la première étape d'une méthode ascendante, il faut

évaluer toutes les agrégations possibles de deux individus parmi les n individus, soit $n(n-1)/2$ possibilités, tandis qu'un algorithme descendant basé sur l'énumération complète évalue toutes les divisions de n individus en 2 sous ensembles non vides, soit $(2^{n-1}-1)$ possibilités. Plusieurs stratégies ont été proposées dans le cadre des méthodes divisives afin de réduire cette complexité [MS64] [CGK78]. Nous proposons dans notre travail (cf chapitre 3), une méthode divisive sur des données monovaluées, quantitatives ou qualitatives, qui réduit cette complexité à $(n-1)$ et qui pourra être utilisée au niveau de l'analyse de données symboliques pour la création d'objets symboliques à partir de bases de données relationnelles.

Définition 1.2.1 (Définition d'une hiérarchie). Soit Ω un ensemble d'individus, $\mathcal{H}$ une famille de classes, $\mathcal{H}$ est une hiérarchie si :

- $\Omega \in \mathcal{H}$
- $\forall$ w $\in \Omega$, {w} $\in \mathcal{H}$
- $\forall$ A, B $\in \mathcal{H}$, A $\cap$ B $\in$ {A, B, $\emptyset$}, c'est à dire que deux classes sont soit disjointes, soit contenues l'une dans l'autre.

Dans le domaine de l'analyse des données, les méthodes divisives de classification ont été principalement développées dans les années 70. Ces méthodes divisives sont itératives et procèdent à chaque itération au choix de la classe à diviser et au partitionnement de cette classe. Différentes stratégies de choix de classes à partager et de partitionnement de la classe choisie ont été proposées. Ces stratégies sont parfois arbitraires, parfois guidées par un souci d'optimisation. Ces méthodes utilisent des critères usuels en analyse de données telles que l'inertie ou le diamètre pour évaluer la qualité d'une partition.

Généralités

Les méthodes divisives, comme toutes les méthodes de l'analyse de données, peuvent s'appliquer à deux types de données, les tableaux individus-variables et les tableaux de dissimilarités.

L 'algorithme général d'une méthode divisive a la forme suivante :

Initialisation

$P_1 = \Omega$;

$k \leftarrow 1$;

Tant Que $k < n$ (n étant le nombre total d'individus) *alors* :

1. choisir $C \in P_k$
2. déterminer (C_1, C_2) une partition de C
3. $P_{k+1} = P_k \cup \{C_1, C_2\} - \{C\}$
4. $k \leftarrow k + 1$;

Fin Tant Que

Les différents algorithmes divisifs de classification se distinguent par :

– le choix de la classe C à diviser à chaque itération ;

– l'algorithme de partitionnement de la classe C en deux classes.

Plusieurs stratégies ont été développées pour résoudre ces deux problèmes majeurs des méthodes divisives de classification.

Le choix de la classe à diviser

À chaque étape k de l'algorithme divisif de classification, on a une partition P_k en k classes de Ω, $P_k = (C_1, C_2, ..., C_k)$ tel que $C_i \cap C_j = \emptyset$ et $\bigcup_{i=1,...,k} C_i = \Omega$. À chaque itération ayant une partition en k classes, on doit choisir une classe afin de la diviser en 2 sous-classes pour obtenir donc $(k+1)$ classes. Plusieurs stratégies de choix ont été développées :

– division de toutes les classes : c'est une stratégie particulièrement simple puisqu'elle consiste à ne pas faire de choix et à diviser toutes les classes à chaque étape ;

– division selon une caractéristique : c'est une stratégie qui consiste à diviser une classe à chaque itération en fonction d'une caractéristique définie arbitrairement, et ce dans le but d'indicer une hiérarchie et d'évaluer ainsi la hauteur des paliers dans l'arbre. Une caractéristique souvent utilisée est le diamètre. Rappelons que le diamètre d'une classe est la plus grande dissimilarité entre deux individus de cette classe. Les classes seront ainsi indicées par leurs diamètres ;

– division de la classe qui optimise la partition : c'est une stratégie qui consiste à choisir de diviser la classe qui donne la meilleure partition au sens d'un critère d'évaluation W. Soit $P_m = (C_1, C_2, ..., C_m)$, une partition en m classes de Ω. Si on divise une classe C_k en deux classes (C_k^1, C_k^2), on obtient la partition $P_{m+1} = (C_1, C_2, ..., C_{k-1}, C_k^1, C_k^2, C_{k+1}, ..., C_m)$. Le choix de la classe C_k à diviser

se fait en optimisant le critère $W(P_{m+1})$. On cherche donc, parmi toutes les partitions en $m+1$ classes, résultant de la division d'une classe, celle qui optimise le critère W.

Exemple 1.2.1. *si on se place dans le cas d'un critère d'évaluation W additif :*

$$W(P_m) = \sum_{k=1}^{m} Q(C_k) \tag{1.2.1}$$

où $Q(C_k)$ évalue la qualité de la classe C_k. Lorsqu'on choisit de diviser une classe C_k en 2 classes C_k^1 et C_k^2, on a :

$$W(P_{m+1}) = W(P_m) - Q(C_k) + Q(C_k^1) + Q(C_k^2) \tag{1.2.2}$$

Donc la partition P_{m+1} qui optimise $W(P_{m+1})$ s'obtient par division de la classe C_k qui optimise la variation du critère Q lorqu'on divise C_k en deux classes, c'est-à-dire la mesure suivante :

$$\Delta(C_k) = Q(C_k^1) + Q(C_k^2) - Q(C_k) \tag{1.2.3}$$

Donc pour choisir la classe à diviser, il faut définir les m partitions en deux classes, des m classes de P_m, puis retenir celle qui optimise $\Delta(C_k)$.

Le bipartitionnement d'une classe

Il existe plusieurs méthodes divisives de classification et différentes stratégies de partitionnement d'une classe C en sous-classes. Une approche naturelle est la division d'une classe C de n individus en deux sous ensembles non vides tout en minimisant le critère d'inertie intraclasse W.

$$W = \sum_{C_k} I(C_k)$$

Cette approche demande une considération de toutes les bipartitions possibles. L'énumération complète de toutes les bipartitions de n individus conduit à évaluer la qualité de $(2^{n-1}-1)$ bipartitions, ce qui n'est pas possible pour un n grand. Afin de réduire cette complexité, une approche proposée dans le cas des variables quantitatives, est de diviser la classe C selon une question binaire de la forme " $Y_i \leq c$? ", avec Y_i une variable quantitative et c la valeur de coupure. Si la classe C est composée de n individus, pour chaque variable quantitative Y_i, il y a $(n-1)$ bipartitions (C_1, C_2) différentes

induites par ce processus de division. En effet, quelque soit le point de coupure entre deux observations consécutives, la bipartition induite est la même. Ce choix de poser seulement $(n-1)$ questions par variable afin de générer toutes les bipartitions, nous a amené à choisir la valeur moyenne entre deux observations consécutives comme valeur de coupure. Si il y a p variables quantitatives, on choisira parmis les $p(n-1)$ bipartitions différentes, celle qui a la plus petite valeur d'inertie intraclasse. En revanche, pour une variable qualitative Y_i, il y a $(2^{mod-1}-1)$ bipartitions (C_1, C_2) différentes si mod désigne le nombre de modalités total pour la variable en question.

1.2.3.3 L'algorithme des cartes topologiques de Kohonen : *SOM*

Les cartes de Kohonen sont issues des travaux du Professeur T. Kohonen, de l'université de technologie d'Helsinki, qui a mis au point l'algorithme qui porte son nom [Koh97]. L'algorithme SOM (Self-Organizing Map), est utilisé de nos jours dans les domaines numériques, domaines dans lesquels il a fait ses preuves. C'est un outil très utilisé pour la visualisation de données multidimensionnelles. En effet, outre sa faculté à regrouper les données similaires au moyen de prototypes comme en quantification vectorielle et/ou en classification, il autorise la conservation de la topologie, d'où sa capacité à produire des représentations ordonnées, qu'on appelle prototypes, ou vecteurs référents, sur une carte. L'algorithme SOM est donc un algorithme d'auto-organisation qui projette l'espace des données sur un espace discret de faible dimension qu'on appelle carte, notée $L(C, W)$. La carte est constituée par un ensemble C de neurones interconnetés. Les cartes utilisées dans la pratique sont le plus souvent des treillis réguliers, dont chaque nœud est occupé par un neurone. À chaque neurone de la carte est associé un vecteur référent w_c de l'espace des données. W est donc l'ensemble de tous les vecteurs référents : $W = w_1, w_2, ..., w_m$.

L'apprentissage effectué par les cartes auto-organisatrices fait en sorte que ces vecteurs référents captent au mieux la densité de probabilité sous-jacente aux observations. Cet apprentissage introduit la conservation de la topologie et impose que deux neurones c et r, voisins par rapport à la topologie discrète de la carte, soient associés à deux vecteurs w_c et w_r, proches par rapport à la distance choisie sur les données.

L'algorithme considère en entrée un ensemble de n observations $X = z_1, z_2, ..., z_n \in \mathbb{R}^p$ et en sortie renvoie un réseau de m neurones. À chaque neurone c est associé un ensemble d'observations et un vecteur référent $w_c \in \mathbb{R}^p$.
L'espace de représentation, L_c, du neurone c est donc l'espace $\mathbb{R}^p$. L'espace de représentation de la partition L est la carte $L(C, W)$. Cet algorithme est compétitif : lors de la présentation d'un individu à la carte on lui associe le neurone le plus proche, c'est-à-dire le neurone dont le vecteur référent w_i est le plus proche au sens de la distance euclidienne du vecteur décrivant l'individu. L'algorithme utilise ainsi la distance euclidienne $d(z_i, w_c) = \|z_i - w_c\|$ entre une observation z_i et le vecteur référent $w_c \in \mathbb{R}^p$.

Une version stochastique et une version batch de l'algorithme existent :

La version Stochastique : l'algorithme des cartes topologiques est un algorithme itératif. Dans la version stochastique, à chaque itération une observation z_i de l'ensemble d'apprentissage est choisie aléatoirement et les distances entre cette observation et tous les neurones de la carte sont calculées. Le neurone c dont le vecteur référent est le plus proche de z_i au sens de la distance euclidienne est appelé neurone gagnant :

$$c = arg \min_{r=1,...,m} \|z_i - w_r\|$$

Après l'affectation de l'individu au neurone le plus proche, tous les vecteurs référents des neurones de la carte sont mis à jour selon la relation suivante de descente de gradient :

$$w_r^{t+1} = w_r^t + \alpha(t) K^T(\delta(c, r))(z_i - w_r^t)$$

où t est le temps, z_i est l'observation présentée au réseau au temps t, $K^T(\delta(c, r))$ est une fonction de voisinage autour du neurone gagnant c et $T = T(t)$ le rayon de voisinage qui décroît au cours du temps. $\alpha(t)$ désigne le pas d'apprentissage. Afin d'assurer la convergence de l'algorithme, il est nécessaire que la fonction fixant le pas d'apprentissage décroisse au cours du temps et satisfasse les conditions de l'approximation stochastique suivantes :

$$\sum_{t=0}^{\infty} \alpha(t) = \infty \quad et \quad \sum_{t=0}^{\infty} \alpha(t)^2 < \infty$$

La fonction de voisinage $K^T(\delta(c, r))$, notée aussi $K^T(\delta_{cr})$, est une fonction noyau positive et symétrique. Cette fonction permet d'introduire des zones d'influence autour

de chaque neurone et est fonction de la distance $\delta(c, r)$ entre le neurone gagnant c et le neurone r sur la carte. Cette distance, notée aussi δ_{cr}, est généralement euclidienne. Dans les premières formes de l'algorithme SOM, la fonction de voisinage utilisée était une fonction "bubble" définie de la manière suivante :

$$h_{cr}(t) = \begin{cases} \alpha(t) & \text{si } \delta_{cr} < T(t) \\ 0 & sinon \end{cases}$$

où $T(t)$ est le rayon de voisinage qui décroît aussi au cours du temps.

Plus tard, des fonctions graduellement décroissantes, telles que les fonctions Gaussiennes, furent proposées car elles accélèrent la convergence de l'algorithme. Un exemple de fonction couramment utilisée est la suivante :

$$K^T(\delta_{cr}) = e^{-\frac{\delta_{cr}^2}{2T(t)^2}}$$

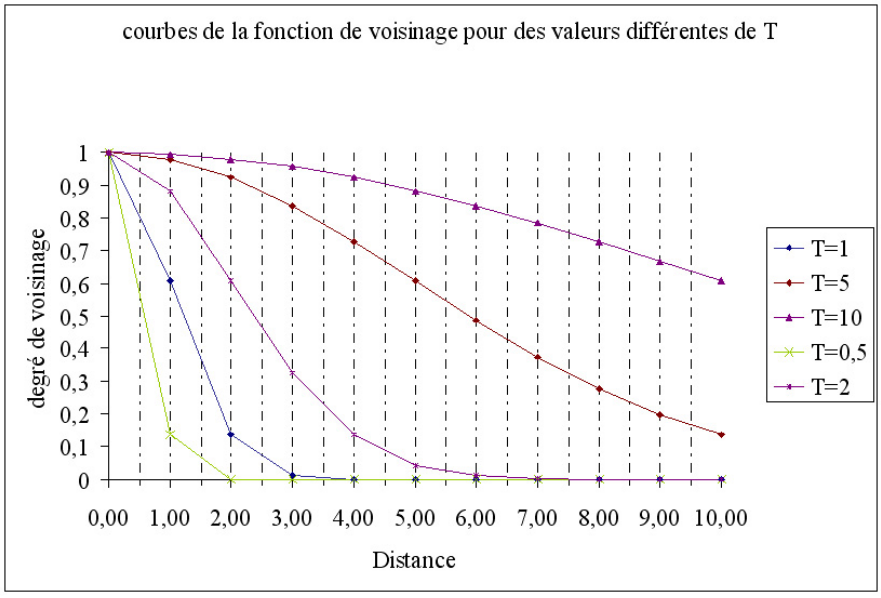


Figure 1.1 – Une famille de fonctions noyaux utilisées pour gérer le voisinage de la carte. Les différentes courbes représentent la fonction $K^T(\delta) = \exp^{-\frac{\delta^2}{2T^2}}$ pour différentes valeurs du paramètre T : du haut vers le bas, T prend les valeurs $10, 5, 2, 1, 0.5$

La formule de mise à jour des vecteurs référents est le résultat de l'approche stochastique reliée au critère de classification que l'on veut minimiser :

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{r\in C} K^T(\delta_{cr})\|z_i - w_r\|^2 \rightarrow \min_W$$

l'algorithme est le suivant :

Algorithme :

Initialisation :

▶ $t = 0$

▶ choisir une carte $L(C, W^0)$ initiale et un système de poids initial W^0

Itération : $t = t + 1$

▶ choisir une observation z_i parmi les n observations appartenant à l'ensemble d'apprentissage.

▶ déterminer le vecteur w_c vainqueur au sens :

$$c = arg \min_{r=1,\ldots,m} \|z_i - w_r\|$$

▶ déterminer la taille du voisinage $T(t)$ et le pas d'apprentissage $\alpha(t)$

▶ mettre à jour le système des vecteurs référents W selon la mise à jour de l'algorithme de gradient stochastique :

$$w_r^{t+1} = w_r^t + \alpha(t) K^T(\delta(c, r))(z_i - w_r^t)$$

Répeter Itération jusqu'à stabilisation

La première étape initialise les vecteurs référents associés aux neurones de la carte. La seconde étape consiste à sélectionner un individu au sein de l'ensemble d'apprentissage par tirage, individu que l'on présente directement au réseau. On fait alors entrer l'individu en compétition de façon à déterminer le neurone gagnant. Le neurone qui a remporté la compétition détermine le centre d'une zone de la carte appelée voisinage, zone dont l'étendue varie au cours du temps. La phase suivante, dite de mise à jour, modifie la position des vecteurs référents de façon à les rapprocher de l'individu présenté au réseau. Les neurones sont d'autant plus rapprochés de l'individu en question qu'ils sont proches sur la carte du neurone vainqueur.

Une version Batch de l'algorithme existe et consiste à mettre à jour les vecteurs référents après présentation de tous l'ensemble d'apprentissage.

La version Batch : de manière identique à l'algorithme des nuées dynamiques [DcG+89], la version batch des cartes de Kohonen [Koh97] [TLGC97] [Tea02] comporte deux phases distinctes : une phase d'affectation et une phase de représentation.

Lors de la phase d'affectation, on définit une fonction d'affectation f de l'espace des données vers la carte C, qui à tout élément z_i associe le neurone dont le vecteur référent est le plus proche de z_i au sens d'une distance généralisée notée d^T et qui fait intervenir tous les neurones de la carte :

$$d^T(z_i, w_{f(z_i)}) = \sum_{c \in C} K^T(\delta(c, f(z_i)))\|z_i - w_c\|^2$$

La fonction d'affectation, notée f_W, est définie comme suit :

$$f_W(z_i) = arg \min_{r \in C} d^T(z_i, w_r) \tag{1.2.4}$$

f_W induit donc une partition $P = \{P_c; c = 1, ..., m\}$ de l'ensemble des observations où chaque partie est définie par $P_c = \{z_i \in X; f(z_i) = c\}$.

Lors de la phase de représentation, l'algorithme met à jour les vecteurs référents de la carte, tout en minimisant une fonction coût, notée E, convenablement choisie. La fonction E mesure donc l'adéquation entre une partition P et une carte topologique $L(C, W)$ et a pour expression :

$$E(f, W) = \sum_{z_i} d^T(z_i, w_{f(z_i)}) \tag{1.2.5}$$

La fonction E étant convexe par rapport aux paramètres W, la minimisation est obtenue pour la valeur qui annule la dérivée :

$$\frac{\partial E}{\partial w_c} = \sum_{z_i} K^T(\delta(c, f(z_i)))(w_c - z_i) = 0$$

On choisit pour cela le système des référents W^* qui minimise E, la solution est unique [Tea02] et est donnée par :

$$w_c^* = \frac{\sum_{r \in C} K^T(\delta_{cr}) Z_r}{\sum_{r \in C} K^T(\delta_{cr}) n_r} \tag{1.2.6}$$

où $Z_r = \sum_{z_i/f(z_i)=r} z_i$, représente la somme de toutes les observations qui ont été affectées au neurone r et $n_r = card(r)$ représente le nombre d'observations affectées au neurone r.

L'algorithme est le suivant :

Initialisation :

► $t = 0$

► choisir une carte $L(C, W^0)$ avec W^0 un système de poids initial.

Itération : $t++$

À l'itération t l'ensemble des référents W^{t-1} de l'étape précédente est connu.

► **Phase d'affectation :** on affecte chaque observation z_i au neurone c défini par $f(z_i) = arg\min_c d^T(z_i, w_c)$

► **Phase d'optimisation :** déterminer le nouveau système des poids W^{t*}. Pour chaque neurone c :

$$w_c^* = \frac{\sum_{r \in C} K^T(\delta_{cr}) Z_r}{\sum_{r \in C} K^T(\delta_{cr}) n_r}$$

Répeter Itération jusqu'à stabilisation

Cet algorithme permet donc de construire une suite de W^0, W^1,..., W^t.

À l'étape t, $f_{W^{t-1}}$ est fixée, la phase d'optimisation détermine le nouveau système de poids W^t, minimum unique de $E(f_{W^{t-1}}, W)$, donné par la relation (1.2.6). On a donc :

$$E(f_{W^{t-1}}, W^t) \leq E(f_{W^{t-1}}, W^{t-1}) \tag{1.2.7}$$

La fonction d'affectation f_{W^t} associée à W^t est définie par la relation (1.2.4), elle permet d'obtenir les inégalités suivantes :

$$\forall z_i \in X \; d^T(z_i, f_{W^t}(z_i)) \leq d^T(z_i, f_{W^{t-1}}(z_i))$$

Soit en utilisant la définition (1.2.5) de $E(f, W)$:

$$E(f_{W^t}, W^t) \leq E(f_{W^{t-1}}, W^t) \tag{1.2.8}$$

Des inégalités (1.2.7) et (1.2.8) nous tirons la double inégalité :

$$E(f_{W^t}, W^t) \leq E(f_{W^{t-1}}, W^t) \leq E(f_{W^{t-1}}, W^{t-1}) \tag{1.2.9}$$

La suite $E_t = E(f_{W^t}, W^t)$ étant décroissante et minorée par zéro, est donc convergente. L'espace des partitions sur l'ensemble d'apprentissage étant fini, le nombre de fonctions d'affectation possibles f_{W^t} est fini, la suite E_t ne peut prendre qu'un nombre fini de valeurs, elle est donc stationnaire. La stationnarité de la suite E_t est effective dès que deux termes consécutifs sont égaux. En effet, si à l'itération t : $E_{t-1} = E_t$, d'après (1.2.9) nous avons l'égalité $E(f_{W^{t-1}}, W^t) = E(f_{W^{t-1}}, W^{t-1})$ dont la solution unique est $W^{t-1} = W^t$ et l'itération t ne modifie ni le système des poids, ni la fonction d'affectation obtenue à l'itération $t-1$. Donc comme pour le cas des nuées dynamiques [DcG+89], on vient de montrer que l'algorithme fait décroître le critère à chaque itération jusqu'à convergence [TLGC97].

Remarque 1.2.1. *Souvent le lien entre les neurones se fait par l'intermédiaire d'une structure de graphe non orienté* (C, Γ). *Cette structure de graphe induit une distance discrète* δ *sur la carte : pour tout couple de neurones* (c, r) *de la carte,* $\delta(c, r)$, *notée* δ_{cr}, *est définie comme étant la longueur du plus court chemin entre* c *et* r *(voir figure 1.2).*

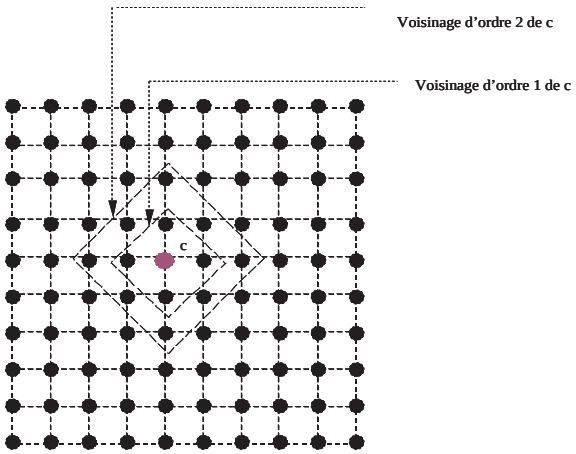


Figure 1.2 – Représentation de la topologie discrète d'une carte topologique à deux dimensions constituée de m=10*10 neurones : chaque point représente un neurone c

1.3 Les motivations des données symboliques

Le volume des données explose : des milliards d'informations sont collectées chaque jour par les organismes. Afin d'améliorer leurs positions concurrentielles, les entreprises d'aujourd'hui visent non seulement à améliorer leur productivité mais aussi à améliorer leur stratégie. Pour cela, il est nécessaire de disposer d'une architecture technique spéci-

fique, capable d'affronter le défi de l'ampleur des volumes (stockage) et des traitements (analyse). En effet, alors que les bases de données sont supposées améliorer la prise de décision, presque tous les progrès technologiques et les concepts d'organisation des bases de données sont concentrés sur la résolution de problèmes transactionnels. Si les nouvelles bases de données permettent de stocker des volumes d'informations toujours plus importants à des coûts de plus en plus faibles, force est de constater que les méthodes d'analyse classiques ne sont plus adaptées pour le traitement des informations qui deviennent de plus en plus riches et complexes. En effet, la représentation tabulaire classique des données a un pouvoir expressif limité, qui ne permet pas de prendre en compte des informations complexes, comme par exemple la variation propre aux données, ou encore des relations de dépendance logique entre variables. De plus, la représentation distincte des individus et des classes dans l'analyse classique, ne permet pas de réutiliser ces dernières et se placer à un niveau d'abstraction supérieur. L'intérêt du choix des données symboliques pour modéliser ce nombre important d'informations est :

- de prendre en compte dans la synthèse, la notion de variabilité intervenant au sein d'un groupe de données. Les résumés obtenus ne se limitent plus aux indicateurs centraux calculés sur les observations ;
- de permettre des analyses ultérieures sur les résumés obtenus, telles que les méthodes de visualisation, les méthodes de classification, etc.

1.4 L'analyse de données symboliques

En analyse de données symboliques, les données sont décrites dans un tableau où chaque case peut contenir non seulement une valeur qualitative ou quantitative unique mais également un ensemble de valeurs, un intervalle, une distribution sur un ensemble de valeur, une loi de probabilité, une fonction ou une règle de dépendance. On dira alors que chaque case du tableau contient une *description symbolique*. L'analyse de telles données nécessite donc l'adaptation de l'analyse de données à l'analyse de données symboliques.

1.4.1 Le formalisme des données symboliques

La définition de variable a été étendue afin de pouvoir décrire un individu sur une variable par plusieurs valeurs du domaine d'observation $\mathcal{O}$, domaine de valeurs élémentaires. On modifie alors le domaine d'arrivée d'une variable symbolique par rapport à celui d'une variable classique.

Une variable symbolique Y est alors définie par une application :

$$Y : \Omega \rightarrow D$$

$$w \rightarrow Y(w) = d$$

avec :

- Ω : l'ensemble des individus. On fera ici la différence entre un individu et une classe d'individus ; les classes appartiennent à $P(\Omega)$, ensemble des parties de Ω.
- D : l'ensemble des descriptions d'individus, aussi bien que de classe d'individus.
- S : l'ensemble des objets symboliques. Un objet symbolique s constitue l'intension d'une classe, il se définit comme étant une description d_s munie d'une fonction a de calcul d'extension. Cette fonction fait appel à un opérateur de comparaison R entre descriptions. L'extension d'un objet symbolique est formée par l'ensemble des individus répondant bien à sa description.

L'objet symbolique s'axe autour des notions clés d'intension et d'extension. Ce formalisme reprend la distinction entre compréhension et étendue d'une idée d'Arnauld et Nicole (1662) cité dans [Did98]. Au lieu de définir l'objet symbolique comme un couple (intension, extension), on définit celui-ci comme un couple (intension, moyen de calcul de l'extension), ce qui peut traduire l'hypothèse cognitive selon laquelle notre cerveau ne contient pas nécessairement tous les exemplaires d'une classe mais un moyen de les reconnaître.

1.4.1.1 Le domaine de description symbolique

Le domaine de description D associé à Ω est construit sur un ensemble de variables Y= y_1, ..., y_p, et peut s'écrire D= $D_1 \times ... \times D_p$, où chaque variable est une application

de la forme $y_j : \Omega \longrightarrow D_j$. La description sur D de tout individu w_i avec $i = 1, ..., n$ se note : $Y(w_i)=(y_1(w_i),..., y_p(w_i))$, où chaque $y_j(w_i)$ désigne la description associée à w_i sur D_j avec $j = \{1, ...p\}$. Tout domaine D_j est un domaine de valeurs symboliques, qui peut s'écrire à partir d'un domaine $\mathcal{O}_j$ de valeurs élémentaires, lequel est de type qualitatif ou quantitatif. Trois types de domaine sont considérés :

- $D_j = \mathcal{O}_j$. On parle de *description univaluée*, binaire, nominale simple ou continue simple. Par exemple $d_w = bleu$.
- $D_j = P(\mathcal{O}_j)$, l'ensemble de parties de $\mathcal{O}$. On parlera alors de *description multi-valuée* nominale multiple ou continue intervalle. Par exemple $d_w = \{bleu, vert\}$.
- $D_j = [0, 1]^m$ avec $m = card(\mathcal{O}_j)$, l'ensemble des fonctions de $\mathcal{O}_j$ dans $[0, 1]$. On parlera alors de *description modale*. Par exemple, d_w peut être une distribution de probabilité sur $\mathcal{O}_j$.

D'une manière générale, on parle d'une *description symbolique* d'une variable et on distingue le domaine d'observation $\mathcal{O}_j$ du domaine d'arrivée D_j de la variable. Les valeurs du domaine d'observation $\mathcal{O}_j$ seront considérées comme des descriptions élémentaires ou encore des valeurs élémentaires à partir desquelles seront construites les descriptions dites symboliques de D_j.

1.4.1.2 Les objets symboliques

Un objet symbolique s est généralement défini par un triplet (a, R, d), où d est une description sur D, R un opérateur de comparaison sur D et a une fonction d'appartenance fonction de d et R, permettant de comparer la description avec une autre description :

$$a : \Omega \rightarrow \mathcal{L}$$

$$w \rightarrow a(w)$$

qui évalue le degré d'appartenance d'un individu w à l'extension de s.

Un objet symbolique modélise un concept ou empiriquement une classe. L'*intension* d'un objet symbolique est l'ensemble des propriétés décrivant le concept ou la classe. Ces propriétés sont des conditions nécessaires et suffisantes pour qu'un individu les satisfaisant appartienne au concept ou à la classe. Un objet symbolique possède aussi

une *extension*, qui est la liste de ses instances. Une instance d'un objet symbolique est un individu satisfaisant son intension.

On note $[d'Rd] \in \mathcal{L}$, le résultat de la comparaison entre deux descriptions d' et d par R, avec $\mathcal{L} = \{Vrai, Faux\} = \{0,1\}$ ou $\mathcal{L} = [0,1]$. Si $\mathcal{L} = \{0,1\}$ on parle d'*objets symboliques booléens*. Si $\mathcal{L} = [0,1]$ on parle d'*objets symboliques modaux*.

1.4.1.3 Différents types d'expressions symboliques

On définit plusieurs formes particulières d'expressions symboliques. Nous commençons par définir l'unité de base de toute expression symbolique :

Définition d'assertion élémentaire : Une assertion élémentaire est l'unité de base de toute expression symbolique. Elle se rapporte à un ensemble d'individus Ω (muni d'un domaine de description D) et s'écrit $e_j = [Y_j R_j d_j]$, où Y_j est de domaine D_j, d_j désigne une description de domaine D_j enfin R_j est un opérateur de comparaison entre Y_j et d_j.

- une assertion élémentaire booléenne est un cas particulier d'assertion élémentaire où D_j est de type univalué ou multi-valué et R_j est l'opérateur d'appartenance ($\in$) ou d'inclusion ($\subseteq$) selon le type de D_j ;
- une assertion élémentaire probabiliste est un cas particulier d'assertion élémentaire où D_j est de type modal.

Définition d'un objet assertion : Une assertion est un objet symbolique s=(a, R, d), qui se rapporte à un ensemble d'individus. L'application a est exprimée sous la forme d'une conjonction d'assertions élémentaires : $a = \bigwedge_{j=1,\ldots,p} [Y_j R_j d_j]$, ou encore : $a = \bigwedge_{j=1,\ldots,p} e_j$. Dans une assertion, on considère que toutes les assertions élémentaires portent sur un seul et même individu sur Ω. On peut donc avoir la notation $a(w) = \bigwedge_{j=1,\ldots,p} [Y_j(w) R_j d_j]$.

La quantité $a(w)$ représente donc le *degré d'appartenance* d'un individu w à l'extension de s.

Une assertion repose donc sur le choix d'une description de D et d'une fonction de

calcul d'appartenance a, définie à partir de fonctions f et h :

$$a : \Omega \to \mathcal{L}$$

$$w \mapsto a(w) = f(h_1(d_1, Y_1(w)), ..., h_p(d_p, Y_p(w)))$$

1. Définitions des fonctions f et h :

   Soit $\mathcal{L}$, un ensemble de valeurs de vérité. En particulier, $\mathcal{L} = \{0, 1\}$ ou $\mathcal{L} = [0, 1]$.

   On considère la famille de fonctions h :

   $$h_j : D_j \times D'_j \to \mathcal{L}$$

   $$(d_j, d'_j) \quad \mapsto h_j(d_j, d'_j)$$

   La fonction h_j mesure l'adéquation d'une description $d'_j \in D'_j$ à une description $d_j \in D_j$. Suivant la nature de D'_j et D_j nous donnerons par la suite quelques exemples de fonctions de comparaison (voir 1.5.1).

   On considère la fonction d'agrégation f :

   $$f_j : \mathcal{L} \times ... \times \mathcal{L} \to \mathcal{L}$$

   $$(x_1, ..., x_p) \qquad \mapsto x$$

   la fonction qui agrège les valeurs de vérité trouvées pour chaque comparaison effectuée sur (D_j, D'_j), $j = \{1, ..., p\}$. Nous donnerons par la suite quelques exemples de fonctions d'agrégations (voir 1.5.2).

   Ces deux fonctions sont prises en compte dans la définition de la fonction d'adéquation, a_s, d'une assertion s.

2. Assertions booléennes

   Définition 1.4.1. une assertion s est dite booléenne, si a est une application à valeurs dans {0,1} :

   $$a : \Omega \to \{0, 1\}$$

   $$w \mapsto a(w)$$

   Définition 1.4.2. L'extension d'une assertion booléenne s est égale à :

   $$ext_w(s) = \{w \in \Omega | a(w) = 1\}$$

soit $a = \bigwedge_j [Y_j \in d_j]$ et w un individu de description $\delta = (\delta_1, ..., \delta_j, ..., \delta_p)$. Dans le cas booléen, l'application a est définie comme :

$$a(w) = \prod_{j=1}^{p} h_j(d_j, \delta_j) \quad où \quad h_j(d_j, \delta_j) = \begin{cases} 1 & si \quad \delta_j \in d_j \\ 0 & sinon \end{cases}$$

3. Assertions modales

Définition 1.4.3. une assertion s est dite modale, si a est une application définie à valeurs dans [0,1] :

$$a : \Omega \rightarrow [0, 1]$$

$$w \mapsto a(w) = \prod_{j=1}^{p} h(d_j, Y_j(w))$$

Définition 1.4.4. L'extension d'une assertion modale s peut être définie de deux manières différentes :

- on considère tout d'abord que tout individu $w \in \Omega$ peut appartenir "plus ou moins" à l'extension de s, en fonction de son degré d'appartenance $a(w)$:

$$ext_\Omega(s) = \{(\omega, a(\omega)) | \omega \in \Omega\}$$

- dans une deuxième approche, on peut considérer que l'appartenance d'un individu ω à l'extension de s est admise si la quantité $a(\omega)$ est au moins égale à un seuil α fixé :

$$ext_\Omega(s) = \{\omega \in \Omega | a(w) \geq \alpha\}$$

Définition d'un objet Horde : on peut voir une horde [Did98] comme une conjonction d'assertions portant chacune sur un individu de Ω. On la note $h(\omega_1, ..., \omega_n) = \bigwedge_{i=1,n} s_i(\omega_i) = (a, R, d)$, avec $a : \Omega^n \longrightarrow \mathcal{L}$ et $d \in D^n$, où D^n est l'ensemble des éléments de Ω^n comprenant n éléments distincts de Ω.

L'extension d'une horde h, notée $ext(h)$, est l'ensemble des éléments $\mathcal{L}_j = (\omega_{1j}, ...\omega_{nj})$ de Ω^n tels que $\forall i = 1, ...n \;\; \omega_{ij} \in ext(s_i)$

1.4.2 Propriétés des données symboliques

On reprend ci-dessous les définitions de diverses propriétés concernant les objets symboliques, données dans [Did89] (sauf quand une autre référence est précisée).

1.4.2.1 Relations et opérations entre objets symboliques

Ordre symbolique : Soit r une relation d'ordre sur D, et soit A l'ensemble des objets symboliques assertions, l'ordre symbolique $\leq_A$ sur A est défini par : a, b $\in$ A, a $\leq_A$ b $\iff ext_\Omega(a) \subseteq ext_\Omega(b)$ et $d_a r d_b$. Il s'agit d'un pré-ordre partiel sur les descriptions et les extensions. Si a $\leq_A$ b, on dit que a hérite de b, que b est plus général que a, que b est un ascendant de a et que a est un descendant de b.

Relation d'équivalence : La relation d'équivalence $=_A$ entre objets symboliques assertions est définie par : a, b $\in$ A, a$=_A$ b $\iff ext_\Omega(a) = ext_\Omega(b)$

Union symbolique : Soient deux objets symboliques s_1, $s_2 \in$ S, l'union symbolique $s_1 \cup_s s_2$ est la conjonction de tous les objets symboliques de S dont l'extension contient l'ensemble des éléments de $ext(s_1) \cup ext(s_2)$.

Intersection symbolique : Soient deux objets symboliques s_1, $s_2 \in$ S, l'intersection symbolique $s_1 \cap_s s_2$ est la conjonction de tous les objets symboliques de S dont l'extension contient l'ensemble des éléments de $ext(s_1) \cap ext(s_2)$.

1.4.2.2 Critères de qualité des objets symboliques

Complétude : Un objet symbolique est complet quand il décrit toutes les propriétés de son extension (un objet complet est la partie intentionnelle d'un concept).

Simplicité : Un objet symbolique s est d'autant plus simple que le nombre d'assertions élémentaires qui le décrivent est plus proche du plus petit nombre d'assertions élémentaires dont la conjonction a la même extension.

Affinement : Un objet symbolique s est d'autant plus affiné que les assertions élémentaires qui le définissent ont une extension proche de celle de s.

Potentiel de description : Le potentiel de description d'un objet symbolique s=(a, R, d) où d=(d_1,..., d_p), est le volume de l'hypercube défini par le produit cartésien des descriptions symboliques booléennes $d_1 \in D_1$,..., $d_p \in D_p$ [Car94].

Opérateur de complétude : Soient $P(\Omega)$ l'ensemble des parties de Ω et A l'ensemble des objets symboliques assertions, avec $\leq_A$ l'ordre symbolique sur A. Soient les applications g : A $\longrightarrow P(\Omega)$ et f : $P(\Omega) \longrightarrow$ A et leur composé h= fog : A $\longrightarrow$ A. h est un opérateur de complétude si il vérifie les propriétés suivantes :

- isotonie : $\forall$ a, b $\in$ A, a$\leq_A$ b $\rightarrow$ h(a) $\leq_A$ h(b)
- idempotence : $\forall$ a $\in$ A, h(h(a))$=_A$ h(a)

1.4.2.3 Critères de qualité des classes d'objets symboliques

Généralisance (stabilité) : Il s'agit de la capacité d'une classe à être représentée par l'objet symbolique de plus petite extension qui contient l'union des extensions des éléments de la classe. Elle peut être mesurée par l'écart entre l'union de l'extension et l'extension de l'union.

Effritement : C'est le plus petit nombre d'objets symboliques dont la réunion des extensions est contenue dans l'extension des éléments de la classe (avec le moins de débordement possible).

1.4.2.4 Critères de qualité des classifications d'objets symboliques

Une classification peut être complète, affinée, simple, avoir une bonne généralisance et un faible effritement suivant que ses classes ou leurs représentants ont les qualités suivantes :

Recouvrance : La recouvrance d'une classification est le degré de recouvrement des extensions des objets symboliques représentant les classes de la classification.

Héritance : L'héritance d'une classification est la qualité de l'héritage des classes entre elles, par exemple le nombre de classes en relation d'héritage.

1.4.3 Conclusion

Face à ce nouveau formalisme et donc cette extension d'ordre sémantique que l'analyse de données symboliques a apporté à l'analyse de données, une nouvelle approche

de représentation, de traitement et d'interprétation des données s'impose. Pour cela, nous présentons dans la suite de ce chapitre, les outils de l'analyse de données symboliques ainsi que quelques méthodes déjà élaborées dans le cadre de l'analyse de données symboliques.

1.5 Outils de l'analyse des données symboliques

Dans le cadre de l'analyse de données symboliques, on distingue habituellement trois types de comparaison :

1. comparaison entre deux descriptions classiques ;
2. comparaison entre deux descriptions symboliques de même ordre ;
3. comparaison de la description symbolique d'un individu à celle d'une classe.

Le premier cas est celui que l'on rencontre habituellement en analyse de données, celui où l'on compare des descriptions qui sont des conjonctions de valeurs atomiques sur chaque descripteur. On peut utiliser donc les indices usuels de similarité/dissimilarité sur les données qualitatives ou quantitatives.

Le deuxième cas porte sur la comparaison de descriptions symboliques de même ordre (relative à deux individus ou deux classes, pouvant être plus complexes que dans le cadre classique). Ici on travaille essentiellement sur les descriptions associées aux assertions symboliques.

Le troisième cas correspond à un jugement d'appariement. On compare ici une description symbolique relative à un individu et une description symbolique relative à une classe. Encore une fois, les objets symboliques impliqués ici sont essentiellement des assertions symboliques.

Dans ces différents cas, on parlera par abus de langage de comparaison d'objets symboliques, bien que seules les descriptions associées aux objets symboliques soient effectivement comparées.

1.5.1 Fonctions de comparaison entre descriptions : h

En analyse de données symboliques, mesurer la ressemblance entre deux individus revient à mesurer la ressemblance entre leurs *vecteurs de description*. Pour comparer deux

vecteurs de descriptions, on procède souvent par comparaison des descriptions variable par variable, puis par agrégation de ces comparaisons. Dans ce qui suit, on présente quelques mesures de dissimilarités qui s'intéressent plus particulièrement aux assertions symboliques.

1.5.1.1 Opérateurs de comparaison entre descriptions univaluées

Nous définissons dans ce qui suit, quelques fonctions de distance élémentaires de $D_j \times D_j$ dans $[0,1]$.

- cas binaire : $h(d_j, d'_j) = 0$ si $d_j = d'_j$, 1 sinon.
- cas nominal : $h(d_j, d'_j) = 0$ si $d_j = d'_j$, 1 sinon.
- cas continu :

$$h(d_j, d'_j) = |d_j - d'_j| / (Max\{d^i_j\}_{i=1...n} - Min\{d^i_j\}_{i=1...n})$$

1.5.1.2 Opérateurs de comparaison entre descriptions multi-valuées

Une description multi-valuée δ, est un ensemble de modalités ou un intervalle du domaine d'observation $\mathcal{O}$. Pour comparer deux descriptions multi-valuées δ_A et δ_B appartenant au même domaine de description $P(\mathcal{O})$, on définit une fonction $h : P(\mathcal{O}) \times P(\mathcal{O}) \longrightarrow \mathbb{R}^+$, qui peut être une similarité, une dissimilarité ou une distance.

Cas intervalle : Soit un ensemble $\mathcal{O}$ de n objets indicés par $k = 1, ..., n$ et décrits par p variables intervalles $Y_1, ..., Y_p$. Soit $(x_{kj})_{n \times p}$ la table de données symboliques. L'objet k est décrit par la $k^{ème}$ ligne :

$$x_k = (x_{k1}, ..., x_{kp}) = ([a_{k1}, b_{k1}], ..., [a_{kp}, b_{kp}])$$

qui correspond au rectangle $Q_k = [a_k, b_k] = [a_{k1}, b_{k1}] \times ... \times [a_{kp}, b_{kp}]$ dans l'espace euclidien $\mathbb{R}^p$. Il existe plusieurs méthodes de calcul de dissimilarités entre 2 rectangles $Q = [a, b]$ et $Q' = [a', b'] \in \mathbb{R}^p$. Nous allons définir certaines d'entre elles qui nous seront utiles par la suite :

1. ***La dissimilarité de Gowda et Diday [BD99a]***

$$d(Q, Q') = \sum_{j=1}^{p} h(Q_j, Q'_j) \tag{1.5.1}$$

$$h(Q_j, Q'_j) = h_\pi(Q_j, Q'_j) + h_s(Q_j, Q'_j) + h_c(Q_j, Q'_j)$$

- Le terme h_π, compare la position de deux intervalles :

$$h_\pi(Q_j, Q'_j) = \frac{|a_j - a'_j|}{|Y_j|}$$

où $|Y_j|$ est l'écart maximum de la variable considérée.

- Le terme h_s compare l'étendue de deux intervalles :

$$h_s(Q_j, Q'_j) = \frac{l_Q - l_{Q'}}{l_s}$$

où $l_Q = |b_j - a_j|$ et $l_{Q'} = |b'_j - a'_j|$ et $l_s = |max(b_j, b'_j) - min(a_j, a'_j)|$

- Finalement le terme h_c compare le contenu de deux intervalles :

$$h_c(Q_j, Q'_j) = \frac{l_Q + l_{Q'} - 2.|Q_j \cap Q'_j|}{l_s}$$

2. ***La distance euclidienne :***

La distance euclidienne entre les milieux $\mu_Q = (a+b)/2$ et $\mu_{Q'} = (a'+b')/2$ respectifs de Q et Q' :

- Le cas d'une dimension, c'est à dire $Q = [a, b]$ et $Q' = [a', b']$ sont deux intervalles de $\mathbb{R}^1$:

$$d^2(Q, Q') = |\mu_Q - \mu_{Q'}|^2 = 1/4|(a - a') + (b - b')|^2 \tag{1.5.2}$$

- Le cas p-dimensionnel :

$$d^2(Q, Q') = \|\mu_Q - \mu_{Q'}\|^2 = 1/4\|(a - a') + (b - b')\|^2 \tag{1.5.3}$$

3. ***La distance de type sommet :***

C'est la somme des distances euclidiennes des 2^p sommets : $u_\varepsilon = a + \varepsilon * (b - a)$ et $u'_\varepsilon = a' + \varepsilon * (b' - a')$ de Q et Q', avec $\varepsilon = (\varepsilon_1, ..., \varepsilon_p) \in \{0, 1\}^p$ est un p-vecteur de zéro et un, et $\varepsilon * u = (\varepsilon_1 u_1, ..., \varepsilon_p u_p)$:

- Le cas d'une dimension :

$$\begin{aligned} d^2(Q, Q') &= \sum_{\varepsilon \in \{0,1\}^1} |u_\varepsilon - u'_\varepsilon|^2 = \sum_{\varepsilon \in \{0,1\}^1} |a - a' + \varepsilon * (b - b' - a + a')|^2 \\ &= (|a - a'|^2 + |b - b'|^2) \end{aligned} \tag{1.5.4}$$

– Le cas p-dimensionnel :

$$d^2(Q,Q') = \sum_{\varepsilon\in\{0,1\}^p} \|u_\varepsilon - u'_\varepsilon\|^2 = \sum_{\varepsilon\in\{0,1\}^p} \|a - a' + \varepsilon * (b - b' - a + a')\|^2$$

$$= \sum_{\varepsilon\in\{0,1\}^p} \sum_{j=1}^{p} [(1-\varepsilon_j)(a_j - a'_j) + \varepsilon_j(b_j - b'_j)]^2$$

$$= 2^{p-1}(\|a - a'\|^2 + \|b - b'\|^2) \tag{1.5.5}$$

4. ***La distance de Hausdorff :***

La distance entre deux ensembles ou deux rectangles Q et Q' est définie par :

$$d_H(Q,Q') = \max\{\sup_{\alpha\in Q}\{\inf_{\beta\in Q'} d(\alpha,\beta)\}, \sup_{\beta\in Q'}\{\inf_{\alpha\in Q} d(\alpha,\beta)\}\} \tag{1.5.6}$$

avec $d(\alpha,\beta)$ est une distance entre deux éléments ou deux intervalles mesurée sur $\mathbb{R}^p$ [Aub94] [CCLV03]. Dans le cas particulier, de choix d'une distance euclidienne pour d, on a :

– Le cas d'une dimension, c'est-à-dire $Q = [a,b]$ et $Q' = [a',b']$ sont deux intervalles de $\mathbb{R}^1$:

$$d(Q,Q') = d_1([a,b],[a',b']) = \max\{|a - a'|, |b - b'|\} \tag{1.5.7}$$

– Le cas p-dimensionnel :

$$d(Q,Q') = d_p([a,b],[a',b']) = (\sum_{j=1}^{p} d_1([a_j,b_j],[a'_j,b'_j])^2)^{\frac{1}{2}}$$

$$= (\sum_{j=1}^{p} (\max\{|a_j - a'_j|, |b_j - b'_j|\})^2)^{\frac{1}{2}} \tag{1.5.8}$$

Cas nominal multiple et ordinal :

1. ***La dissimilarité de Gowda et Diday [BD99a]***

$$d(Q_j,Q'_j) = h_s(Q_j,Q'_j) + h_c(Q_j,Q'_j)$$

$$h_s(Q_j,Q'_j) = \frac{l_Q - l_{Q'}}{l_s}$$

où l_Q = nombre de catégories dans Q= $|Q|$ et $l_{Q'}$=nombre de catégories dans $Q' = |Q'|$ et $|Q_j \cap Q'_j|$ = nombre de catégories dans $Q_j \cap Q'_j$ et l_s= nombre de catégories dans $Q_j \cup Q'_j = l_Q + l_{Q'} - |Q \cap Q'|$

$$h_c(Q_j, Q'_j) = \frac{l_Q + l_{Q'} - 2.|Q_j \cap Q'_j|}{l_s}$$

2. ***La Fonction de comparaison de Ichino :*** on définit une fonction $h : P(\mathcal{O}) \times P(\mathcal{O}) \rightarrow \mathbb{R}^+$ telle que pour $\delta_A \in P(\mathcal{O})$ et $\delta_B \in P(\mathcal{O})$:

$$h(\delta_A, \delta_B) = |\delta_A \oplus \delta_B| - |\delta_A \cap \delta_B| + 1/2 \ (\ 2\ |\delta_A \cap \delta_B| - |\delta_A| - |\delta_B|) \qquad (1.5.9)$$

Avec | | est le cardinal quand $\mathcal{O}$ est discret et l'étendue de l'intervalle lorsque $\mathcal{O}$ est continu. $\oplus$ est l'union jointe qui est définie par :

- $\delta_A \oplus \delta_B = [min(a_i, b_i), max(a_s, b_s)]$ si δ_A et δ_B sont deux intervalles $[a_i, a_s]$ et $[b_i, b_s]$.
- $\delta_A \oplus \delta_B = \delta_A \cup \delta_B$ si δ_A et δ_B sont deux ensembles.

Remarque 1.5.1. *Si δ_A et δ_B sont deux intervalles avec $\delta_A \not\subset \delta_B$ et $\delta_B \not\subset \delta_A$, alors h mesure la distance entre les milieux de δ_A et δ_B*

1.5.1.3 Opérateurs de comparaison entre descriptions modales

Une description modale est soit une distribution de probabilité soit une fonction d'appartenance à un ensemble flou. Nous allons nous intéresser ici aux distributions de probabilité. Soit un ensemble $\mathcal{O}$ fini, deux distributions δ_A et δ_B qui associent un poids $\delta(x)$ à un élément x de $\mathcal{O}$, qui peut être une probabilité subjective fournie par un expert ou une fréquence associée à l'élément x. Pour comparer deux groupes, il ne sera pas possible d'utiliser la distance du χ^2, car elle permet plutôt de comparer deux lignes d'un tableau de fréquences totales. Les fréquences sont dites totales quand elles sont calculées sur le même ensemble d'individus. Or, dans notre cas, les fréquences sont "conditionnelles" car elles sont calculées sur deux groupes différents. Il existe cependant des mesures de distances entre distributions, dont la plus simple est la distance euclidienne dans le cas discret et la norme L_2 dans le cas continu :

$$d^2(\delta_A, \delta_B) = \sum_{x \in O} (\delta_A(x) - \delta_B(x))^2 \qquad (1.5.10)$$

$$d^2(\delta_A, \delta_B) = \int_R (\delta_A(x) - \delta_B(x))^2 dx \tag{1.5.11}$$

Une autre distance est la distance de Hellinger appelée aussi distance de Matusita [Mat51], [Mat55] :

$$d^2(\delta_A, \delta_B) = \sum_{x \in O} (\sqrt{\delta_A(x)} - \sqrt{\delta_B(x)})^2 \tag{1.5.12}$$

$$d^2(\delta_A, \delta_B) = \int_R (\sqrt{\delta_A(x)} - \sqrt{\delta_B(x)})^2 dx \tag{1.5.13}$$

Une autre distance appelée la **distance d'affinité** [BN85], [BN00] (qui sera utilisée dans une des applications au chapitre 5) :

$$d^2(\delta_A, \delta_B) = 2 * (1 - a(\delta_A, \delta_B)) \tag{1.5.14}$$

avec a est le coefficient d'affinité défini comme suit :

$$a(\delta_A, \delta_B) = \sum_{l \in O} \sqrt{\delta_A(l) \times \delta_B(l)}$$

1.5.2 Opérateurs d'agrégation : f

Après la comparaison des descriptions variable par variable, on agrège ces comparaisons. Les fonctions d'agrégation proposées sont les suivantes (p_j : poids de la variable Y_j) :

- La somme des comparaisons, comme la distance City Block : $\sum_{j=1}^{p} p_j h_j(d_j, d'_j)$.
- La racine carré de la somme des carrés des comparaisons, comme la distance euclidienne.
- Le maximum des comparaisons, comme pour la distance de Chebyshev :

$$Max_{j=1}^{p}\{p_j h_j(d_j, d'_j)\}$$

- La distance de Minkowsky d'ordre α : soit deux objets symboliques s_1 et s_2 de vecteurs de descriptions $\delta_1 = (\delta_1^1, ..., \delta_1^p)$ et $\delta_2 = (\delta_2^1, ..., \delta_2^p)$. Une mesure de ressemblance entre s_1 et s_2 est une fonction :

$$d : \Omega \times \Omega \to \mathbb{R}^+$$

$$(w_1, w_2) \to d(w_1, w_2) = (\sum_{j=1}^{p} h_j(d_j, d'_j)^\alpha)^{1/\alpha} \tag{1.5.15}$$

avec h_j les fonctions de comparaison entre les descriptions sur une variable symbolique Y_j.

1.5.3 Opérateurs d'appariement

Comme dans le cas des mesures de dissimilarités, on s'intéresse ici aux assertions symboliques, sachant qu'ici les deux assertions à comparer ont des status distincts : l'un ayant le rôle de sujet, l'autre de référent. Ceci suppose que ces deux éléments soient munis de descriptions compatibles. On dira que le domaine D de description d'un sujet est compatible avec le domaine D' de description d'un référent, si on peut écrire : $D = D_1 \times ... \times D_p$ et $D' = D'_1 \times ... D'_p$, où D_j et D'_j sont définis à partir du même ensemble $\mathcal{O}_j$ de valeurs élémentaires et où D_j est de type moins général que D'_j $(\forall j = 1, ..., p)$.

On mesure alors l'appariement à l'aide de fonctions globales d'appariement de la forme $f \circ \rho : D \times D' \longrightarrow [0; 1]$, où f est une fonction d'agrégation définie de $([0; 1])^p$ dans $[0; 1]$, et où ρ s'écrit $\rho = (\rho_1, ...\rho_p) : D \times D' \longrightarrow ([0; 1])^p$, chaque fonction d'appariement élémentaire ρ_j étant définie de $D_j \times D'_j$ dans $[0; 1]$.

Dans plusieurs cas, l'appariement est vu comme une similarité classique, laquelle peut être obtenue par transformation d'une dissimilarité. De nombreux indices d'appariement élémentaires rentrent dans ce cadre et sont définis par la transformation : $\rho = 1 - h$, où h fait référence à une fonction de distance élémentaire.

1.6 Travaux actuels en analyse de données symboliques

Nous allons citer quelques travaux importants réalisés au cours de la dernière décennie en analyse de données symboliques. Dans les méthodes de visualisation, une extension de l'analyse en composante principale a été réalisée. Les recherches de A. Choukria [Cho98] [CCD99] étendent les méthodes d'analyse factorielle à des données de type intervalle. Plusieurs méthodes de classification ont été adaptées aux données symboliques. Des méthodes de classification hiérarchique et pyramidale [BD99a] ont été déjà réalisées ainsi qu'une adaptation d'une méthode divisive de classification dans les travaux de M. Chavent [Cha97]. La méthode des nuées dynamiques a été aussi adaptée dans les travaux de A. De Reyniès [dR03] et [CCLV03]. Par ailleurs, en relation avec l'approche

descriptive citons la thèse de V. Stephan [Ste98] qui traite de l'extraction de données symboliques par généralisation. Ces travaux seront détaillés dans le chapitre suivant.

En ce qui concerne les méthodes explicatives, de nombreux travaux en segmentation ont été réalisés. Enfin, pour des ouvrages plus généraux concernant le formalisme, les méthodes et les applications de l'analyse de données symboliques on peut se référer à [BD99a].

Chapitre 2

Extraction de données symboliques : une approche supervisée par généralisation

2.1 Introduction

En analyse de données, les méthodes de généralisation permettent une synthèse des informations contenues dans des tableaux, en décrivant des concepts sous-jacents aux données. Elles sont non seulement un outil descriptif pour l'utilisateur, mais aussi une étape intermédiaire permettant d'autres analyses sur ces concepts. Ayant un ensemble d'individus G de Ω, ensemble de la population, le but de la généralisation est de construire une bonne représentation de G par un vecteur multidimensionnel résumant toutes les descriptions des individus (valeur réelle, valeur binaire, catégories ou modalités, ...). Une méthode classique consiste à associer à ce vecteur un poids et un scalaire résumant la dispersion de ces individus [SPA83]. Une seconde approche proposée dans le cadre de l'analyse de données symboliques (données ayant une structure complexe telles que les intervalles, les distributions, ...) [DK91] permet d'extraire à partir d'une base de données relationnelle un ensemble de descriptions. Cette généralisation constitue une première et importante étape de l'analyse de données symboliques car elle permet d'associer aux groupes d'individus une description de nature symbolique [Ste98] [SHL00]. Cette généralisation permet de résumer un ensemble de tuples, représentant un groupe, par une "bonne" description symbolique.

L'intérêt du choix des données symboliques pour modéliser les informations détaillées est à la fois :

- de prendre en compte dans la synthèse, la notion de variabilité intervenant au sein d'un groupe de données. Les résumés obtenus ne se limitent pas aux indicateurs centraux calculés sur les observations du groupe ;
- de permettre des analyses des données ultérieures sur les résumés obtenus.

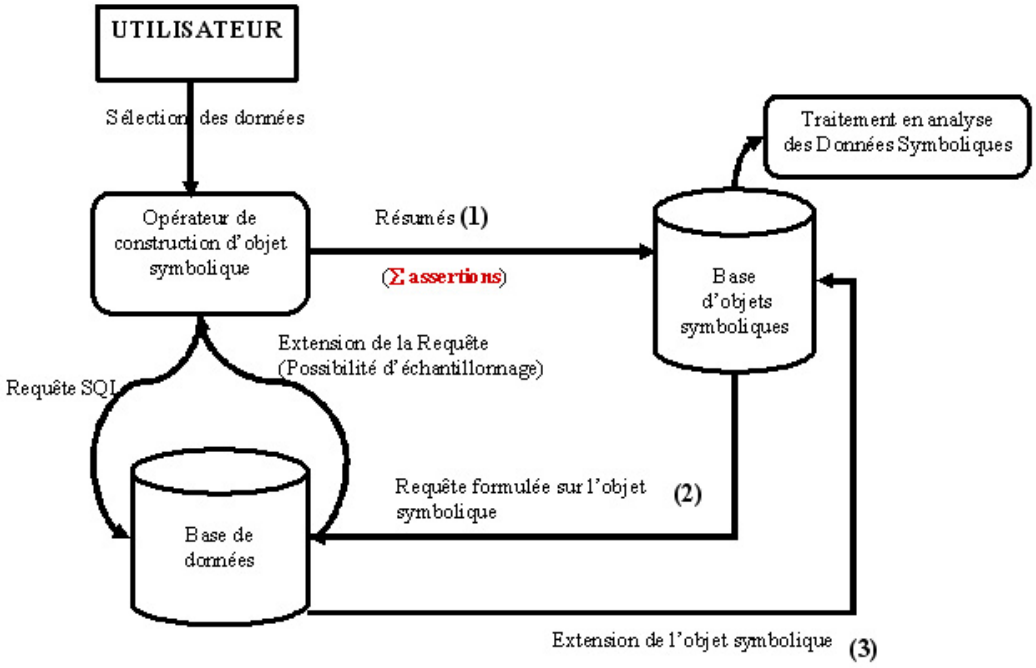


Figure 2.1 – Interface entre une base de données et les méthodes de l'analyse de données symboliques

Cette approche améliore le couplage entre le domaine de l'analyse de données et le domaine des bases de données relationelles (figure 2.1). En effet, le problème de résumé ou de synthèse consiste à décrire un ensemble de tuples provenant d'une base de données relationnelle par un objet générique (**(1)** figure 2.1). Généralement, les supports des agrégats (somme, moyenne, compte, minimum, maximum) sont utilisés pour modéliser cet objet générique qui est mis dans une table de données appartenant à la base. Dans ce cas, pour retrouver les tuples vérifiant l'objet générique on passe par un calcul de distance.

En revanche, la synthèse par généralisation symbolique permet une interaction entre les bases de données relationnelles et les bases des objets symboliques (figure 2.1) modélisant les objets génériques. En effet, la modélisation symbolique permet à tout moment le retour à la base de données par requête SQL (**(2)** figure 2.1) afin de retrouver les tuples vérifiant l'objet générique (**(3)** figure 2.1).

L'originalité de cette approche réside dans le fait que l'on permet à l'utilisateur de ne pas travailler sur une seule table relationnelle mais d'**exploiter les liens existant entre les tables**. La problèmatique consiste donc à traiter des informations provenant de plusieurs tables et de les résumer par un ensemble d'assertions (figure 2.1). Celles-ci constituent le tableau de données en entrée pour des traitements en analyse de données symboliques.

Souvent cette approche, sensible aux observations aberrantes et aux descriptions non homogènes, présente une sur-généralisation dans le sens où certaines descriptions incluent des individus dont la probabilité d'être éléments des groupes de départ est faible, éléments appelés atypiques ou virtuels. Une idée de réduction a été déjà développée dans le cadre des travaux de Véronique Stéphan [Ste98]. Cette réduction a permis surtout d'éliminer les individus atypiques mais sans pour autant améliorer l'homogénéité de certaines descriptions.

Dans ce chapitre nous allons introduire dans un premier temps, la méthode de généralisation utilisée dans le cadre de l'analyse de données symboliques [Ste98] [SHL00]. Nous allons ensuite, exposer les problèmes de cette généralisation et la réduction proposée dans des travaux antérieurs comme solution alternative aux observations atypiques [Ste98].

2.2 Extraction par la méthode de généralisation

Une base de données relationnelle peut stocker différents types de données ainsi que les relations entre ces données. La méthode de généralisation proposée dans le cadre de l'analyse de données symboliques peut être appliquée au niveau des bases de données relationnelles. Cette généralisation part d'une requête SQL et par regroupement résume

les tuples décrivant un groupe par une assertion de la forme donnée au chapitre 1, qui représente le concept associé au groupe. Dans ce qui suit, nous allons considérer deux ensembles, à savoir Ω l'ensemble des données représentant la population des individus ou son échantillonnage lorsque la taille de l'extension de la requête est trop grande pour être traitée en mémoire centrale, et l'ensemble E représentant les concepts issus des groupes d'individus. Les caractéristiques des individus, des groupes et les relations sont stockées dans la base de données. Nous supposons que l'utilisateur connait la structure de la base de données et qu'il est capable d'écrire une requête SQL.

Exemple 2.2.1. *Pour illustrer le principe d'extraction de données symboliques par généralisation, on considère cette partie du schéma relationnel d'une base de données nommée PERSONNEL :*

Position (Nom position, Activite, Salaire moyen, Departement)

Personnel (Id, Formation, Années de service, Sexe, Age, Nom position)

Supposons qu'on veuille construire des descriptions des positions dans une usine, selon l'information sur la carrière du personnel. Soit $\widetilde{Y_1}$ = *Formation,* $\widetilde{Y_2}$ = *Années de service,* $\widetilde{Y_3}$ = *Age, les trois variables définies par la clause* ***select*** *de la requête SQL suivante :*

Select *Id, Nom position, Formation, Années de service, Age*

From *Personnel*

Ω *est l'ensemble du personnel enregistré dans la base de données. Le personnel est groupé en classes, ici les positions dans l'usine. La table 2.1 représente le résultat de la requête SQL.*

Chaque individu appartient à un groupe, définie dans le champs `Nom position`. *Supposons que le nombre de groupes soit égal à K qui correspond au nombre de groupes différents donnés dans le champs* `Nom position`. *Le regroupement sur la variable nominale* `Nom position` *de la colonne* $j = 2$ *de la table 2.1 et les valeurs observées permettent de grouper les n individus en un des K groupes de la variable* `Nom position`.

Id	Nom position	Formation	Années de service	Age
1	Directeur	GRH	3	45
⋮	⋮	⋮	⋮	⋮
i	Chef de projet	informatique	5	35
⋮	⋮	⋮	⋮	⋮
n	Ouvrier	Technicien	20	40

Tableau 2.1 – Ensemble de tuples résultat de la requête SQL

Le tableau obtenu après généralisation est un tableau de données symboliques de taille $K \times p$, avec p le nombre d'attributs décrivant les groupes, où chaque case contient une description (voir tableau 2.2) :

E	Formation	Années de service	Age
⋮	⋮	⋮	⋮
Directeur	{info(20%), mark(50%), GRH(30%)}	[1 ; 28]	[32 ; 57]
⋮	⋮	⋮	⋮

Tableau 2.2 – Tableau symbolique obtenu après généralisation

Pour chaque label $e_i \in E$, on définit un objet symbolique $s_i = (a_{e_i}, R, d_{e_i})$, qui correspond à une généralisation des caractéristiques du personnel appartenant à la même $i^{ème}$ position dans l'usine. Pour le groupe "Directeur" de cette usine par exemple, 20% des directeurs ont une formation en informatique, 50% ont une formation en marketing et 30% ont une formation en gestion des ressources humaines. Les directeurs ont entre une année à 28 années de service. L'age de ce groupe de directeurs varie entre 32 ans et 57 ans. L'ensemble E est un ensemble d'objets ou d'éléments qui sont décrits par les variables symboliques (`Formation`, `Années de service`, `Age`*).*

En utilisant les agrégats, la modélisation du groupe "Directeur" peut être, par exemple, la suivante :

Directeur *Marketing* 14.5 44.5

Pour cela, on applique la fonction agrégat qui calcule la moyenne pour les attributs `Années de service` *et* `Age`. *Pareillement, pour l'attribut* `Formation`, *on sélectionne le mode du groupe Directeur (c'est la formation dominante obtenue à partir d'une fonction basée sur les agrégats).*

2.2.1 L'opérateur de généralisation

Dans le processus de généralisation, on considère que la population initiale $\Omega = \{1, ..., n\}$ est décrite par les n tuples renvoyés par une requête SQL. Les propriétés des individus sont caractérisées par p variables mono-valuées $\widetilde{Y_1}, ..., \widetilde{Y_p}$, qui correspondent aux p colonnes associées à la table renvoyée par la requête SQL, excepté les deux premiers attributs. Donc, à chaque individu $w \in \Omega$ correspond un vecteur de description $\widetilde{Y}(w) = (\widetilde{Y_1}(w), ..., \widetilde{Y_p}(w))$.

La table résultat de la requête SQL est de la forme suivante (voir tableau 2.3) :

Ω	G	$\widetilde{Y_1}$	...	$\widetilde{Y_j}$	...	$\widetilde{Y_p}$
1				...		
⋮				...		
i	$G(i)$	...	...	$\widetilde{Y_j}(i)$		
⋮				...		
n				...		

Tableau 2.3 – Ensemble de tuples résultat de la requête SQL

À chaque individu $w \in \Omega$, correspond un vecteur $(\widetilde{Y_1}(w), ..., \widetilde{Y_p}(w))$ et un label $G(w) \in \{e_1, ..., e_K\}$. De manière similaire, à chaque variable mono-valuée $\widetilde{Y_j} : \Omega \longrightarrow \mathcal{O}_j$, $j = \{1, ..., p\}$, correspond le $(j+2)^{ème}$ attribut de la clause Select. $\mathcal{O}_j$ est l'espace d'observation de $\widetilde{Y_j}$ défini à partir du type du $(j+2)^{ème}$ attribut.

Définition 2.2.1. Opérateur de généralisation [SHL00]

Avant la construction des objets symboliques, on définit une variable symbolique $Y_j : E \longrightarrow D_j$, où $E = \{e_1, ..., e_K\}$ est l'ensemble des labels des modalités prises par la variable mono-valuée G.

Ayant une population Ω décomposée en K ensembles de groupes $(G_1, G_2, ..., G_K)$, chaque variable symbolique $Y_j : E \longrightarrow D_j$ est définie à partir de la variable mono-valuée $\widetilde{Y_j}$ définie sur la population. Les deux principaux opérateurs de généralisation utilisés sont :

1. Pour une variable quantitative $\widetilde{Y_j}$, on définit une variable intervalle Y_j : $Y_j(k) = [a_k, b_k]$ tel que $a_k = \min_{w \in G_k} \widetilde{Y_j}(w)$ et $b_k = \max_{w \in G_k} \widetilde{Y_j}(w)$ avec $w \in \Omega$ un individu du groupe G_k, $k \in \{1, ..., K\}$. Dans ce cas, D_j est l'ensemble de tous les intervalles fermés inclus dans le domaine de $\widetilde{Y_j}$.

2. Pour une variable qualitative $\widetilde{Y_j}$, la variable symbolique correspondante peut être :
   - une variable multi-valuée Y_j, c'est-à-dire une correspondance (set-valued function) où $Y_j(k)$ représente l'ensemble des valeurs observées sur les individus du groupe G_k et D_j est l'ensemble des parties du domaine de $\widetilde{Y_j}$;
   - une variable modale Y_j où $Y_j(k) = (\eta_1^k, \ldots, \eta_l^k)$ est le l-tuple qui représente les poids associés aux catégories $1, \ldots, l$ de la variable $\widetilde{Y_j}$.

Le tableau obtenu après généralisation est un tableau de données symboliques de taille $K \times p$, où chaque case contient une description δ_{ij} (voir tableau 2.4) :

E	Y_1	...	Y_j	...	Y_p
e_1			...		
$\vdots$			...		
e_i	...	...	$Y_j(i) = \delta_{ij}$		
$\vdots$			...		
e_K			...		

Tableau 2.4 – Tableau symbolique obtenu après généralisation

Pour chaque label $e_i \in E$, on définit un objet symbolique $s_i = (a_{e_i}, R, d_{e_i})$, qui correspond à une généralisation des caractéristiques des individus appartenant au même groupe. L'ensemble E est un ensemble d'objets ou d'éléments qui sont décrits par les variables symboliques $(Y_1, ..., Y_p)$.

L'élément $d_{e_i} = (\delta_{i1}, ..., \delta_{ip})$, noté aussi d_i pour simplifier et appelé description symbolique de l'élément i, est un vecteur qui correspond à la $i^{ème}$ ligne du tableau de données symboliques. d_i est la description de e_i par les variables symboliques $(Y_1, ..., Y_p)$.

La relation R est définie pour chaque variable. Si on compare un individu $w \in \Omega$ avec la description symbolique d_i de l'élément e_i de E, la relation R est la relation d'appartenance ($\in$). Si on compare un élément de D à la description d_i, dans ce cas la relation R est la relation d'inclusion ($\subseteq$).

2.2.2 Les requêtes SQL utilisées pour l'extraction

On présente maintenant les requêtes SQL qui sont utilisées par l'opérateur de généralisation pour l'extraction d'assertions à partir des bases de données relationnelles. Toutes les techniques des bases de données relationnelles sont utilisées afin de construire des tables pertinentes pour l'analyse. Par un même exemple nous allons illustrer ces cinq types de tables résultats des requêtes SQL prises en considération par l'opérateur de généralisation et permettant la construction de la base d'objets symboliques.

Exemple base VIN. *La base de données relationnelle utilisée est la base VIN. Les individus que nous voulons décrire sont les châteaux. La base contient* 23 *châteaux, qui ont été notés par* 21 *experts à partir d'un échantillon de bouteilles issues de* 3 *années différentes (*1983, 1985 *et* 1990*). Chaque château possède une appellation appartenant à une région. Le schéma relationnel de la base VIN est illustré par la figure 2.2.*

2.2.2.1 La requête de sélection de la population et son échantillonnage

C'est la requête qui permet de spécifier la population Ω, la variable de regroupement des individus de Ω ainsi que les variables descriptives (déjà introduite dans la section

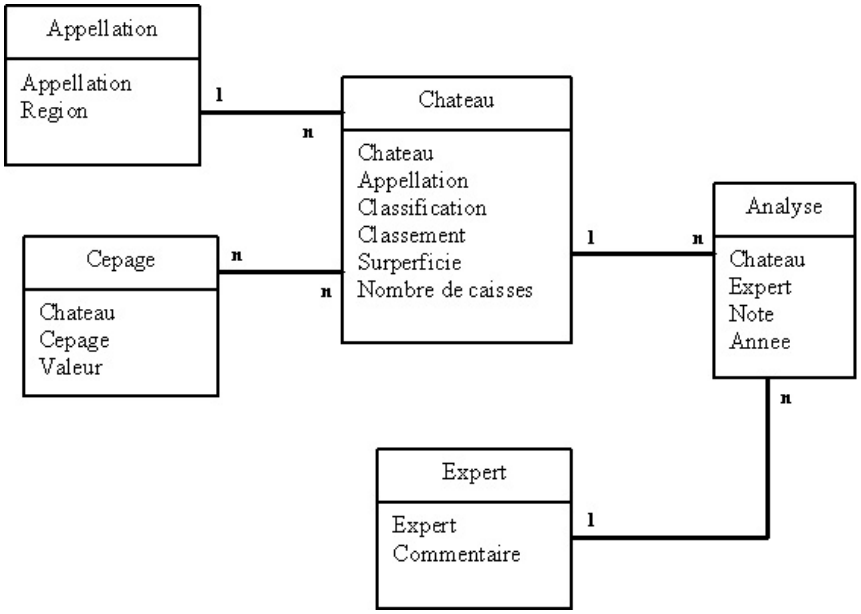


Figure 2.2 – Le schéma relationnel de la base de données VIN

2.2). Cette requête est utilisée pour extraire les informations suivantes :

$$Ind\_ID, Group\_ID, Ind\_V1, ..., Ind\_Vk$$

La table générée par la requête SQL doit contenir au moins les trois colonnes : Ind_ID (correspondant aux identifiants des individus), Group_ID (correspondant aux labels des groupes) et une variable Ind_Vj (décrivant les individus). Les variables quantitatives conduisent à la création de variables de type intervalle, alors que les variables qualitatives conduisent à la création de listes de valeurs ou de distributions de probabilités/fréquences.

Lorsque la taille de l'extension de la requête est trop importante, nous effectuons un échantillonnage. Pour plus de détails sur la méthode d'échantillonnage proposée voir [Ste98].

Exemple base VIN. *Revenons à l'exemple de la base VIN : pour cette première requête, nous avons choisi de sélectionner les châteaux par année comme individus (voir tableau 2.5), identifiés par la colonne* `ID` *issue du produit cartésien entre l'ensemble des châteaux et l'ensemble des années. La requête liste un château par année, par ligne. L'identifiant*

du groupe est défini par la seconde colonne et est le nom du château. Donc, à chaque château sera associé un objet symbolique. Les 21 experts ont noté les 23 châteaux pour les trois années (1983, 1985 et 1990) mais certaines notes peuvent être manquantes. Les individus sont donc décrits par les 21 notes des experts (variables quantitatives) et deux variables qualitatives : la `Classification` *et le* `Classement`*. La requête permettant d'obtenir la table 2.5 est une requête analyse croisée qui est une spécificité d'Access, et on ne la retrouve généralement pas dans les autres SGBD. La requête SQL correspondante est la suivante :*

Transform *Sum(Analyse.Note)* ***As*** *[La valeur]*

Select *[Chateau].[Chateau] & [annee]* ***AS*** *ID, Chateau.Chateau, Chateau.Classification, Chateau.Classement*

From *Chateau* ***Inner Join*** *Analyse* ***ON*** *Chateau.Chateau = Analyse.Chateau*

Group by *[Chateau].[Chateau] & [annee], Chateau.Chateau, Chateau.Classification, Chateau.Classement*

Pivot *Analyse.Expert*

ID	Chateau	Classification	Classement	Expert1	...	Expert21
Ausone1983	Ausone	No		70	...	70
Ausone1985	Ausone	No		56	...	78
Ausone1990	Ausone	No		74	...	82
⋮	⋮	⋮		⋮	...	⋮
Margaux1983	Margaux	No		66	...	95
Margaux1985	Margaux	No		75	...	96
Margaux1990	Margaux	No		84	...	75

Tableau 2.5 – Les tuples résultats de la requête de sélection de la population (Château) de l'exemple VIN

Le résultat de l'opérateur de généralisation sur cette table est un ensemble de 23 groupes. À chaque groupe est associé une assertion décrivant un château et qui est constituée de 21 variables symboliques intervalles et de 2 variables symboliques multinominales (à savoir `Classification` *et* `Classement`*). Les variables intervalles constituent chacune*

la variation des notes d'un expert pour un château.

2.2.2.2 La requête de sélection des variables portant sur les groupes

La requête SQL fournie, doit permettre d'extraire des informations supplémentaires décrivant les groupes :

$$Group\_ID, Group\_V1, ..., Group\_Vk$$

la table générée par la requête doit au moins contenir deux colonnes (Group_ID correspondant aux labels des groupes et une variable Group_Vj sur les groupes). Cette requête permet d'enrichir la description d'une assertion par une ou plusieurs caractéristiques du groupe. Comme ces caractéristiques sont observées au niveau des groupes, elles sont appelées variables natives uni-valuées.

Exemple base VIN. *Revenons à l'exemple de la base VIN, nous pouvons ajouter des attributs décrivant les châteaux. Par exemple nous pouvons rajouter les attributs* `appellation`, `superficie` *et* `nombre de caisses` *(voir tableau 2.6). La requête correspondante est la suivante :*

Select *Chateau, Appellation, Superficie, Nombre de Caisses*

From *Chateau*

Chateau	Appellation	Superficie	Nombre de Caisses
Ausone	Saint Emilion	7	1800
Cheval Blanc	Saint Emilion	35	12500
⋮	⋮	⋮	⋮
Margaux	Margaux	66	25000

Tableau 2.6 – Exemple de table résultat de la requête de sélection des variables Appellation, Superficie et Nombre de Caisses portant sur les Châteaux

2.2.2.3 La requête de sélection d'une variable native multi-valuée

Cette requête est utilisée afin d'extraire une variable multi-valuée qui décrit les groupes. Ceci conduit à l'enrichissement de la description d'une assertion par une variable multi-valuée. Les informations extraites par la requête sont les suivantes :

$$Group\_ID, QUALIT\_V, Cardinal$$

La première colonne de la table résultat contient les labels des groupes, la seconde contient une des modalités de la variable multi-valuée et la troisième une pondération de la modalité.

Exemple base VIN. *Pour ce type de requête, nous pouvons ajouter les cépages pour chaque château. Chaque château est constitué d'un ou plusieurs cépages (voir tableau 2.7). La requête correspondante est la suivante :*

Select *Chateau.Chateau, Cepage, Valeur*

From *Chateau, Cepage*

Where *Chateau.Chateau=Cepage.Chateau*

Chateau	Cepage	Valeur
Ausone	Cabernet-Franc	50
Ausone	Merlot	50
Cheval Blanc	Cabernet-Franc	40
Cheval Blanc	Merlot	60
⋮	⋮	⋮
Margaux	Cabernet-Franc	2,5
Margaux	Cabernet-Sauvignon	75
Margaux	Merlot	20
Margaux	Petit-Verdot	2,5

Tableau 2.7 – La table résultat de la requête de sélection de la variable native multi-valuée Cépage

2.2.2.4 La requête de sélection d'une nouvelle variable symbolique

Les informations extraites par cette requête sont les suivantes :

$$Group\_ID, Min\_Value, Max\_Value$$

Avec Min_Value et Max_Value sont respectivement les bornes inférieure et supérieure de l'intervalle de la nouvelle variable décrivant les groupes correspondant à la variable de regroupement $Group\_ID$.

2.2.2.5 Les requêtes de sélection de taxonomies

Lorsqu'il existe une taxonomie sur les valeurs d'une variable particulière, cette information peut être ajoutée grâce à des requêtes SQL. Deux manières différentes permettent d'extraire des taxonomies :

- $$ATT\_Value, NIV1\_Value, NIV2\_Value, \ldots, NIVp\_Value$$

  Cette représentation est appropriée si toutes les feuilles de la taxonomie ont la même profondeur. Dans le cas d'une succession de relations $1-n$ du schéma, le résultat de la requête SQL fournit l'ensemble des chemins complets de la taxonomie. Un chemin complet est une succession d'arêtes reliant une feuille à la racine. Un exemple typique est une requête qui retourne : Ville, département, région, pays et qui définie une taxonomie sur la variable Ville.

- $$FILS\_ATT\_Value, Parent\_ATT\_Value$$

  Cette représentation Fils/Parent d'une taxonomie lie les différentes valeurs du domaine d'un attribut. Dans le cas d'une auto relation $1-n$, le résultat de la requête nous fournit simplement l'ensemble des arêtes de la taxonomie. Cette limitation est dûe à l'impossibilité en SQL de calculer une fermeture transitive.

  Exemple base VIN. *Dans la base VIN, une taxonomie peut être créée au niveau de la variable* `Region`. *En effet, comme on le voit au tableau 2.8, chaque appellation appartient à une région particulière (voir figure 2.3).*

Appellation	Region
Etranger	Monde
France	Monde
Graves	France
Haut-medoc	Medoc
Libournais	France
Margaux	Medoc
Medoc	France
Pauillac	Medoc
Pomerol	Libournais
Saint-emilion	Libournais
Saint-estephe	Medoc
Saint-julien	Medoc

Tableau 2.8 – La table résultat de la requête de sélection de taxonomies (Fils/Parent)

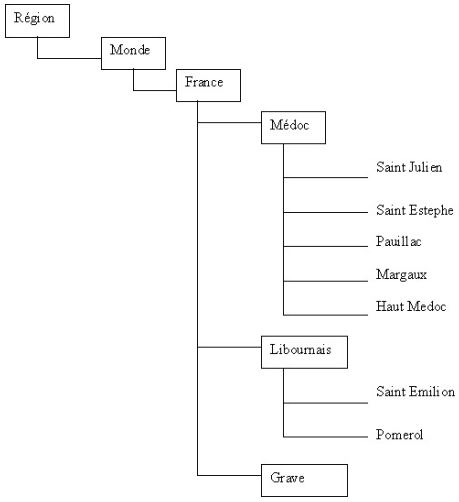


Figure 2.3 – Vue de la taxonomie du tableau 2.8

2.2.3 Les variables de type mère-fille dans une base de données

Les variables de type mère-fille sont définies à partir des règles de non applicabilité de certaines variables. Par exemple une variable "fille" Y devient inapplicable lorsqu'une autre variable "mère" Y' prend ses valeurs dans un sous ensemble S' de son domaine d'observation $\mathcal{O}'$. Par exemple, la variable `salaire` ne peut être observée que sur l'ensemble des personnes actives. Cette connaissance supplémentaire peut être associée aux assertions par la définition de régles de la forme suivante :

Si Y'$\in$ S' alors Y applicable

Exemple base VIN. *Dans la base VIN, les variables* `Classement` *et* `Classification` *sont des variables de type mère-fille. En effet, la variable* `Classement` *n'est applicable que si la variable* `Classification` *prend la valeur "Yes".*

Si Classification $\in \{Yes\}$ *Alors Classement applicable*

2.2.4 La jointure symbolique

Une des caractéristiques importantes et utiles de l'opérateur de généralisation est la jointure symbolique. En effet, disposant de deux tableaux d'assertions décrivant les mêmes objets par deux ensembles de variables différentes, la jointure des deux tableaux est possible. Par exemple (voir figure 2.4), on peut décrire des régions par leurs ménages dans un premier temps et par leurs écoles dans un deuxième temps. La jointure des deux tableaux symboliques permet d'avoir un tableau symbolique décrivant les mêmes régions par les ménages et les écoles.

Définition 2.2.2 (La jointure de deux tableaux symboliques). Soient deux tableaux symboliques X_1 et X_2. Les variables symboliques décrivant ces deux tableaux sont respectivement $Y_{11}...Y_{1p}$ et $Y_{21}...Y_{2q}$. Les deux tableaux X_1 et X_2 décrivent respectivement les deux ensembles E_1 et E_2. La jointure de X_1 et X_2, notée join(X_1, X_2), est un tableau symbolique défini comme suit :

- $E = E_1 \cap E_2$ (*i.e.* l'ensemble des entités ou objets symboliques du tableau symbolique obtenu est l'intersection des deux ensembles d'objets symboliques de X_1 et

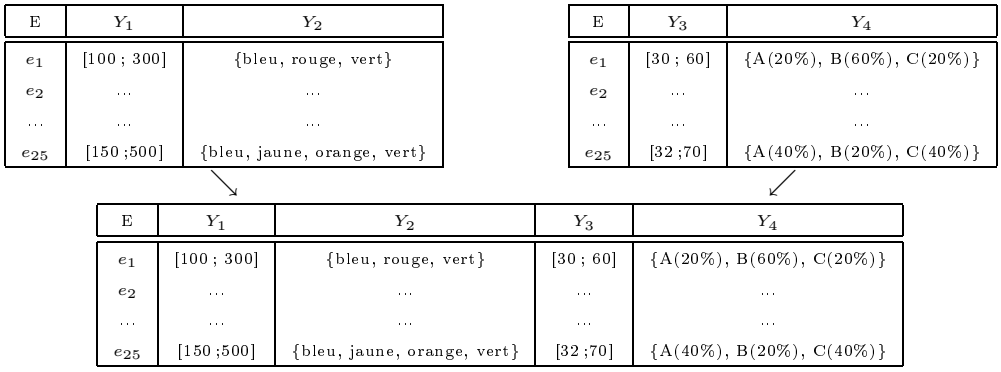


Figure 2.4 – La jointure entre deux tableaux d'assertions décrivant les mêmes objets symboliques par deux ensembles de variables

X_2) ;

- les variables décrivant join(X_1, X_2) sont : $Y_{11}...Y_{1p}, Y_{21}...Y_{2q}$ (*i.e.* la concaténation des deux ensembles de variables décrivant X_1 et X_2) ;
- pour chaque $e \in E$, on définit join$(X_1, X_2)(e)$= $(X_1(e), X_2(e))$. Ainsi, le tableau symbolique X = join(X_1, X_2) résultant, a le format $|E| \times (p + q)$;
- la définition de taxonomies sur des variables de X_1 ou X_2 reste maintenue sur le tableau résultant ;
- les variables de type mère-fille définies par des règles dans les tableaux X_1 ou X_2 sont aussi maintenues au niveau du résultat de la jointure.

2.3 Critères de qualité pour évaluer une description généralisée

On reprend ci-dessous les définitions de divers critères concernant les descriptions symboliques, données dans [Ste98].

- Le critère de recouvrement

  La qualité d'une description est facteur de son pouvoir de recouvrement. Une assertion s est une bonne généralisation de G si elle recouvre correctement les

individus appartenant à G :

$$Rec(s, G) = \frac{card(ext(s|G))}{card(G)}$$

C'est un des critères les plus utilisés pour sélectionner les descriptions de bonnes qualités. La valeur de ce critère est toujours comprise entre 0 et 1. Quand la valeur du critère est proche de 1, la description s est de bonne qualité.

- Le critère d'homogénéité

  La qualité d'une description dépend de l'homogénéité de la répartition des individus. L'assertion s est une bonne généralisation si les individus de G sont répartis de manière uniforme dans l'hypercube induit par s. Le modèle descriptif correspondant aux assertions suppose en effet :

  – une répartition uniforme des individus sur les intervalles de généralisation ;

  – une indépendance entre variables.

  Ce critère mesure donc la cohérence entre l'hypothèse d'uniformité de la répartition des points dans l'hypercube et la répartition observée.

- Le critère de densité

  Nous mesurons la qualité d'un ensemble d_i de descriptions d'un groupe G_i généralisé, par un bon compromis entre l'homogénéité de la distribution des individus dans l'ensemble de descriptions et l'ensemble de recouvrement de d_i dans G_i.

  Définition 2.3.1. : Le critère de densité d'une assertion [Ste98]

  Nous mesurons la qualité d'une assertion par un critère de densité, qui est le nombre d'individus recouverts par l'assertion par unité de volume de la description.

  Soit un groupe G_i, une partie de la population Ω, et son assertion correspondante s_i de description d_i. On définit la densité de s_i par rapport à G_i comme :

$$Dens(s_i) = \frac{|ext(s_i|G_i)|}{vol(d_i)}$$

  $d_i = (\delta_{i1}, \delta_{i2}, ..., \delta_{ip})$ étant la description de G_i, le volume de d_i est défini comme :

$$vol(d_i) = \prod_{j=1}^{p} \mu(\delta_{ij}) \quad \text{où} \quad \mu(\delta_{ij}) = \begin{cases} card(\delta_{ij}) & \text{si} \quad Y_j \quad qualitative \\ \overline{\delta_{ij}} - \underline{\delta_{ij}} & \text{si} \quad Y_j \quad quantitative \end{cases}$$

Dans le cas d'existence de règles, le volume est remplacé par le potentiel de description [Car96], comme nous l'avons indiqué dans le chapitre 1.
Donc plus la densité au sein de l'assertion est élevée, plus la qualité de la généralisation augmente.

2.4 Problème de la généralisation symbolique et réduction

Soit G un groupe d'individus de Ω, sélectionné par une requête à partir d'une base de données relationnelle. Le but de la généralisation est de chercher une représentation de cet ensemble d'individus de manière à résumer la requête en une description plus simple que l'énumération de tous les tuples de son extension. Cette généralisation des groupes G_i, à partir d'une table relationnelle par des assertions, induit souvent une surgénéralisation. En effet, cet opérateur symbolique traite les groupes variable par variable et ne prend pas en compte les associations qui peuvent exister entre les variables. Ceci se traduit par l'existence d'un grand nombre d'observations potentielles, observations qui ne correspondent à aucune description des individus de G_i. Cet opérateur est aussi sensible aux observations aberrantes, observations dont la probabilité d'appartenir au concept associé à G_i est faible.

L'amélioration de la qualité de la généralisation est cruciale dans la suite des traitements en analyse de données symboliques, car les méthodes sont souvent sensibles aux observations aberrantes et à l'hétérogénéité de certaines descriptions.

Dans ce qui suit, nous allons présenter la méthode de réduction permettant d'éliminer les observations atypiques, qui a été proposée par V. Stéphan dans le cadre de sa thèse [Ste98].

Une amélioration de l'opérateur de généralisation a été réalisée. Cette amélioration consiste à greffer une méthode de spécialisation permettant de réduire la description initiale obtenue par généralisation, en autorisant l'élimination des individus atypiques afin de minimiser le volume de la description. La notion d'assertion $\alpha-$généralisante utilisée dans l'algorithme de réduction proposé [Ste98] a été introduite. En effet, l'assertion s $\alpha-$généralisante par rapport à G correspond à l'assertion de plus forte densité sous

contrainte d'un recouvrement supérieur ou égal à $(\alpha \times card(G))$.

Définition 2.4.1. Une assertion $s = (d, a)$, $\alpha-$généralisante par rapport à G, est une assertion qui vérifie :

$$card(ext(a|G) \geq \alpha \times |G|)$$

$$vol(d) = min_{s'}(\{vol(d')/|ext(a'|G)| \geq \alpha \times |G|\})$$

Le but de cette réduction est de trouver un seuil α^* optimal qui offre le meilleur compromis entre la réduction du volume et la perte d'information qu'elle induit. Il s'agit de calculer pour tout seuil, la densité retenue de la α-généralisation associée, c'est-à-dire la densité de l'assertion α-généralisante par rapport à la densité de l'assertion initiale. Le seuil α^* correspond au point de plus forte décroissance sur la courbe de l'inverse de la densité relative.

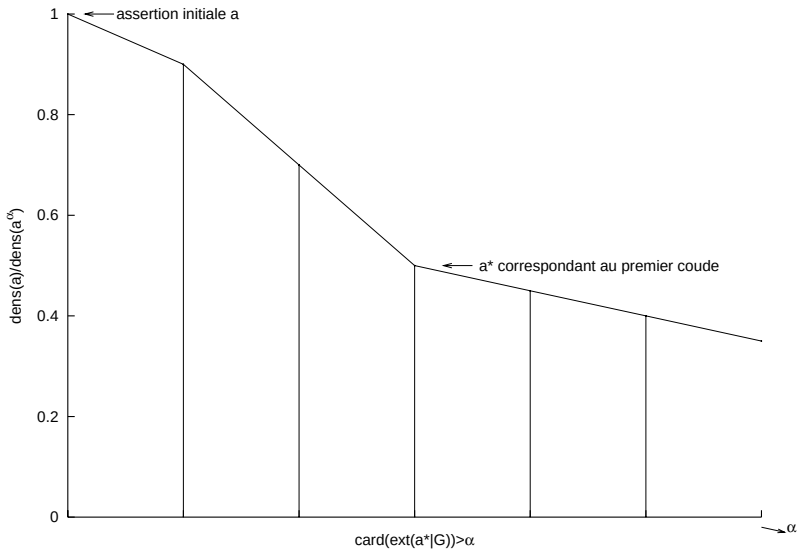


Figure 2.5 – Courbe de l'inverse de la densité relative

Sur la figure 2.5, on a en ordonnée le rapport $\frac{dens(a)}{dens(a^\alpha)}$, avec α le seuil de recouvrement variable. $s = (d, a)$ correspond à l'assertion généralisante, tandis que $s^\alpha = (d^\alpha, a^\alpha)$ correspond à l'assertion α-généralisante.

L'algorithme de réduction de [Ste98] est fondé sur les deux critères suivants :

– le calcul de la densité d'un hypercube qui s'appuie sur le calcul du recouvrement et du volume d'une assertion. Ces critères prennent en compte à la fois des variables

quantitatives, qualitatives et taxonomiques ;

- le choix de l'assertion réduite optimale qui correspond au meilleur point de la courbe de décroissance construite à partir de l'ensemble des solutions admissibles. Ce point correspond donc à la solution optimale, dans le sens où elle offre un bon compromis entre le pouvoir de recouvrement sur les individus et une description de faible volume.

Par un exemple succinct, nous présentons le problème de la généralisation. Nous allons utiliser les données de Ruspini constituées de 75 points.

Exemple 2.4.1. ***Exemple basé sur les données de Ruspini [Rus70]***

Soit un groupe G_1 formé de 75 points de $\mathbb{R}^2$ ($\widetilde{Y1}$ et $\widetilde{Y2}$). Les tuples formant G_1 et la description de ce groupe après l'application de la généralisation sont présentés aux figures 2.6, 2.7. La représentation suivante, appelée assertion, est une représentation de l'objet

Id	Groupe	$\widetilde{Y1}$	$\widetilde{Y2}$
1	G_1	45	17
2	G_1	53	18
...	G_1	...	...
75	G_1	23	68

Figure 2.6 – Les tuples formant G_1

E	Y1	Y2
G_1	[2, 131]	[17, 115]

Figure 2.7 – La description de G_1 après généralisation

symbolique associé au tableau de données de Ruspini :

$$a = [Y_1 \in [2, 131]] \wedge [Y_2 \in [17, 115]]$$

La figure 2.8, représente l'hypercube obtenu après généralisation, associé à la description des données de Ruspini.

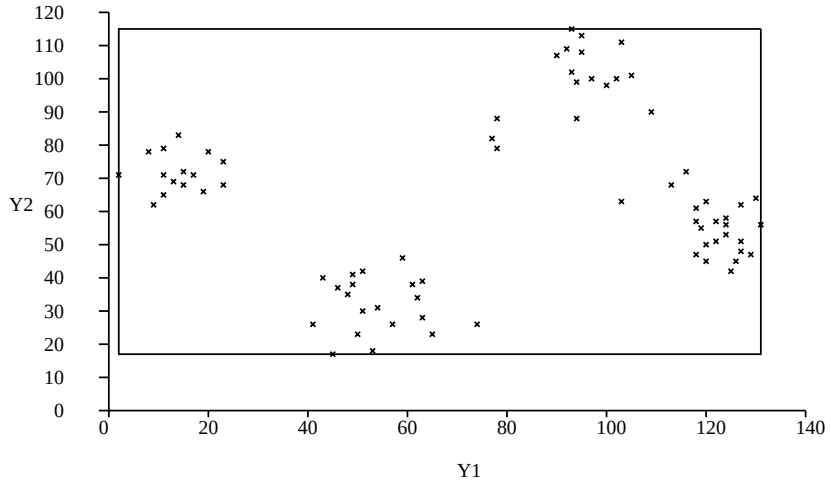


Figure 2.8 – L'hypercube généralisant le groupe G_1

La généralisation symbolique obtenue a induit une sur-généralisation, qui se traduit par la présence de zones ne possédant aucune observation, ce qui diminue la qualité de cette dernière. Considérons à présent l'individu w de description $\delta = (40, 110)$. Nous vérifions facilement que $a(w) = 1$, alors que la description de w ne correspond à aucune description ayant participée à la généralisation.

2.5 Conclusion

L'opérateur de généralisation symbolique permet d'obtenir des objets symboliques à partir d'une base de données relationnelle de manière supervisée. Cette généralisation étant basée sur un regroupement variable par variable, elle permet d'obtenir des descriptions assez simples mais présente souvent une sur-généralisation. Cette sur-généralisation se traduit par la présence d'observations atypiques ainsi que des descriptions non homogènes.

La réduction s'avère utile pour réduire les observations atypiques mais n'améliore pas pour autant l'homogénéité de certaines descriptions. Dans le cas de l'exemple de Ruspini, une réduction ne peut pas être appliquée et donc la qualité de notre description ne pourra pas s'améliorer. Dans le chapitre qui suit, nous allons proposer une étape de

décomposition permettant d'améliorer l'homogénéité des descriptions.

Deuxième partie

Amélioration du processus d'extraction des données symboliques et nouvelle méthode de classification d'un tableau de dissimilarités

Chapitre 3

Modélisation et extraction de données symboliques

3.1 Introduction

La généralisation supervisée à partir des bases de données relationnelles décrite au chapitre 2 favorise des descriptions hétérogènes quand des associations entre les variables existent. L'application de la méthode de réduction peut ne pas améliorer l'homogénéité de ces descriptions. Pour résoudre ce problème, **nous proposons une méthode de décomposition des descriptions par partitionnement des individus**. Ce partitionnement vise à réduire les associations entre les variables tout en enrichissant les descriptions et possède l'avantage d'associer à chaque classe d'individus sa généralisation au moyen d'une assertion. Cette décomposition nécessite une méthode de partitionnement sur chacun des groupes comme par exemple une méthode divisive de classification. Une méthode de décomposition, proposée dans ce travail, est issue de la méthode de classification décrite dans la thèse de Marie Chavent [Cha97], [Cha98]. La construction des différents segments se réalise par division successive de l'ensemble des individus. Un nœud est toujours divisé en deux nœuds fils en minimisant le critère d'inertie intraclasse. Ces deux nœuds fils sont obtenus en choisissant la meilleure coupure sur une variable. La classification obtenue est facilement interprétable grâce à l'arbre de décision binaire obtenu. **Plusieurs modélisations seront proposées afin de décrire les objets obtenus**.

Dans ce chapitre et dans un premier temps, nous allons décrire la méthode divi-

sive de classification proposée [Cha98], [Gol02], [GL02], [GL03]. Nous décrirons ensuite, l'association de cette dernière au processus de généralisation pour finir par exposer les performances et l'utilité de cette décomposition dans l'enrichissement des descriptions par généralisation symbolique. Dans un deuxième temps, **nous proposons une seconde approche par classification qui permet de construire un ensemble de classes homogènes qui seront modèlisées sous la forme d'objets symboliques** [CCLG01]. La méthode de classification utilisée est l'algorithme des cartes auto-organisatrices de Kohonen [Koh97].

3.2 Amélioration de la méthode de généralisation

Nous proposons d'utiliser la méthode divisive développée par Marie Chavent dans le cadre de la décomposition [Cha97], [CS99], [Cha00]. La méthode de M. Chavent a été développée dans le cadre des données symboliques et sur des tableaux de dissimilarités. Dans cette section, nous allons commencer par détailler la méthode proposée par M. Chavent. Comme dans notre cas les données sont quantitatives ou qualitatives, **nous adaptons cette méthode à notre cas afin de l'intégrer au processus de généralisation symbolique**.

3.2.1 La décomposition : une méthode divisive de classification

Dans le cadre des données symboliques [Cha97], l'ensemble des objets à classer est constitué d'objets symboliques qui sont décrits par des variables intervalles, multi-valuées ou modales. La recherche de la solution optimale ne se fait qu'à partir des partitions admissibles qui doivent respecter les structures des variables. Le choix du critère d'évaluation des bipartitions construites à chaque étape est un critère d'homogénéité H égal à la double somme pondérée des carrés des dissimilarités entre les individus appartenant à la même classe :

$$H(C) = frac12\mu(C) \sum_{w_j \in C} \sum_{w_{j'} \in C} p_j p_{j'} d^2_{jj'}$$

où p_j est le poids de l'individu w_j ; $d_{jj'}$ est la dissimilarité entre les individus w_j et w'_j et $\mu(C) = \sum_{w_j \in C} p_j$.

Ce critère nécessite le calcul du tableau de dissimilarités. Une bipartition admissible est une partition induite par une question binaire qui dépend du type de la variable. En effet, elle est de la forme "$Y_i \leq c$" si Y_i est une variable univaluée, avec c une valeur de coupure du domaine de Y_i et de la forme "$Y_i \in \{m_k, ..., m_j\}$" si Y_i est une variable multi-valuée ou modale, avec $\{m_k, ..., m_j\}$ est une partie du domaine de Y_i. L'algorithme divisif proposé dans [Cha97] est le suivant :

Initialisation :

$P_1 = \Omega$;

$k \leftarrow 1$;

Tant Que $k < K - 1$ **alors** :

1. pour chaque classe $C \in P_k$, choisir parmi les bipartitions (C_1, C_2) de C induites par les *questions binaires*, la partition qui minimise :

$$H(C_1) + H(C_2) = \frac{1}{2\mu(C_1)} \sum_{w_j \in C_1} \sum_{w_{j'} \in C_1} p_j p_{j'} d^2_{jj'} + \frac{1}{2\mu(C_2)} \sum_{w_j \in C_2} \sum_{w_{j'} \in C_2} p_j p_{j'} d^2_{jj'}$$

2. puis choisir la classe $C \in P_k$ qui maximise :

$$W(P_k) - W(P_{k+1}) = H(C) - H(C_1) - H(C_2)$$

3. $P_{k+1} = P_k \cup \{C_1, C_2\} - \{C\}$
4. $k \leftarrow k + 1$;

Fin Tant Que

K est le nombre de classes choisi et qui permet d'arrêter le processus divisif, alors que $\mu(C_1)$ et $\mu(C_2)$ sont les poids respectifs de C_1 et C_2.

Remarque 3.2.1. *L'arbre de décision obtenu est une hiérarchie sur les* $(K+1)$ *classes qui est indicée par* $W(P_k) - W(P_{k+1})$ *où les singletons sont les feuilles de l'arbre. Ainsi une classe divisée avant une autre est représentée par un palier plus haut dans l'arbre hiérarchique.*

3.2.2 Adaptation de la méthode divisive à notre cas

Quand le nombre d'individus est grand (dépassant 1000), le temps nécessaire au calcul du tableau de dissimilarités entre les individus deux à deux augmente rapidement. Comme dans notre cas les données sont quantitatives ou qualitatives, le centre de gravité d'une classe peut être calculé.

La complexité de l'algorithme est liée au nombre des bipartitions :

- Si la variable Y_i est quantitative, on évalue au maximum $(n-1)$ bipartitions, n étant le nombre d'individus de la classe en question. En effet, si on dispose de n observations, pour la variable Y_i il y a $(n-1)$ points de coupure sur cette variable (Figure 3.1).

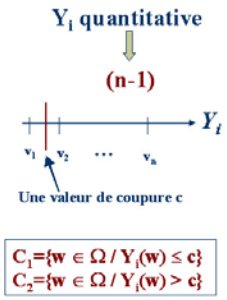


Figure 3.1 – Les différentes coupures possibles pour une variable quantitative à n valeurs

- Si la variable Y_i est qualitative ordinale, on évalue au maximum $(m-1)$ bipartitions, m étant le nombre de modalités de Y_i (Figure 3.2).

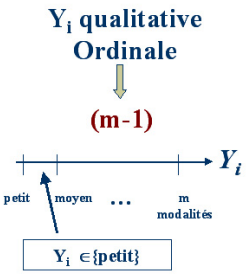


Figure 3.2 – Les différentes coupures possibles pour une variable qualitative ordinale à m modalités

- Dans le cas d'une variable qualitative nominale, on se heurte à un problème de

complexité et le nombre de dichotomies du domaine d'observation est alors égal à $(2^{m-1}-1)$. Dans notre cas et dans le cas pratique, nous nous limitons aux variables qualitatives nominales ayant au maximum 12 modalités.

Chaque individu i est décrit par p variables, Y_1, Y_2, ...,Y_p, par un vecteur w_i, doté d'un poids p_i, avec $i = 1, ..., n$. Généralement les poids p_i des individus sont égaux à 1 ou à $1/n$.

Le critère d'inertie étant un critère classique et universellement utilisé pour l'évaluation d'une partition, nous choisissons ce critère pour mesurer l'homogénéité des classes de la partition. Dans ce cadre l'inertie d'une classe C se calcule en fonction du centre de gravité g_C de la classe :

$$I_{g_C}(C) = \sum_{w_i \in C} p_i d_M^2(w_i, g_C) = H(C) \tag{3.2.1}$$

avec :

$$g_C = \frac{1}{\mu(C)} \sum_{w_i \in C} p_i w_i \qquad \text{et} \qquad \mu(C) = \sum_{w_i \in C} p_i \quad , \quad p_i \text{ est le poids de l'individu } \mathrm{w_i}$$

La distance d_M choisie est :

- la distance euclidienne si les variables sont continues ou qualitatives ordinales ;
- la distance du ϕ^2 sur le tableau disjonctif complet dans le cas des variables qualitatives nominales.

M étant une matrice symétrique définie positive :

- Cas quantitatif ou qualitatif ordinal

$$\forall w_i \in R^p, d_M^2(w_i, g_k) =^t (w_i - g_k) M (w_i - g_k) \tag{3.2.2}$$

Avec g_k, le centre de gravité de la classe C_k, et si $M = Id$ nous avons $d^2(w_i, g_k) = \sum_{j=1}^{p} (w_i^j - g_k^j)^2$

- Cas qualitatif nominal

Dans ce cas, les individus sont décrits par un ensemble de variables qualitatives nominales. Une étape préliminaire consiste d'abord à recoder les valeurs observées sous la forme d'un tableau à codage disjonctif complet formé de valeurs dans

l'espace $B^m = \{0,1\}^m$.

$$\phi^2(w_i, g_k) = \sum_{j=1}^{m} \frac{1}{f_{.j}} (\frac{f_{ij}}{f_{i.}} - g_k^j)^2$$

avec $f_{ij} = \frac{w_i^j}{n_{..}}$, $f_{i.} = \frac{n_{i.}}{n_{..}}$, $f_{.j} = \frac{n_{.j}}{n_{..}}$, $n_{..} = np$ où n est le nombre total d'individus, p le nombre total de variables, m le nombre total de modalités et w_i^j le représentant de l'individu i pour la modalité j dans l'espace $B = \{0,1\}$.

$$n_{.j} = \sum_{i=1}^{n} w_i^j$$

$$n_{i.} = \sum_{j=1}^{n} w_i^j = p$$

$$g_k^j = \frac{\sum\limits_{w_i \in C_k} f_{i.} \frac{f_{ij}}{f_{i.}}}{\sum\limits_{w_i \in C_k} f_{i.}} = \frac{w_i^j(k)}{p * n(k)}$$

avec $w_i^j(k) = \sum\limits_{w_i \in C_k} w_i^j$ et $n(k)$ le nombre d'individus dans la classe C_k.

$$\phi^2(w_i, g_k) = \sum_{j=1}^{m} \frac{np}{n_{.j}} (\frac{w_i^j}{n_{i.}} - g_k^j)^2$$

$$\phi^2(w_i, g_k) = (n * p) \sum_{j=1}^{m} \frac{1}{n_{.j}} (\frac{w_i^j}{p} - g_k^j)^2 \tag{3.2.3}$$

Donc le critère d'inertie d'une classe C dans le cas qualitatif nominal est le suivant :

$$I_{g_C}(C) = \sum_{w_i \in C} f_{i.} \phi^2(w_i, g_C)$$

Étant donné un espace muni d'une métrique euclidienne, alors minimiser le critère d'homogénéité associé à la partition en deux classes (C_1, C_2) de C revient, d'après le théorème de Huygens, à maximiser l'inertie interclasse B, définie par :

$$B(C_1, C_2) = \mu(C_1) d^2(g_C, g_{C_1}) + \mu(C_2) d^2(g_C, g_{C_2})$$

Dans notre cas, nous utilisons le critère d'inertie interclasse B pour l'évaluation des bipartitions. L'utilisation de ce critère permet de réduire considérablement la complexité des calculs ce qui permet d'obtenir une implémentation efficace de l'algorithme.

De plus, l'inertie interclasse B correspond aussi au critère de Ward [War63] :

$$B(C_1, C_2) = \frac{\mu(C_1) * \mu(C_2)}{\mu(C_1) + \mu(C_2)} \ d^2(g_{C_1}, g_{C_2})$$

Le critère d'évaluation de la bipartition dans notre cas se résume donc à la simple distance entre les centres de gravité des deux sous-classes C_1 et C_2 formant la bipartition de C. De plus, lors de l'évaluation de l'ensemble des partitions admissibles, le calcul des centres de gravités g_{C_1} et g_{C_2} de deux coupures consécutives, c_i et c_{i+1}, se déduit facilement grâce aux formules :

$$g_{C_1}(c_{i+1}) = \frac{\mu(C_1) g_{C_1}(c_i) + \mu(A) g_A}{\mu(C_1) + \mu(A)}$$

$$g_{C_2}(c_{i+1}) = \frac{\mu(C_2) g_{C_2}(c_i) - \mu(A) g_A}{\mu(C_2) - \mu(A)}$$

Si la variable Y traitée est quantitative, A est l'ensemble des individus dont la valeur de Y est comprise entre c_i et c_{i+1}. Si la variable Y traitée est qualitative ordinale, A est l'ensemble des individus vérifiant la modalité entre les deux coupures consécutives c_i et c_{i+1}.

L'algorithme proposé est donc le suivant :

Initialisation :

$P_1 = \Omega$;

$k \leftarrow 1$;

Tant Que $k < K - 1$ **alors** :

1. Pour chaque classe $C \in P_k$
   - Pour chaque variable Y
     - Pour chaque coupure c_i, calculer l'inertie interclasse $B(C_1, C_2)$ de la bi-partition (C_1, C_2) de C

$$B(C_1, C_2) = \frac{\mu(C_1) * \mu(C_2)}{\mu(C_1) + \mu(C_2)} \, d^2(g_{C_1}(c_i), g_{C_2}(c_i))$$

   Parmi toutes les bipartitions induites, retenir celle qui maximise B

2. Puis choisir la classe $C \in P_k$ qui maximise :

$$W(P_k) - W(P_{k+1}) = I_{g_C}(C) - I_{g_{C_1}}(C_1) - I_{g_{C_2}}(C_2)$$

3. $P_{k+1} = P_k \cup \{C_1, C_2\} - \{C\}$
4. $k \leftarrow k + 1$;

Fin Tant Que

Le traitement des valeurs manquantes :

Dans la pratique il y a souvent des données manquantes. Plusieurs techniques ont été développées pour résoudre ce problème. Certaines de ces techniques éliminent les individus ayant des valeurs manquantes, d'autres estiment plutôt la distance entre deux vecteurs de données ayant des valeurs manquantes. La méthode que nous proposons traite les valeurs manquantes. Comme nous avons choisi le critère d'inertie, cela nécessite au départ le calcul des centres de gravité de toutes les bipartitions induites pour chaque étape. Le calcul d'un centre de gravité se réalise sur l'ensemble des valeurs observées.

L'individu ayant une valeur manquante sur la variable traitée, sera ensuite affecté à la classe de la bipartition induite dont le centre de gravité est le plus proche. La technique choisie ici pour le calcul de la distance entre le vecteur w_i et le vecteur centre de gravité g_k, contenant des valeurs manquantes, est la suivante [JD88] :

On définit d'abord la distance d_j entre deux vecteurs sur une variable j :

$$d_j = \begin{cases} 0 & \text{si } w_i^j \text{ ou } g_k^j \text{ est manquant} \\ w_i^j - g_k^j & \text{sinon} \end{cases}$$

ensuite la distance entre w_i et g_k est :

$$d^2(w_i, g_k) = \frac{p}{p - d_0} \sum_{j=1}^{p} d_j^2$$

avec d_0, le nombre des valeurs manquantes dans w_i ou g_k ou les deux. Quand il n'y a pas de valeurs manquantes, notez que d n'est autre que la distance euclidienne.

3.2.3 Intégration de la décomposition à la généralisation symbolique

Cette décomposition a été intégrée au processus de généralisation symbolique à partir des bases de données relationnelles. Pour chaque groupe extrait de la base de données, la décomposition est appliquée sur ses individus. Le nombre de classes, K pour chaque groupe, est fixé par l'utilisateur. Nous allons utiliser le critère de densité, défini au chapitre 2, pour mesurer l'apport de la décomposition. Rappelons que la densité d'une assertion s de description d correspondante à un groupe G est définie comme :

$$Dens(s) = \frac{|ext(s|G)|}{vol(d)}$$

Exemple 3.2.1. *On reprend l'exemple de la figure 2.8 du chapitre 2 [Rus70], formé d'un groupe G_1 de 75 individus répartis sur deux variables quantitatives Y_1 et Y_2. Sur ce groupe, l'opérateur de généralisation produit l'assertion suivante :*

$$a = [Y_1 \in [2, 131]] \wedge [Y_2 \in [17, 115]]$$

La valeur du critère de densité calculé sur cette assertion est de l'ordre de 0.006.
L'application sur ce groupe de la méthode de décomposition, basée sur la méthode divisive proposée précédemment, permet de le diviser en 4 sous-groupes. Dans la première itération, la classe à diviser est l'ensemble des individus du groupe G_1 et parmi les p(n-1) bipartitions possibles (p : nombre de variables, n : nombre d'individus), soit 148 bipartitions (C1, C2), on choisit celle qui a la plus grande inertie interclasse. Ceci se traduit par

la question binaire "$Y_1 \leq 75.5$". Dans un deuxième temps, on divise l'une des classes C1 ou C2. La meilleure solution est de choisir C1 et sa bipartition (C11, C12) correspondant à la question binaire "$Y_2 \leq 54$", car elle induit la plus grande diminution de l'inertie intraclasse. Ensuite la classe C2 est choisie parmi l'ensemble des classes (C11, C12, C2) pour être divisée en (C21, C22). La question binaire correspondante est "$Y_2 \leq 75.5$". Comme le nombre de classes fixé au départ est atteint, on s'arrête à ce niveau. L'arbre de décision final induit par notre méthode divisive est donc le suivant :

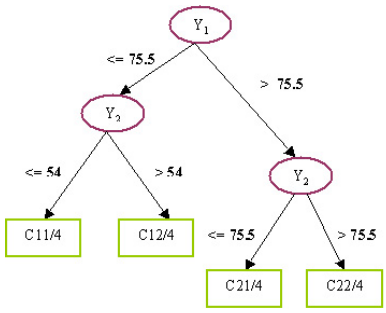


Figure 3.3 – Arbre de décision produit par la méthode divisive

Cet arbre de décision (figure 3.3) peut être aussi représenté par l'arbre hiérarchique de la figure 3.4. Une classe divisée avant une autre est représentée plus haut dans l'arbre hiérarchique afin d'avoir l'ordre de découpage (figure 3.4).

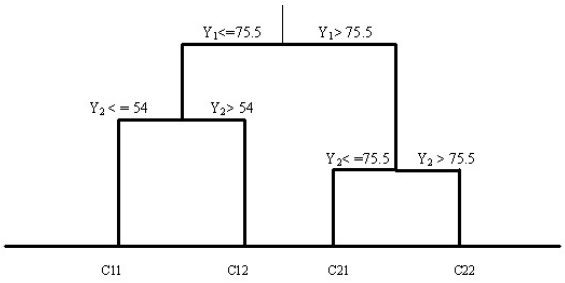


Figure 3.4 – Arbre hiérarchique produit par la méthode divisive

Les associations entre certaines variables favorisent l'apparition des individus ne cor-

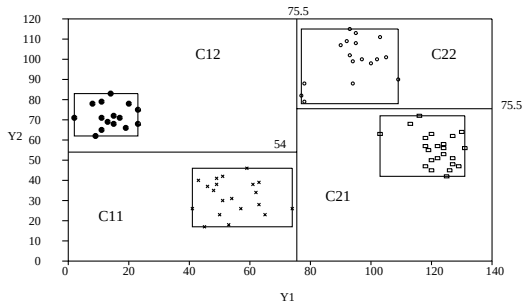


Figure 3.5 – La décomposition en 4 sous-classes et la généralisation obtenue

respondant pas à des individus observés au niveau de la description obtenue par généralisation symbolique. Comme on le voit sur la figure 3.5, la décomposition réalisée a permis d'éliminer les zones vides correspondant aux individus non observés et de créer des classes homogènes au sens de notre critère d'inertie sélectionné. Pour quantifier l'amélioration apportée aux descriptions par cette décomposition, on peut utiliser le critère de densité proposé par [Ste98]. En effet, la nouvelle valeur du critère de densité après décomposition est de l'ordre de $0,1$.

3.2.4 Modélisation des résultats de la décomposition

Chaque groupe étant divisé en un nombre fixe de sous-groupes. Nous avons montré dans le paragraphe précédent que cette décomposition peut être modélisée par un arbre de décision ou par un arbre hiérarchique. Avec ces modélisations, les traitements ultérieurs nécessiteront l'utilisation d'outils de la théorie des arbres, par exemple la distance entre deux groupes devra être basée sur une distance entre deux arbres. **Nous proposons une autre approche en utilisant la notion d'objets symboliques**.

Hordes :

La modélisation du groupe et donc du concept associé, peut être réalisée par un objet Horde. Rappelons qu'une horde est une conjonction d'assertions portant chacune sur un individu. Par exemple le groupe précédent pourra être défini par :

$$[[Y_1(w_1) \leq 75.5] \wedge [Y_2(w_1) \leq 54]] \wedge [[Y_1(w_2) \leq 75.5] \wedge [Y_2(w_2) > 54]]$$

$$\wedge[[Y_1(w_3) > 75.5] \wedge [Y_2(w_3) \leq 75.5]] \wedge [Y_1(w_4) > 75.5] \wedge [Y2(w_4) > 75.5]]$$

où chaque élément de cette horde est défini à partir d'une feuille de l'arbre de décision.

Le groupe peut être modélisé par une disjonction d'un nombre fini d'assertions. Chaque assertion représente aussi une feuille de l'arbre obtenu. Nous définissons donc une nouvelle représentation que l'on appelle multi-assertion.

Multi-assertion :

Définition 3.2.1 (Multi-assertions booléennes). Une multi-assertion booléenne M_s est un ensemble de k objets symboliques $\{(d_1, R, a_1), ..., (d_k, R, a_k)\}$ où chaque objet symbolique est une assertion. k est le nombre de sous-groupes. Une multi-assertion booléenne est donc une disjonction d'assertions booléennes.

Dans ce cas, l'extension de cette multi-assertion booléenne est égale à :

$$ext_\omega(M_s) = \{\omega \in \Omega | \max_{j=1,...,k}(a_j(\omega)) = 1\}$$

Une fois la décomposition réalisée, l'application de notre opérateur de généralisation sur le groupe G_1 renvoie la multi-assertion booléenne suivante :

$$[Y_1 \in [41, 74] \wedge Y_2 \in [17, 46]] \vee [Y_1 \in [2, 23] \wedge Y_2 \in [62, 83]]$$

$$\vee[Y_1 \in [103, 131] \wedge Y_2 \in [42, 72]] \vee [Y_1 \in [77, 109] \wedge Y_2 \in [79, 115]]$$

Définition 3.2.2 (Multi-assertions modales). Une multi-assertion modale M_s est un ensemble de k objets symboliques $\{(d_1, R, a_1), ..., (d_k, R, a_k)\}$ où chaque objet symbolique est une assertion. k est le nombre de sous-groupes. Une multi-assertion modale est donc une disjonction d'assertions modales.

L'extension d'une multi-assertion modale M_s peut être définie de deux manières différentes :

- on considère tout d'abord que tout individu $\omega \in \Omega$ peut appartenir "plus ou moins" à l'extension de M_s, en fonction de son degré d'appartenance calculé comme suit :

$$ext_\omega(M_s) = \{\omega \in \Omega | (w, \max_j(a_j(\omega)))\}$$

– dans une deuxième approche, on peut considérer que l'appartenance d'un individu ω à l'extension de M_s est admise si le maximum des quantités $a_j(\omega)$ est au moins égale à un seuil α fixé :

$$ext_\omega(M_s) = \{\omega \in \Omega | \max_j(a_j(\omega)) \geq \alpha\}$$

3.2.5 Conclusion

L'intégration d'une décomposition dans le processus de généralisation améliore la qualité des assertions obtenues car elle permet d'homogénéiser les groupes. L'apport de cette décomposition peut être mesurer grâce aux critères de qualité définis au chapitre 2, session 2.3. Cette amélioration n'entraîne pas une perte d'information, comme dans la plupart des cas. Au contraire, elle apporte des connaissances supplémentaires sur les groupes extraits des bases de données relationnelles. La modélisation des résultats de la décomposition par le formalisme symbolique permet le couplage avec le domaine des bases de données relationnelles. En effet, nous avons proposé deux modélisations utilisant le formalisme symbolique. La première utilise le type horde déjà défini par E. Diday dans [Did89]. Nous proposons une deuxième modélisation en définissant un nouveau type du formalisme symbolique, qu'on a appelé multi-assertion.

3.3 Approche non supervisée : Construction automatique d'objet symbolique par classification

Nous présentons maintenant une approche non supervisée de construction des données symboliques. Cette approche est basée sur une classification par l'algorithme des cartes topologiques auto-organisatrices, suivie d'une modélisation de la classification obtenue. La classification automatique joue un rôle essentiel dans l'analyse de données par la découverte de structures intéressantes. De nombreuses méthodes classiques de classification automatique adoptent la même stratégie qui consiste à réduire d'abord la dimension des données par des méthodes telles que l'analyse factorielle, puis à réduire aussi le nombre d'individus par d'autres méthodes rapides de classification. L'algorithme

des cartes auto-organisatrices de Kohonen a deux sources d'inspiration, à savoir la classification et la réduction de dimension. Cet algorithme produit une carte topologique, à savoir un ensemble de neurones généralement disposés en forme rectangulaire sur une surface bidimensionnelle comprenant un nombre important de neurones, chacun caractérisé par un prototype. La proximité de ces neurones dans la carte correspond à la proximité des individus dans l'espace initial (de grande dimension), c'est ce que l'on appelle la préservation de la topologie. Bien que l'application naïve de l'algorithme des cartes auto-organisatrices permet de préserver la topologie, elle ne permet pas pour autant d'avoir des classes bien homogènes. En effet, l'utilisation de cette méthode pour la classification permet soit d'attacher chaque neurone à une classe (on parle alors de *micro-classes*), soit d'attacher plusieurs neurones à une classe (on parle alors de *macro-classes* ou *super classes* [HD98]). Dans le premier cas, on dispose d'une carte dont chaque case est caractérisée par un ensemble d'individus et il serait intéressant de modéliser ensuite chacune de ces cases afin de s'en servir de tableau de données pour d'autres algorithmes de classification appropriés. Plusieurs études [HD98], [US90], [ZyL93], [Mur95], [ZyL93] se sont penchées sur cette stratégie de réduction des données par les cartes topologiques suivie d'un algorithme de classification classique sur la carte. Dans ce cas, le tableau de données initial est remplacé par un autre tableau réduit dont chaque ligne est muni d'un poids (normalement le nombre d'individus dans les micro-classes), et d'un scalaire résumant la dispersion de ses individus. Nous proposons une modélisation permettant de préserver au maximum les informations portées par chacune des micro-classes. La modélisation est suivie de l'application d'un algorithme de classification approprié. Le cas des macro-classes quant à lui, s'appuie sur une classification appliquée sur la carte.

L'idée de développement d'une classification après une réduction des données par le biais d'une carte topologique n'est donc pas nouvelle puisque certains chercheurs comme Murtagh [Mur95], Ultsch [US90], Ambroise [ASBT00] et Zhang [ZyL93] ont introduit chacun cette notion d'une manière différente. Murtagh utilise la notion de la classification par contrainte de contingence. Ambroise combine la classification hiérarchique et le SOM. Toutes ces méthodes remplacent le tableau de données initial par un autre tableau réduit dont les lignes représentent les neurones et les variables sont les moyennes des variables

originales calculées sur les neurones. Ce nouveau tableau réduit est alors traité avec un algorithme classique de classification. Ultsch utilise quant à lui la notion de matrice unifiée, ou encore U-Matrix, basée sur la notion de plateau et vallée sur la U-Matrix. Zhang, introduit une nouvelle méthode appelée le SOM analytique, basée elle aussi sur la notion de plateau et vallée de la carte de densité SOM.

Notre approche est différente de ces dernières, puisqu'on parle de modélisation qui permet d'avoir des structures plus riches. Contrairement aux méthodes citées ci-dessus, qui résument les données par des scalaires, notre méthode se base sur une modélisation des neurones par des descriptions et donc réduit la perte d'information.

3.3.1 La modélisation des neurones d'une carte topologique

Le processus d'apprentissage de l'algorithme SOM renvoie un ensemble de neurones associés à des individus. L'association des individus au même neurone traduit leur similarité. Ce processus donne aussi une notion de voisinage qui signifie que les neurones voisins sont semblables. Nous proposons une modélisation qui permet d'intégrer la variabilité des individus d'un neurone. Deux modélisations sont proposées : une modélisation des micro-classes par des assertions et une modélisation des macro-classes par des objets plus complexes.

3.3.1.1 Modélisation des Micro-classes

Les variables traitées étant continues, deux approches peuvent caractériser cette modélisation [CL00], [CCLG01] :

- **Modélisation par des variables intervalles :** une approche ensembliste qui permet de représenter chaque neurone par l'ensemble des individus et donc par une assertion booléenne [DK91] ayant comme intension le prototype associé au neurone. Pour cela, on utilisera l'opérateur de généralisation symbolique décrit au chapitre 2 :
  - soit z_i, le vecteur de description du $i^{ème}$ individu de dimension p, avec $i = 1, ..., n$;
  - soit z_{ij}, la $j^{ème}$ composante du $i^{ème}$ individu avec $j = 1, ..., p$;

– w_c le vecteur poids du neurone c.

la modélisation symbolique d'un neurone c est la suivante :

$$\bigwedge_j [a_{cj}, b_{cj}], \text{ pour } j = 1, ..., p$$

avec : $a_{cj} = \min\{z_{ij}/i \in w_c\}$ et $b_{cj} = \max\{z_{ij}/i \in w_c\}$

Dans ce cas, l'entrée de la méthode de classification symbolique sera une séquence d'objets assertions booléennes $s_1, s_2, ...$, décrits par des variables intervalles.

– **Modélisation par une distribution Gaussienne :** une approche probabiliste qui permet de représenter chaque case par une loi décrivant la variabilité des valeurs observées. En effet, comme chaque case de la carte topologique est associée à un hyper-rectangle l'alternative est d'associer à ces hyper-rectangles une distribution uniforme. Une autre est de représenter chaque neurone par une distribution normale multi-variée de moyenne $m^{(r)}$ et matrice de variance-covariance V_r, où $m^{(r)}$ est le vecteur des moyennes des variables calculées sur les individus du neurone r, et V_r calculée aussi sur les individus du neurone r.

3.3.1.2 Modélisation des Macro-classes

L'idée est d'appliquer un algorithme de classification sur la carte et ensuite de modéliser les neurones appartenant à la même classe qu'on appelle macro-classes par des descriptions. Nous proposons une première approche qui consiste à modéliser les macro-classes par un objet assertion. Pour ce faire, on applique l'opérateur de généralisation symbolique sur l'ensemble des individus formant la macro-classe. Mais avec cette approche on perd l'information d'appartenance des individus à différents neurones.
Une autre approche permettant de garder cette information consisterait à modéliser les macro-classes par une multi-assertion ou par une horde (voir 3.2.4). Nous proposons donc de modéliser chaque neurone appartenant à la macro-classe par une assertion après application de l'opérateur de généralisation et ensuite la macro-classe sera modélisée par une disjonction d'assertions ou une conjonction d'assertions.

3.4 Conclusion

Dans ce chapitre, nous avons évoqué le problème de l'hétérogénéité des descriptions créées par la généralisation symbolique supervisée. Nous avons ensuite proposé une solution qui consiste à décomposer les descriptions afin d'améliorer la qualité de ces dernières. La méthode proposée possède certaines limites connues des méthodes de segmentation : on est confronté au problème de la complexité lorsque les variables qualitatives ont beaucoup de modalités. Le problème de la détermination du nombre de classes n'est pas non plus résolu. En effet, c'est à l'utilisateur de le fixer. Il serait donc intéressant de trouver un critère qui correspond à un compromis entre le critère d'inertie et celui de la densité d'une description qui permet de fixer le nombre de classes. Le nombre de classes optimal peut aussi correspondre au meilleur point de la courbe de croissance du critère de densité (voir figure 3.6).

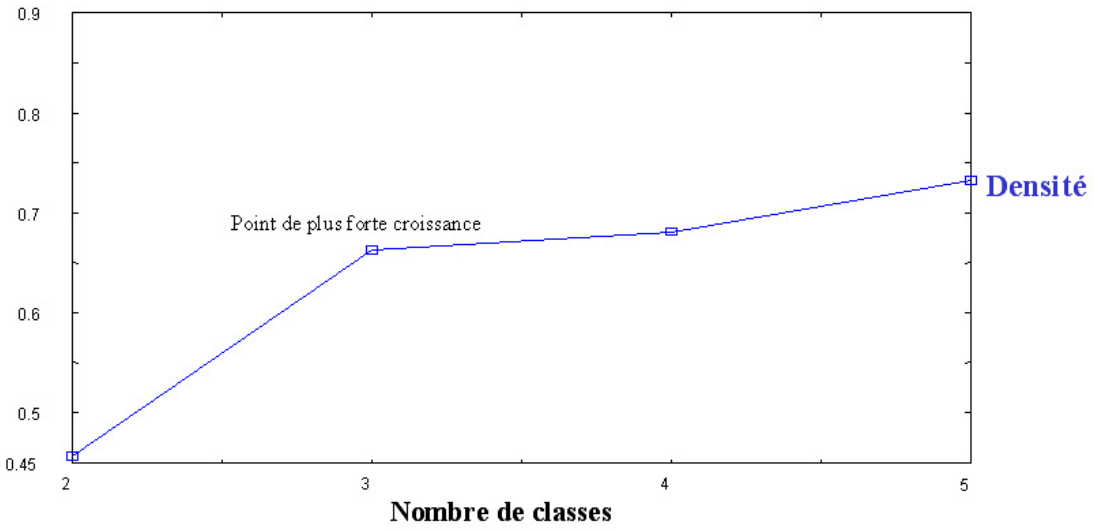


Figure 3.6 – Le point de plus forte croissance du critère de densité correspond à 3 classes

Des modélisations ont été proposées afin d'associer à chacun des groupes trouvés une description généralisant les valeurs observées au sein du groupe. Nous avons aussi proposé une généralisation symbolique non supervisée basée sur l'algorithme des cartes topologiques de Kohonen. Cette généralisation permet de réduire les données et d'obtenir aussi des descriptions ayant une structure complexe et décrivant des groupes homogènes.

Chapitre 4

Adaptation des cartes topologiques auto-organisatrices aux tableaux de dissimilarités

4.1 Introduction

En analyse de données, on est souvent amené à traiter des données ayant une structure complexe, comme par exemple des données **symboliques** [BD99a], introduites au chapitre 1 (données ayant une structure complexe telles que les intervalles, les distributions, ...), des données **semi-structurées** (arbres, documents, XML), des données **fonctionnelles** (chaque individu est décrit par une fonction régulière discrétisée en un nombre fini de points d'observation), etc. Dans ce cadre, les méthodes classiques de classification basées sur le calcul d'un centre de gravité ne peuvent plus être utilisées car les individus ne sont plus décrits par de simples vecteurs de $\mathbb{R}^n$. Afin de résoudre ce problème, plusieurs solutions sont envisageables selon la nature des données : par exemple l'utilisation de techniques de recodage des descriptions [dR02] [dR03] pour les données symboliques, ou encore l'utilisation d'un opérateur de projection dans le cas de données fonctionnelles [RS97] [CGGR04]. Cependant, ces méthodes nécessitent une bonne connaissance *a priori* des données, et imposent en général à l'utilisateur de fixer certains meta-paramètres (par exemple, le choix de la base de projection dans le cas de données fonctionnelles).

Nous proposons comme solution alternative une adaptation des cartes de

Kohonen aux tableaux de dissimilarités [GCG03] [GCGR04]. En effet, le SOM est basé sur la notion de centre de gravité et malheureusement ce concept n'est pas applicable aux données complexes. Notre but est de modifier l'algorithme des cartes topologiques afin de permettre son application aux mesures de dissimilarités. Cette approche permet un traitement aisé des différents types de données, car seule la définition d'une mesure de dissimilarité est nécessaire au déroulement de la méthode.

La méthode que nous proposons répond aux objectifs suivants :

- traiter aussi bien les données classiques que les données complexes ;
- fournir une interprétation des neurones et donc des classes obtenues.

Dans notre algorithme, nous nous sommes basés sur la version "batch" des cartes topologiques de Kohonen [TLGC97] [Tea02] décrite au chapitre 1, section 1.2.3.3. Rappelons que dans cette version nuées dynamiques des cartes topologiques les prototypes ne sont plus recalculés à chaque fois qu'on présente une observation mais après une phase d'affectation de l'intégralité des données. C'est une version non stochastique de l'algorithme des cartes topologiques.

4.2 Cartes topologiques sur tableaux de dissimilarités

L'idée de travailler sur des tableaux de dissimilarités, permet de partir de données plus générales et de traiter tous types de variables (classiques et complexes).

Ayant des tableaux de dissimilarités, nous allons maintenant présenter l'algorithme des cartes auto-organisatrices sur ce type de tableau, qu'on appelle **DissSOM**.

4.2.1 Principe

Comme pour un SOM classique, la carte est décrite par un graphe (C, Γ). La principale différence est que l'on ne travaille plus sur l'espace $\mathbb{R}^p$ mais sur un ensemble sur lequel une dissimilarité notée d est définie (c'est un espace métrique où la dissimilarité est une distance).

L'espace de représentation L_c du neurone c est l'ensemble des parties de Ω de cardinal q fixé : chaque neurone c est représenté par un ***Individu référent*** : $a_c = \{z_{j_1}, ...z_{j_q}\}$,

avec $z_{j_i} \in \Omega$. On voit donc, que contrairement à l'algorithme classique, où chaque référent peut librement évoluer dans l'espace des données tout entier, dans l'approche proposée ici chaque neurone n'a qu'un nombre fini de représentations à sa disposition.
L'espace L de représentation de la partition est la carte $L(C, a)$, avec $a = \{a_c; c = 1, ..., m\}$ l'ensemble de tous les individus référents de la carte.

On définit une dissimilarité généralisée, d^T de $\Omega \times P(\Omega)$ dans $\mathbb{R}^+$, comme une mesure d'adéquation entre un individu $z_j \in \Omega$ et un individu référent a_c, par :

$$d^T(z_i, a_c) = \sum_{r \in C} K^T(\delta_{rc}) \sum_{z_j \in a_r} d^2(z_i, z_j)$$

avec $K^T(\delta_{rc})$ la fonction noyau qui dépend de la distance entre le neurone c et le neurone r et du paramètre de température T (par exemple, $K^T(\delta) = e^{-\frac{\delta^2}{2T^2}}$). Rappelons que plus le paramètre T est petit, plus le nombre de neurones inclus dans le voisinage est réduit.

Pendant l'apprentissage, on cherche à minimiser la fonction coût E suivante, en alternant les phases d'affectation et les phases de représentation :

$$E(f, L(C, a)) = \sum_{z_i \in \Omega} d^T(z_i, a_{f(z_i)}) = \sum_{z_i \in \Omega} \sum_{r \in C} K^T(\delta(f(z_i), r)) \sum_{z_j \in a_r} d^2(z_i, z_j) \qquad (4.2.1)$$

Cette fonction mesure l'adéquation entre une partition induite par la fonction d'affectation et une carte $L(C, a)$.

Lors de la phase d'affectation, la fonction d'affectation f est celle qui affecte tout individu z_i au neurone de la carte le plus proche au sens de la dissimilarité généralisée d^T. En cas d'égalité on affecte z_i au neurone de plus petit indice :

$$f(z_i) = arg \min_{c \in C} d^T(z_i, a_c) \qquad (4.2.2)$$

Par la définition de la fonction d'affectation la décroissance de E est assurée.

Lors de la phase de représentation, on cherche le système d'individus référents a^* qui représente au mieux l'ensemble des observations au sens de E. Le critère E étant additif, cette étape d'optimisation combinatoire peut être réalisée de manière indépendante pour chaque neurone. On minimise en effet m fonctions de la forme :

$$E_r = \sum_{z_i \in \Omega} K^T(\delta(f(z_i), r)) \sum_{z_j \in a_r} d^2(z_i, z_j) \tag{4.2.3}$$

Donc pour chaque neurone r, on cherche l'individu référent a_r qui minimise E_r. Autrement dit, trouver les q meilleurs individus $z_j \in \Omega$ qui représentent au mieux le neurone r. En cas d'égalité on sélectionne les individus z_j de plus petit indice.

Le critère E décroît donc jusqu'à convergence. Dans la version batch classique, la minimisation de la fonction E est immédiate car la position des vecteurs référents est donnée comme le centre de gravité du nuage de points pondéré par la fonction K.

4.2.2 L'algorithme DissSOM

Initialisation :

▶ $k = 0$

▶ choisir un système initial de référents a^0 et la carte $L(C, a^0)$. Fixer T à T_{max} et le nombre total d'itérations à N_{iter}

Itération : $k++$

À l'itération k, l'ensemble des individus référents a^{k-1} de l'étape précédente est connu. Calculer la nouvelle valeur de

$$T = T_{max} * (\frac{T_{min}}{T_{max}})^{\frac{k}{N_{iter-1}}} \qquad \text{[Tea02]}$$

▶ **Phase d'affectation :** mettre à jour la fonction d'affectation f_{a^k} associée au système a^{k-1}. On affecte chaque observation au référent défini à partir de l'équation (4.2.2).

▶ **Phase de représentation :** déterminer le nouveau système a^{k*} qui minimise la fonction $E(f_{a^k}, L(C, a))$. Pour chaque neurone, a_c^{k*} est défini de manière unique à partir de l'équation (4.2.3).

Répéter Itération jusqu'à ce que l'on atteigne $T = T_{min}$

4.2.3 Critères de convergence

Malgré son apparente simplicité, l'algorithme des cartes topologiques s'est révélé fort complexe à analyser d'un point de vue mathématique. Dans le cas multidimensionnel, il

n'existe pas à ce jour de résultats théoriques prouvant de manière rigoureuse la convergence de l'algorithme. Cela semble dû au fait qu'il est extrêmement difficile d'exprimer l'ordre dans un réseau de dimension supérieure à un. La convergence de l'algorithme stochastique dans le cas des cartes monodimensionnelles a été déjà prouvée [CF87] [CF97]. En pratique, il est difficile de fixer certains paramètres tels que le nombre d'itérations, N_{iter}, à effectuer pour observer la convergence de l'algorithme, ou les dimensions de la carte, $xdim$ et $ydim$, etc. Pour cela, nous allons présenter une série de critères dont l'évolution au cours du déroulement de l'algorithme permettent d'en juger la convergence et par la même de juger la pertinence des choix des différents paramètres de la carte considérée. Plusieurs de ces critères sont issus des méthodes de classification automatiques traditionnelles.

4.2.3.1 Erreur Kmeans

Dans l'algorithme Kmeans [Mac65], l'auteur définit l'erreur Kmeans comme la somme des distances au carré entre chaque individu et le prototype qui lui est associé, le prototype étant le centre de gravité dans le cas abordé. Cette erreur correspond à une erreur de quantification. Dans notre cas, le prototype ne correspond pas au centre de gravité de la classe dans laquelle se situe un individu et l'erreur Kmeans, dans notre cas, s'écrit alors :

$$E_{kmeans} = \sum_{z_i \in \Omega} \sum_{z_j \in a_{f(z_i)}} d^2(z_i, z_j)$$

4.2.3.2 Erreur de distorsion moyenne

$$E_{koho} = \sum_{z_i \in \Omega} \sum_{r \in C} K^T(\delta(f(z_i), r)) \sum_{z_j \in a_r} d^2(z_i, z_j)$$

Ce critère est la somme du produit de deux facteurs. Le premier facteur $K^T(\delta(f(z_i), r))$ contribue à la conservation de la topologie, tandis que le second, $\sum_{z_j \in a_r} d^2(z_i, z_j)$, contribue à la bonne représentation des entrées par les individus référents.

4.2.3.3 Changements d'affectations

Le nombre de changements d'affectation d'individus aux classes entre deux itérations successives est un des indicateurs les plus importants pour juger de la stabilité de la carte. En effet, le nombre de changements d'affectation doit diminuer rapidement et tendre vers 0 pour qu'il y ait stabilité. Il faut cependant noter que l'absence de changements d'affectation ne signifie pas qu'il y ait convergence de l'algorithme, car la version batch de l'agorithme étant déterministe, il est fort possible que les individus référents soient des minima locaux.

4.2.4 Initialisation de la carte

Dans notre algorithme les prototypes associés aux neurones de la carte ne peuvent être que les individus appartenant à l'ensemble d'apprentissage. Il existe donc deux différentes méthodes d'initialisation, l'initialisation "semi-aléatoire" et l'initialisation par analyse factorielle sur tableaux de distances (AFTD).

4.2.4.1 Initialisation "semi-aléatoire"

Cette stratégie d'initialisation est naturellement bien adaptée à notre cas. Elle consiste à choisir aléatoirement les vecteurs référents parmi les vecteurs de données.

4.2.4.2 Initialisation par AFTD

Cette méthode d'initialisation fait appel à la technique de l'analyse factorielle sur tableaux de distances (AFTD). Le but de l'analyse factorielle étant d'obtenir une représentation d'un ensemble d'individus dans un espace de dimension moindre de façon à maximiser l'inertie portée par cet espace. Ayant un tableau de dissimilarités nous utiliserons l'AFTD après constante additive décrite au chapitre 1, section 1.2.3.1. Le plan principal d'inertie fourni par l'AFTD est le plan qui conserve le mieux les proximités entre les individus. Or la principale caractéristique d'une carte de Kohonen est justement cette conservation de la notion de proximité entre espace des données et espace des neurones. En conséquence, l'algorithme de Kohonen va chercher une configuration

des prototypes respectant au mieux cette contrainte de proximité, et bien que rien ne contraigne les prototypes à être rassemblés sur un même plan, on peut estimer qu'après la phase d'organisation, les prototypes d'une carte de Kohonen seront disposés à des distances relativement faibles du plan principal d'inertie du nuage des données initiales. Ceci suggère d'initialiser la carte en plaçant au départ de l'algorithme les prototypes sur ce plan principal d'inertie, afin de limiter l'ampleur des modifications lors de l'apprentissage. De plus, en préorganisant ces prototypes, c'est-à-dire en posant côte à côte sur le plan des prototypes censés représenter des données proches, on limite encore le risque de formation de défauts topologiques en préservant tout dépliage de grande ampleur de la carte. On débutera donc l'apprentissage avec une grille de neurones posés sur le plan principal d'inertie.

Les étapes de l'algorithme permettant d'initialiser les prototypes au moyen d'une AFTD sont les suivantes :

1. calcul des valeurs propres et vecteurs propres ;
2. tri des résultats et choix des deux vecteurs propres associés aux deux premières valeurs propres ;
3. création d'une grille de points sur le plan formé par les deux vecteurs, centrée sur l'origine du nouveau repère, de dimensions $(var(x), var(y))$, avec $var(x) = |x_{min} - x_{max}|$ où x_{min} et x_{max} désignent respectivement les valeurs minimales et maximales observées dans l'espace de projection et $var(y) = |y_{min} - y_{max}|$, où y_{min} et y_{max} désignent respectivement les valeurs minimales et maximales observées dans l'espace de projection. On obtient ainsi les m $(xdim \times ydim)$ vecteurs coordonnées, ψ_i, des m neurones dans l'espace de projection (voir Figure 4.1) ;
4. ayant les coordonnées de l'ensemble des individus sur le plan de projection, déterminer le système d'individus référents initial a^0. Chaque neurone c sera représenté par l'individu i le plus proche de ψ_c au sens de la distance euclidienne dans l'espace de projection.

Le choix de la dimension de la carte en terme de neurones est un problème délicat qui ne se pose pas pour les autres algorithmes de classification non supervisée par prototypes, pour lesquels on fixe uniquement le nombre de classe à priori. En allant un peu plus loin

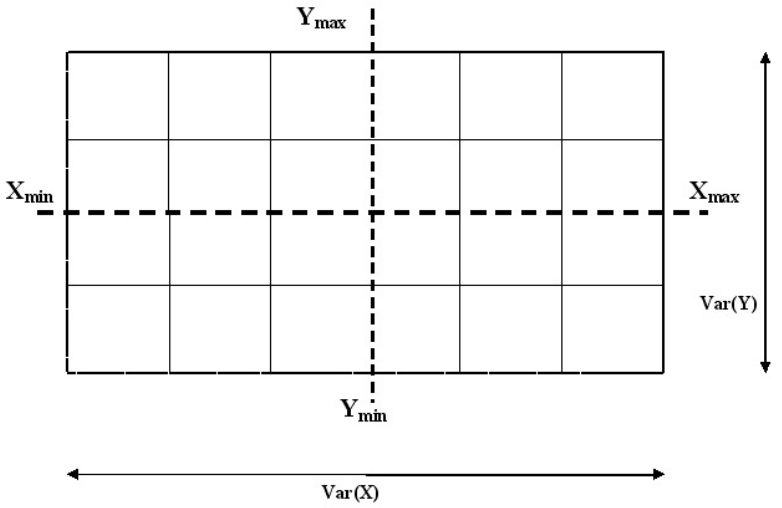


Figure 4.1 – Forme de la carte d'initialisation

dans la logique qui consiste à adapter les cartes aux données, il devient possible lors d'une initialisation par AFTD de fixer les dimensions latérales d'une carte en terme de nombre de neurones en utilisant les valeurs propres associées aux deux axes formant le plan principal d'inertie. Pour un nombre de classes fixé à priori, le nombre de neurones de chaque côté de la grille rectangulaire est proportionnel à la racine carrée de la valeur propre (écart-type) associée à l'axe directeur du même côté [Ele99]. En pratique, il s'agira de résoudre les deux équations suivantes :

$$\begin{cases} xdim \times ydim = & \text{nombre de classes} \\ \frac{xdim}{ydim} = & \frac{\sqrt{\lambda_1}}{\sqrt{\lambda_2}} \end{cases}$$

où $xdim$ et $ydim$ sont respectivement les valeurs approchées de la longueur et la largeur de la carte en terme de neurones, tandis que λ_1 et λ_2 sont respectivement les première et deuxième valeurs propres.

L'intérêt de cette méthode est de permettre de placer un plus grand nombre de neurones suivant l'axe portant la plus grande part d'inertie, et donc de placer d'autant plus de neurones que le nuage de points est allongé suivant une des deux directions principales.

4.2.5 La visualisation par l'AFTD

La représentation de la carte et des données se fait sur les plans de l'AFTD. Comme pour l'initialisation, on a utilisé les techniques de l'AFTD décrite au chapitre 1 (voir 1.2.3.1). En effet, l'analyse factorielle est communément utilisée pour visualiser la forme générale du nuage de points représentant un ensemble de données multidimensionelles par projection des individus sur un sous-espace formé par les deux ou trois premiers vecteurs propres (projection plane ou tridimensionnelle). Les prototypes d'une carte topologique, ici les individus référents, appartiennent eux aussi à l'espace dans lequel sont représentées les données ; il semble censé de les projeter dans le même sous-espace afin d'en visualiser les propriétés. Lors d'une projection bi ou tridimensionnelle, on tracera également les liens entre les prototypes associés à des unités voisines sur la carte, de façon à obtenir non plus un nuage de points en projection, mais bien une grille de points.

Cette méthode de visualisation est également parfaitement cohérente avec l'initialisation par AFTD, surtout si on garde en tête qu'à la suite d'une telle initialisation, il ne sera pas utile de modifier énormément la position des prototypes. L'expérience montre d'ailleurs que, lors d'une initialisation sur un plan autre que le premier plan d'inertie, la représentation des prototypes la plus adaptée est souvent celle faite sur le plan d'inertie grâce auquel on a initialisé les prototypes.

La visualisation par AFTD offre également l'énorme intérêt de pouvoir superposer les individus référents aux données, ceux ci coexistant dans le même espace d'origine. Il est donc facile de se rendre compte si la carte s'est correctement déployée sur l'ensemble du nuage de données, et d'associer visuellement les individus à la classe à laquelle ils sont censés appartenir.

Dernier avantage, une projection sur un sous-espace de l'AFTD est totalement déterministe, c'est-à-dire que deux projections issues des deux exécutions différentes, donneront systématiquement le même résultat visuel.

Il est également possible de projeter la grille des prototypes sur un sous-espace de dimension 3 formé par les trois premiers axes factoriaux du nuage de données. La visualisation d'une grille suivant trois dimensions nécessite toutefois de modifier l'angle de vue afin de saisir la forme globale de cette grille. Le fait de faire tourner la carte permet

en effet au cerveau de reconstruire la forme tridimensionnelle de celle-ci.

Tout comme l'initialisation par l'AFTD, les limitations principales proviennent du fait que lorsque les trois ou quatre premiers plans factoriels sont proches, la projection sur les sous-espaces de visualisation n'est pas totalement fiable. Il sera donc nécessaire d'indiquer le pourcentage d'inertie porté par ce sous-espace, afin que l'utilisateur évalue la fiabilité de la projection.

Une visualisation utilisant le Multi-Dimensional Scaling (MDS) [Tor52] qui est une méthode de réduction de dimensionalité peut être envisagée. L'avantage de cette méthode de réduction de dimensionalité est que les dissimilarités ne sont pas nécessairement des distances euclidiennes. Cependant son gros inconvénient est, outre sa lenteur, que l'algorithme de minimisation de la fonction coût est susceptible de tomber dans des minima locaux, et donc peut ne pas fournir de solution convenable, et peut également donner des résultats différents sur les mêmes données.

4.2.6 Exemples d'applications sur données simulées

On présente ici quelques traitements réalisés avec l'adaptation des cartes topologiques auto-organisatrices aux tableaux de dissimilarités. Dans tous ces exemples sur données simulées, la représentation d'un neurone se fait par un seul individu et donc q est fixé à 1.

1. Dans le premier cas, on dispose d'un ensemble de 500 observations ayant une forme particulière sur $\mathbb{R}^2$, voir figure 4.2.

   Dans un premier temps, nous avons opté pour la distance euclidienne. La méthode prend donc en entrée un tableau de distance euclidienne. On a fixé le nombre de neurones sur la carte à (9×3), l'intervalle de variation de $T = [0.4 : 6]$. Dans ce qui suit (de la figure 4.3 à la figure 4.6), nous présentons l'évolution de la carte au cours des itérations. La figure 4.3 illustre la carte initiale avec une initialisation "semi-aléatoire" sur les données. La figure 4.6 représente la carte finale obtenue projetée sur les données, après apprentissage avec l'algorithme proposé. Le résultat est satisfaisant, on obtient en effet, une bonne quantification de l'espace, tout en

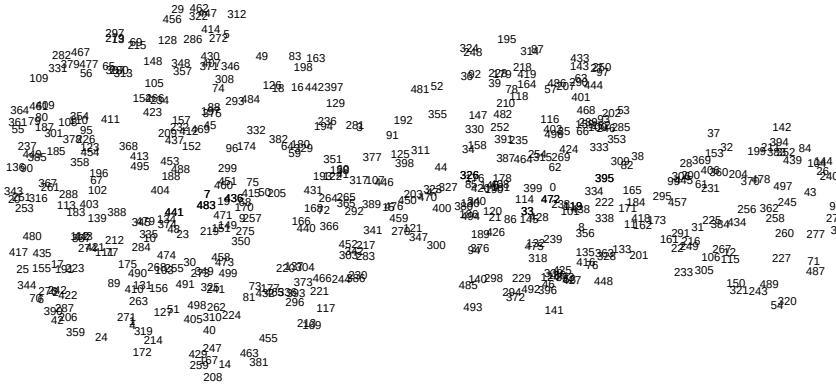


Figure 4.2 – Données sur $\mathbb{R}^2$

conservant la topologie des données.

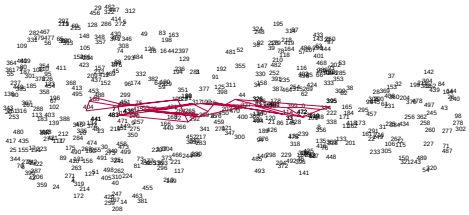


Figure 4.3 – Carte initiale (9×3) (initialisation "semi-aléatoire)

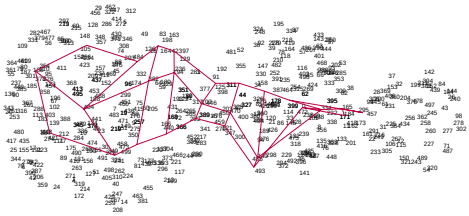


Figure 4.4 – Carte après 50 itérations

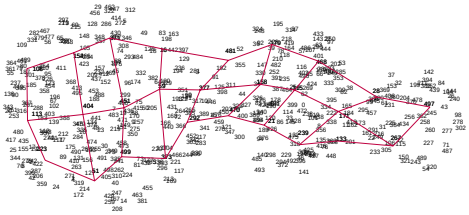


Figure 4.5 – Carte après 100 itérations

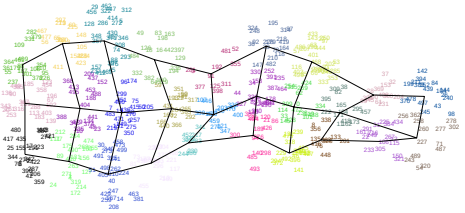


Figure 4.6 – Carte Finale

Dans un deuxième temps, nous avons opté pour la distance de City-Block. La méthode prend donc en entrée un tableau de distance L_1. On a gardé les mêmes paramètres d'entrée que pour la distance euclidienne. On représente sur la figure 4.7 la carte finale obtenue sur le plan de l'AFTD. Là aussi on obtient une bonne quantification de l'espace des données tout en conservant la topologie.

2. Dans la deuxième application, on dispose d'un ensemble de 1000 observations for-

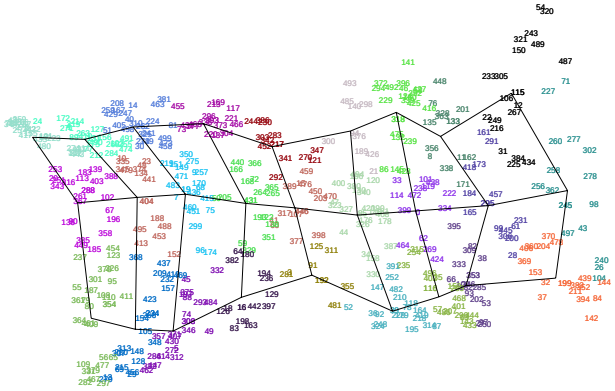


Figure 4.7 – Carte Finale sur plan de l'AFTD

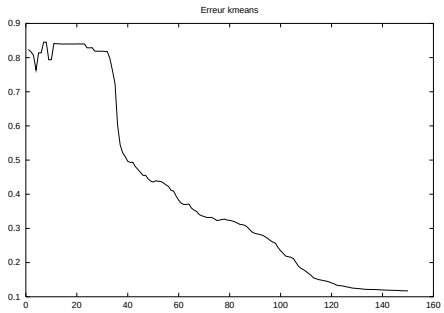
mant un cylindre dans $\mathbb{R}^3$. La méthode prend en entrée un tableau de distance euclidienne. On a fixé la taille de la carte à (21×3).

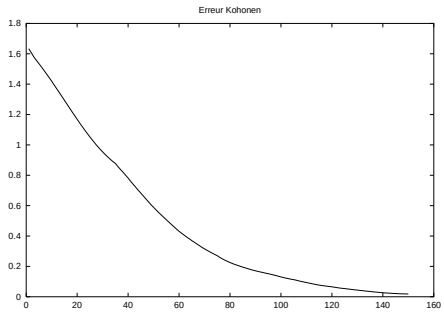
Liste des paramètres

Paramètres	Valeurs
Dissimilarité	distance euclidienne
Ensemble d'apprentissage	1000
Nombre d'itérations (N_{iter})	150
Nombre de neurones	$63 : 21 \times 3$
Initialisation	"Semi-aléatoire"
cardinal individus référents : q	1

Sur la figure 4.8, nous présentons la courbe d'erreur Kmeans et celle de distorsion moyenne.

Dans les figures allant de 4.9 à 4.12, nous présentons les données et l'évolution de la carte durant l'apprentissage. La figure 4.9 présente la carte initiale sur le cylindre avec une initialisation "semi-aléatoire". Sur la carte finale, présentée dans la figure 4.12, on remarque une bonne quantification et un bon déploiement de la carte, tout en préservant la topologie.





$$E_{kmeans} = \sum_{z_i \in \Omega} \sum_{z_j \in a_{f(z_i)}} d^2(z_i, z_j)$$

erreur Kmeans

$$E_{koho} = \sum_{z_i \in \Omega} \sum_{r \in C} K^T(\delta(f(z_i), r)) \sum_{z_j \in a_r} d^2(z_i, z_j)$$

erreur de distorsion moyenne

Figure 4.8 – Les erreurs Kmeans et de distorsion moyenne

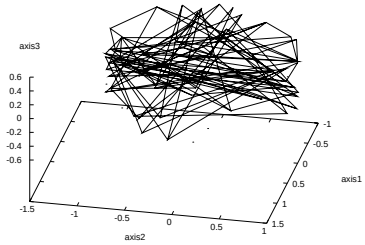


Figure 4.9 – La carte (21×3 neurones) initiale (initialisation "semi-aléatoire") et nuage des points

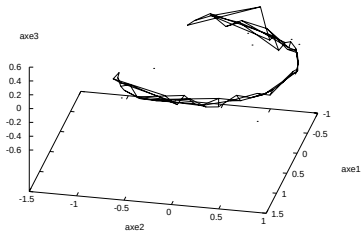


Figure 4.10 – La carte après 50 itérations

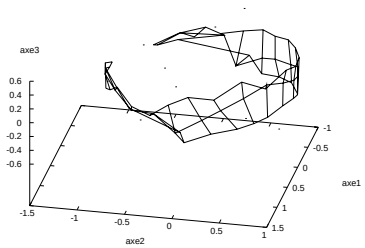


Figure 4.11 – La carte après 100 itérations

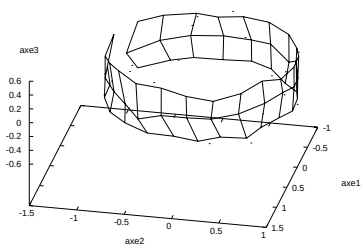


Figure 4.12 – La carte finale

4.3 Cartes topologiques sur tableaux de dissimilarités et cadre symbolique

Cette version des cartes topologiques sur tableaux de dissimilarités s'applique directement au cadre symbolique, dès lors que l'on peut associer une dissimilarité aux données manipulées. Pour le cas des assertions symboliques, on a vu au chapitre 1 (voir section 1.5.1) qu'il existe de très nombreux indices disponibles, y compris dans le cas des assertions munies de règles de dépendance entre les variables. Pour les données symboliques plus complexes que de simples assertions, il apparaît possible de définir des dissimilarités adaptées, et ainsi de leur appliquer la méthode.

On présente ici une application réalisée avec la méthode proposée appliquée aux assertions symboliques. Cette application concerne des températures minimales et maximales mensuelles observées dans 265 stations météorologiques chinoises [1]. Une représentation naturelle de la température "mensuelle" d'une station est l'intervalle constitué par la moyenne des minima journaliers et la moyenne des maxima journaliers observés dans cette station durant ce mois. Ayant les tableaux des températures minimales et maximales des 265 stations entre les années 1979 et 1988, nous avons constitué un tableau symbolique décrivant les 265 stations. En effet, chaque station est décrite par 12 variables (correspondant aux 12 mois de l'année) de type intervalles (voir tableau 4.1).

Station	Janvier	Fevrier	...	Novembre	Decembre
Abag Qi	[-24.9 ; -17]	[-22.3 ; -12.8]	...	[-16.4 ; -6.2]	[-24.7 ; -14.8]
⋮	⋮	⋮	⋮	⋮	⋮
Hailaer	[-28.6 ; -22.5]	[-25.5 ; -19.7]	...	[-17.4 ; -9.3]	[-25.5 ; -20.0]
⋮	⋮	⋮	⋮	⋮	⋮

Tableau 4.1 – Tableau symbolique décrivant les températures des 265 stations entre 1979 et 1988

Chaque intervalle est constitué par la moyenne des minima mensuels et la moyenne

[1] les données sont disponible à l'URL http ://dss.ucar.edu/datasets/ds578.5/data

des maxima mensuels durant les 10 ans. Dans ce qui suit, nous allons décrire les paramètres utilisés pour cette application (distance, dimensions de la carte, nombre d'itérations, etc). Le choix de ces paramètres est important pour le bon déroulement de l'algorithme. Nous allons ensuite décrire les résultats obtenus.

4.3.1 Distance de Hausdorff

Dans cette première application, nous avons utilisé la distance de Hausdorff déjà décrite au chapitre 1 (voir section 1.5.1) et dont la définition est la suivante :

$$d(Q,Q') = (\sum_{j=1}^{p}(\max\{|a_j - a'_j|, |b_j - b'_j|\})^2)^{\frac{1}{2}}$$

où $Q = (I_1, ..., I_p)$ et $Q' = (I'_1, ...I'_p)$ une paire d'éléments décrits par p intervalles et $I_j = [a_j, b_j]$.

Les paramètres utilisés pour le déroulement de notre algorithme sont les suivants :

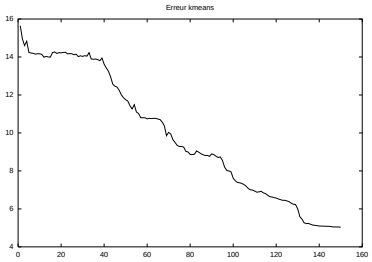
Paramètres	Valeurs
Dissimilarité	Hausdorff sur données intervalles
Ensemble d'apprentissage	265
Nombre d'itérations (N_{iter})	150
Nombre de neurones	$30 : 10 \times 3$
Initialisation	"semi-aléatoire"
cardinal individus référents : q	1

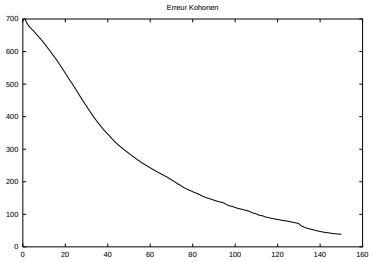
Sur la figure 4.13, nous présentons la courbe de l'erreur Kmeans et celle de distorsion moyenne.

Sur la figure 4.14, nous présentons sur le plan de l'AFTD le nuage des points et la carte initiale (initialisée "semi-aléatoirement"). La figure 4.15 présente la carte finale obtenue après apprentissage.

Les tableaux 4.2 et 4.3 représentent les descriptions des 30 individus référents de la classification obtenue.

Sur les figures 4.16 et 4.17, nous présentons les courbes allant de la classe 1×1 à la classe 10×3. Dans chaque case figurent les intervalles mensuels d'évolution des 30 individus référents. La classification obtenue respecte les températures observées. Nous





$$E_{kmeans} = \sum_{z_i \in \Omega} \sum_{z_j \in a_{f(z_i)}} d^2(z_i, z_j) \qquad E_{koho} = \sum_{z_i \in \Omega} \sum_{r \in C} K^T(\delta(f(z_i), r)) \sum_{z_j \in a_r} d^2(z_i, z_j)$$

erreur kmeans — erreur de distorsion moyenne

Figure 4.13 – Les erreurs Kmeans et de distorsion moyenne

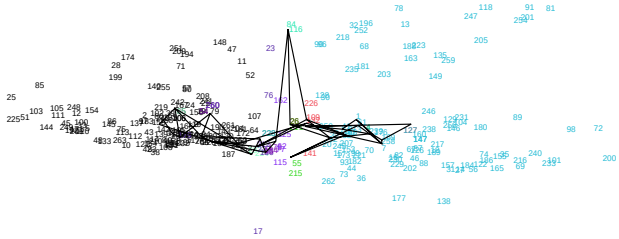


Figure 4.14 – Carte initiale et nuage des points sur le plan de l'AFTD

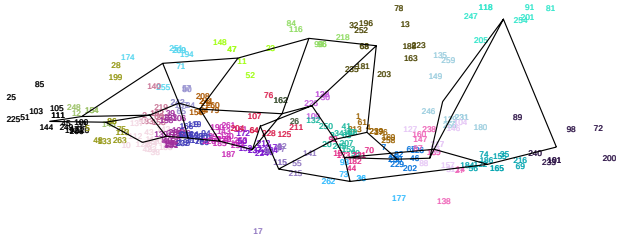


Figure 4.15 – Carte finale sur le plan de l'AFTD

		Mois					
Classe	individu référent	Janvier	Février	Mars	Avril	Mai	Juin
1*1	Shenzhen	[12.3 ; 17.2]	[12.6 ; 18.0]	[16.0 ; 21.2]	[20.8 ; 24.0]	[24.3 ; 27.0]	[26.5 ; 28.8]
2*1	XiaMen	[10.5 ; 14.2]	[10.4 ; 15.0]	[12.5 ; 16.0]	[17.1 ; 19.9]	[21.2 ; 23.7]	[24.4 ; 28.2]
3*1	Dali	[6.8 ; 8.8]	[8.4 ; 12.1]	[11.4 ; 14.2]	[14.3 ; 16.7]	[18.1 ; 20.9]	[19.4 ; 21.7]
4*1	Dehua Jiuxian-shan	[3.3 ; 6.7]	[2.9 ; 7.6]	[7.0 ; 11.1]	[10.5 ; 13.7]	[14.0 ; 16.6]	[15.9 ; 18.7]
5*1	Qamdo	[-4.3 ; -1.2]	[-2.7 ; 3.4]	[2.8 ; 5.7]	[6.3 ; 9.2]	[11.1 ; 15.1]	[13.9 ; 17.6]
6*1	Minxian	[-8.7 ; -4.3]	[-4.9 ; -1.4]	[-0.5 ; 2.6]	[4.9 ; 7.8]	[9.7 ; 12.2]	[12.7 ; 14.8]
7*1	Otog Qi	[-12.5 ; -7.2]	[-9.8 ; -3.5]	[-2.4 ; 2.8]	[7.1 ; 10.7]	[13.9 ; 17.5]	[19.5 ; 21.2]
8*1	Siping	[-17.0 ; -10.3]	[-11.7 ; -7.6]	[-4.3 ; 0.5]	[5.5 ; 10.7]	[15.3 ; 16.9]	[20.4 ; 22.5]
9*1	Sonid Youqi	[-19.1 ; -11.3]	[-15.3 ; -7.2]	[-8.0 ; -0.3]	[3.7 ; 9.5]	[13.4 ; 16.5]	[18.0 ; 21.6]
10*1	Huade	[-18.5 ; -12.8]	[-15.9 ; -9.1]	[-9.4 ; -2.1]	[1.2 ; 6.7]	[10.5 ; 13.7]	[14.9 ; 18.0]
1*2	Luodian	[7.1 ; 12.8]	[9.0 ; 15.6]	[13.0 ; 19.7]	[18.6 ; 22.2]	[22.6 ; 25.3]	[24.5 ; 26.6]
2*2	Neijiang	[5.4 ; 9.2]	[7.0 ; 11.6]	[9.6 ; 15.3]	[16.1 ; 19.1]	[20.0 ; 23.5]	[22.6 ; 25.8]
3*2	Yaan	[4.3 ; 8.5]	[5.7 ; 10.6]	[8.7 ; 13.3]	[14.9 ; 17.4]	[18.8 ; 22.6]	[21.7 ; 24.6]
4*2	Wanyuan	[1.9 ; 5.9]	[3.8 ; 8.1]	[7.0 ; 11.6]	[13.9 ; 16.0]	[17.5 ; 20.3]	[21.6 ; 24.0]
5*2	TianShui	[-3.5 ; 0.1]	[-0.9 ; 3.9]	[3.2 ; 8.5]	[10.9 ; 13.3]	[15.6 ; 18.3]	[19.3 ; 21.7]
6*2	Pingliang	[-6.2 ; -2.3]	[-4.0 ; 1.3]	[0.5 ; 6.1]	[9.2 ; 11.6]	[13.8 ; 16.7]	[17.7 ; 20.4]
7*2	YinChuan	[-9.9 ; -5.1]	[-7.2 ; -1.7]	[0.7 ; 5.1]	[10.0 ; 13.0]	[15.7 ; 19.1]	[20.8 ; 22.4]
8*2	Fuxin	[-13.3 ; -8.0]	[-8.5 ; -5.2]	[-1.9 ; 2.0]	[6.5 ; 12.3]	[16.2 ; 18.9]	[20.7 ; 23.0]
9*2	Jixi	[-19.8 ; -13.4]	[-15.2 ; -10.6]	[-7.4 ; -1.9]	[3.1 ; 8.4]	[11.8 ; 15.2]	[14.6 ; 20.4]
10*2	Nagqu	[-16.7 ; -10.5]	[-13.2 ; -6.6]	[-6.8 ; -3.1]	[-3.1 ; -0.2]	[2.3 ; 6.0]	[6.7 ; 9.9]
1*3	Guangchang	[3.5 ; 9.0]	[5.3 ; 10.8]	[9.0 ; 14.3]	[16.7 ; 19.1]	[21.3 ; 24.4]	[24.2 ; 27.4]
2*3	Yuanling	[3.2 ; 6.8]	[4.6 ; 9.6]	[7.6 ; 12.6]	[14.9 ; 17.2]	[19.6 ; 22.3]	[22.8 ; 25.7]
3*3	Wuhu	[1.2 ; 4.5]	[2.2 ; 7.0]	[6.8 ; 11.0]	[14.6 ; 16.4]	[20.1 ; 23.0]	[23.0 ; 26.5]
4*3	Haoxian	[-1.2 ; 1.4]	[0.9 ; 4.6]	[5.9 ; 10.0]	[13.5 ; 15.7]	[19.6 ; 22.5]	[23.5 ; 26.7]
5*3	Dezhou	[-4.6 ; -1.1]	[-1.7 ; 1.5]	[4.5 ; 8.6]	[11.8 ; 16.0]	[19.2 ; 22.1]	[24.2 ; 26.8]
6*3	YanTai	[-3.5 ; 0.7]	[-2.0 ; 1.5]	[3.1 ; 6.0]	[9.5 ; 13.8]	[16.1 ; 19.4]	[20.7 ; 24.3]
7*3	Bachu	[-7.5 ; -4.3]	[-3.3 ; 1.2]	[6.0 ; 10.1]	[15.2 ; 17.7]	[19.0 ; 22.9]	[22.5 ; 25.9]
8*3	Ejin	[-13.0 ; -8.6]	[-9.7 ; -2.6]	[-1.8 ; 3.7]	[8.7 ; 13.0]	[17.7 ; 21.4]	[20.6 ; 25.6]
9*3	Keshan	[-25.6 ; -19.5]	[-20.3 ; -13.9]	[-10.6 ; -3.9]	[1.4 ; 7.4]	[10.8 ; 15.0]	[15.3 ; 21.3]
10*3	Hailaer	[-28.6 ; -22.5]	[-25.5 ; -19.7]	[-17.1 ; -7.8]	[-1.7 ; 4.9]	[8.9 ; 13.0]	[14.5 ; 19.7]

Tableau 4.2 – Les descriptions des 30 individus référents finaux durant les six premiers mois de l'année

		Mois					
Classe	individu référent	Juillet	Août	Septembre	Octobre	Novembre	Décembre
1*1	Shenzhen	[27.7 ; 29.5]	[27.5 ; 29.0]	[26.3 ; 27.6]	[22.7 ; 25.4]	[18.7 ; 21.4]	[14.3 ; 16.8]
2*1	XiaMen	[26.5 ; 29.5]	[27.0 ; 28.6]	[24.6 ; 27.2]	[22.2 ; 24.4]	[17.2 ; 20.4]	[12.7 ; 16.0]
3*1	Dali	[19.4 ; 20.9]	[18.9 ; 20.2]	[16.8 ; 18.6]	[13.2 ; 17.1]	[10.2 ; 12.8]	[6.9 ; 10.0]
4*1	Dehua Jiuxian-shan	[18.4 ; 20.1]	[17.6 ; 20.0]	[15.1 ; 17.8]	[11.4 ; 15.0]	[7.2 ; 10.9]	[3.7 ; 6.7]
5*1	Qamdo	[14.8 ; 17.6]	[13.6 ; 16.7]	[11.1 ; 14.0]	[6.0 ; 11.1]	[1.0 ; 3.7]	[-3.0 ; 0.5]
6*1	Minxian	[15.0 ; 17.1]	[14.8 ; 16.0]	[10.3 ; 11.9]	[5.1 ; 8.1]	[-0.8 ; 2.3]	[-7.4 ; -3.1]
7*1	Otog Qi	[19.7 ; 23.7]	[18.7 ; 21.6]	[12.4 ; 16.0]	[4.3 ; 10.2]	[-4.2 ; 1.2]	[-12.5 ; -6.8]
8*1	Siping	[22.1 ; 24.8]	[20.8 ; 23.7]	[14.4 ; 17.3]	[5.9 ; 9.5]	[-6.4 ; 1.0]	[-13.9 ; -6.8]
9*1	Sonid Youqi	[20.3 ; 24.5]	[18.4 ; 20.5]	[11.8 ; 15.0]	[2.7 ; 8.4]	[-8.9 ; -0.2]	[-16.0 ; -8.9]
10*1	Huade	[17.2 ; 20.8]	[16.0 ; 17.2]	[9.2 ; 12.3]	[0.6 ; 6.3]	[-10.8 ; -2.5]	[-16.5 ; -10.7]
1*2	Luodian	[25.2 ; 28.5]	[25.8 ; 28.2]	[23.5 ; 25.4]	[18.7 ; 22.3]	[14.2 ; 18.7]	[9.8 ; 14.2]
2*2	Neijiang	[24.9 ; 27.2]	[24.7 ; 28.1]	[20.7 ; 23.0]	[15.7 ; 19.8]	[11.7 ; 15.3]	[6.7 ; 10.4]
3*2	Yaan	[24.1 ; 25.5]	[23.7 ; 25.7]	[19.5 ; 21.8]	[14.9 ; 18.2]	[10.6 ; 14.2]	[5.4 ; 9.3]
4*2	Wanyuan	[22.8 ; 25.6]	[23.0 ; 26.0]	[18.4 ; 20.7]	[12.7 ; 16.6]	[8.9 ; 11.8]	[3.6 ; 7.6]
5*2	TianShui	[20.9 ; 22.9]	[21.0 ; 22.7]	[14.9 ; 18.0]	[8.9 ; 12.9]	[3.2 ; 6.4]	[-3.3 ; 1.2]
6*2	Pingliang	[19.6 ; 21.3]	[18.7 ; 20.7]	[12.8 ; 15.9]	[6.5 ; 10.7]	[0.9 ; 4.3]	[-6.8 ; -1.0]
7*2	YinChuan	[21.5 ; 24.7]	[20.6 ; 23.1]	[14.1 ; 17.3]	[6.5 ; 11.7]	[-0.9 ; 3.0]	[-10.4 ; -4.6]
8*2	Fuxin	[22.7 ; 25.9]	[22.1 ; 24.6]	[15.7 ; 19.2]	[7.2 ; 10.8]	[-4.4 ; 2.5]	[-10.3 ; -5.4]
9*2	Jixi	[19.6 ; 22.5]	[18.9 ; 22.7]	[13.0 ; 15.7]	[4.0 ; 6.8]	[-7.7 ; -3.1]	[-16.1 ; -10.3]
10*2	Nagqu	[7.8 ; 9.7]	[7.3 ; 9.5]	[3.7 ; 6.2]	[-2.5 ; 2.2]	[-10.1 ; -6.0]	[-13.8 ; -9.2]
1*3	Guangchang	[28.3 ; 30.6]	[27.2 ; 29.8]	[23.4 ; 26.4]	[18.5 ; 21.3]	[12.0 ; 15.8]	[6.4 ; 11.4]
2*3	Yuanling	[26.4 ; 29.3]	[25.1 ; 29.8]	[21.6 ; 24.5]	[15.8 ; 19.4]	[10.5 ; 15.0]	[4.7 ; 9.3]
3*3	Wuhu	[26.3 ; 30.2]	[24.8 ; 28.7]	[21.8 ; 24.7]	[15.5 ; 19.4]	[9.5 ; 13.0]	[3.2 ; 6.6]
4*3	Haoxian	[25.8 ; 28.7]	[24.7 ; 26.9]	[19.8 ; 22.7]	[13.3 ; 17.2]	[6.8 ; 10.7]	[0.4 ; 3.5]
5*3	Dezhou	[25.8 ; 27.6]	[24.4 ; 26.3]	[18.6 ; 22.2]	[12.7 ; 16.4]	[4.2 ; 8.7]	[-2.5 ; 1.0]
6*3	YanTai	[23.9 ; 26.2]	[23.4 ; 25.5]	[20.3 ; 22.9]	[14.0 ; 17.0]	[5.5 ; 10.8]	[-0.8 ; 3.1]
7*3	Bachu	[25.4 ; 27.8]	[22.9 ; 27.4]	[18.8 ; 21.5]	[10.3 ; 13.5]	[1.3 ; 4.9]	[-7.8 ; -3.2]
8*3	Ejin Qi	[24.0 ; 27.9]	[23.0 ; 25.7]	[16.5 ; 18.6]	[4.8 ; 10.5]	[-5.9 ; 1.3]	[-14.4 ; -7.4]
9*3	Keshan	[20.3 ; 23.4]	[17.6 ; 21.8]	[11.2 ; 13.9]	[1.6 ; 4.6]	[-12.7 ; -6.9]	[-23.4 ; -16.1]
10*3	Hailaer	[18.0 ; 21.2]	[15.4 ; 19.0]	[8.6 ; 10.9]	[-1.3 ; 2.5]	[-17.4 ; -9.3]	[-25.5 ; -20.0]

Tableau 4.3 – Les descriptions des 30 individus référents finaux durant les six derniers mois de l'année

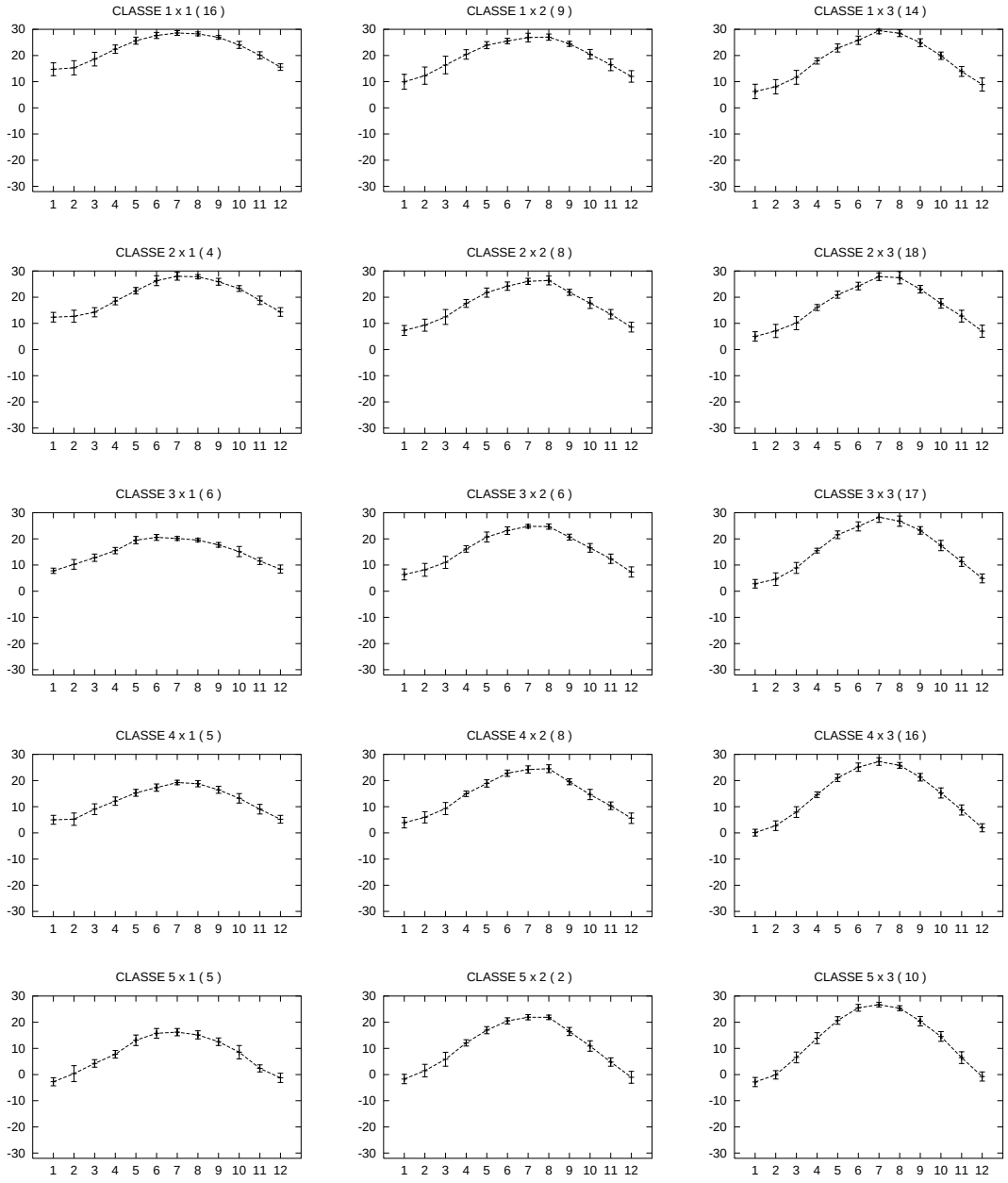


Figure 4.16 – Les 15 premiers individus référents de la carte (10×3). Pour chaque neurone on représente les intervalles mensuels d'évolution de la température annuelle

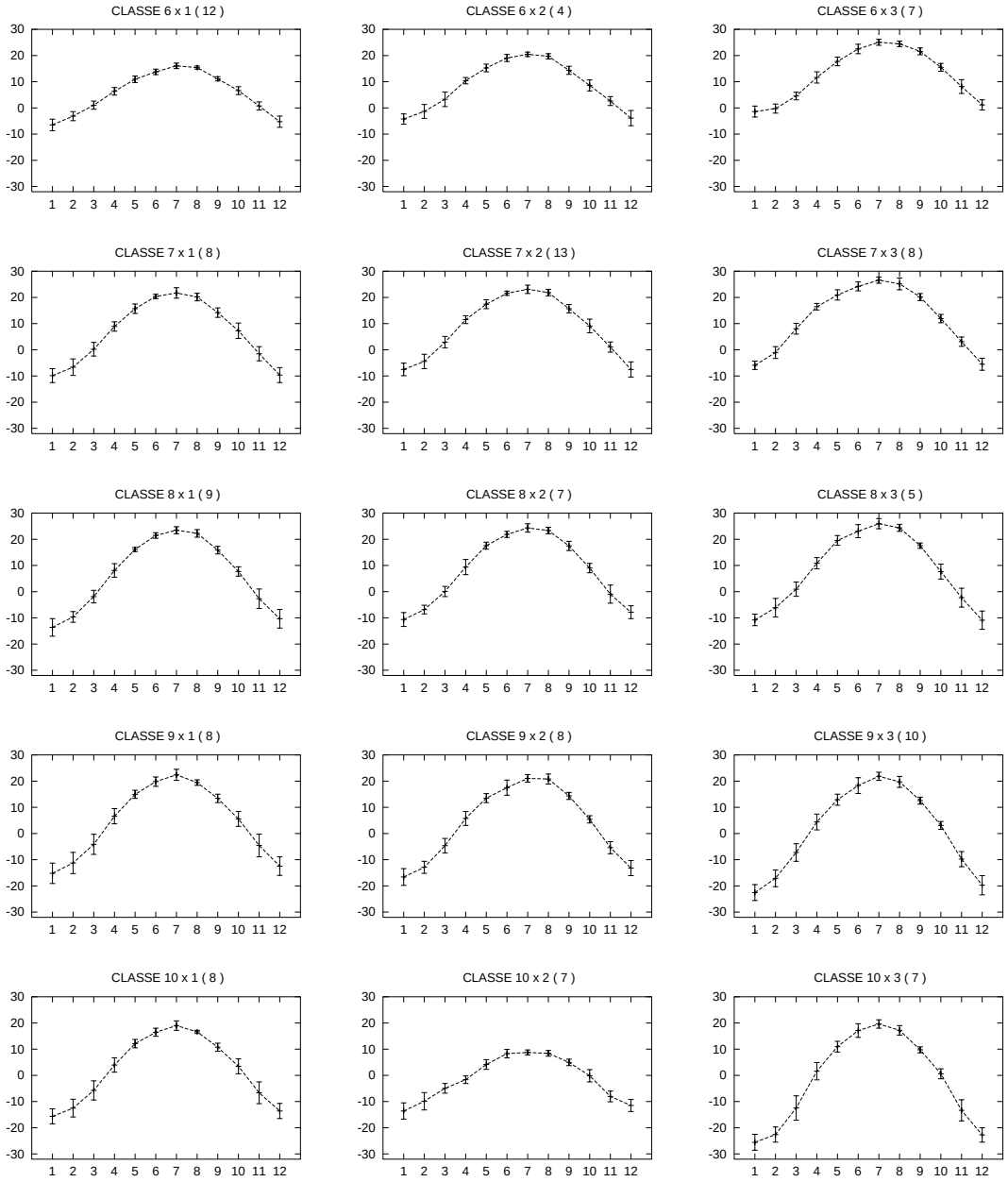


Figure 4.17 – Les 15 individus référents restant de la carte (10×3). Pour chaque neurone on représente les intervalles mensuels d'évolution de la température annuelle

remarquons que la Chine se divise en deux grandes régions climatiques. Les classes 1×1, 2×1, 3×1, 4×1, 5×1, 1×2, 2×2, 3×2, 4×2, 5×2, 1×3, 2×3, 3×3, 4×3, 5×3, se trouvent au Sud et Sud-Est de la Chine et représentent un climat chaud. Les classes 6×1, 7×1, 8×1, 9×1, 10×1, 6×2, 7×2, 8×2, 9×2, 10×2, 6×3, 7×3, 8×3, 9×3, 10×3, se trouvent au Nord et à l'Ouest de la Chine et représentent un climat froid. Ces régions sont caractérisées par des écarts de températures très importants.

Nous présentons maintenant sur la figure 4.18, la carte de la Chine avec les différentes classes obtenues. Chaque classe est représentée par ses individus rattachés à l'individu référent. Cette figure nous permet de déduire que la température varie de façon plus importante en fonction de la latitude qu'en fonction de la longitude.

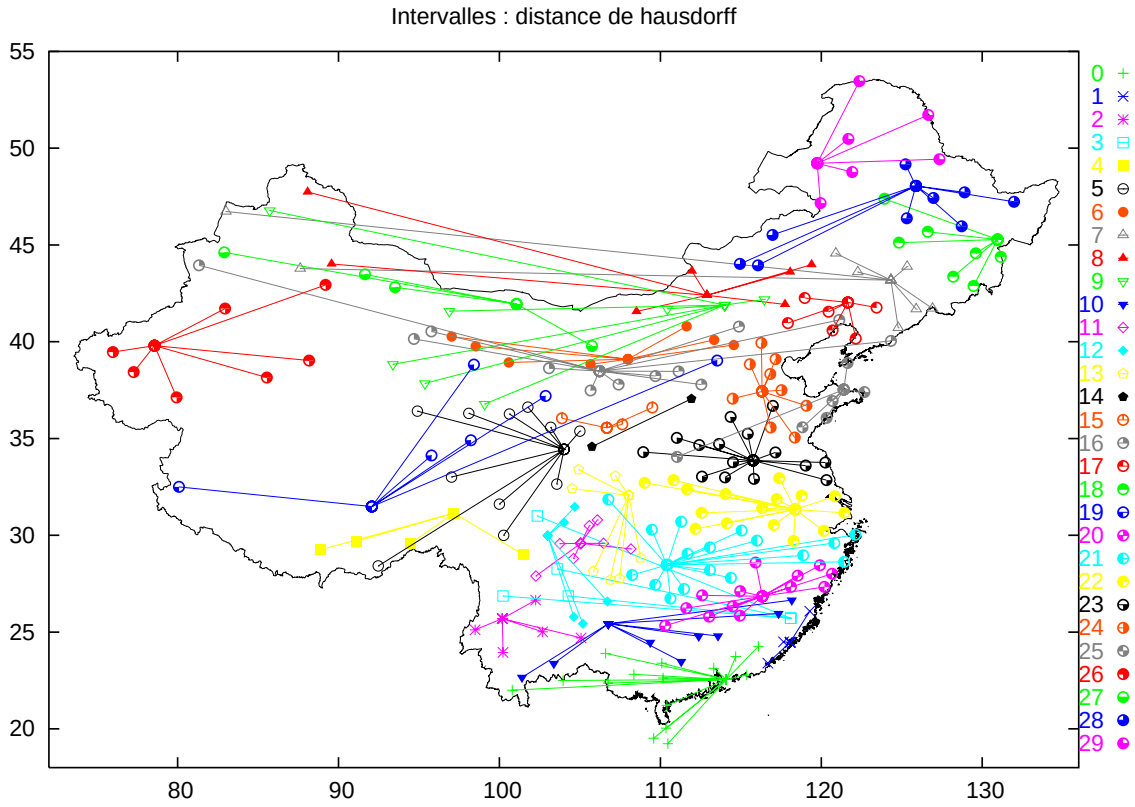


Figure 4.18 – La carte de la Chine et la classification obtenue avec la distance de Hausdorff sur intervalles

4.3.2 Distance euclidienne

La distance utilisée pour cette application est la distance euclidienne, déjà décrite au chapitre 1 (voir section 1.5.1) et dont la formule est la suivante :

$$d(Q, Q') = 1/4\|(a - a') + (b - b')\|^2$$

où $Q = (I_1, ..., I_p)$ et $Q' = (I'_1, ...I'_p)$ une paire d'éléments décrits par p intervalles et $I_j = [a_j, b_j]$.

Les paramètres utilisés pour le déroulement de notre algorithme sont les suivants :

Paramètres	valeurs
Dissimilarité	Euclidienne sur données intervalles
Ensemble d'apprentissage	265
Nombre d'itérations (N_{iter})	150
Nombre de neurones	$30 : 10 \times 3$
Initialisation	"semi-aléatoire"
cardinal individus référents : q	1

Sur la figure 4.19, nous présentons sur le plan de l'AFTD, le nuage des points et la carte initiale (initialisée "semi-aléatoirement"). Sur la figure 4.20, nous présentons la courbe de l'erreur Kmeans et celle de distorsion moyenne. La figure 4.21 présente la carte finale obtenue après apprentissage.

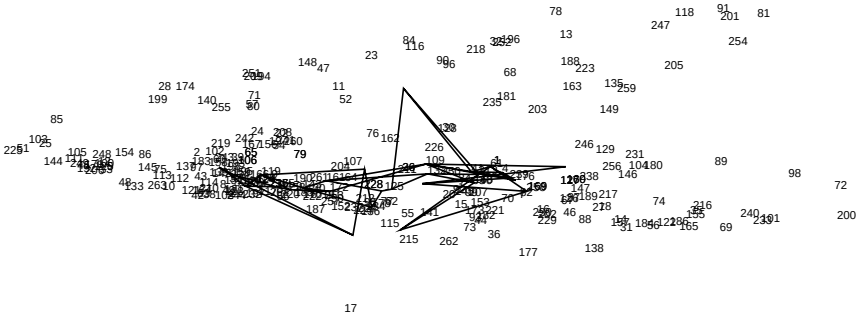
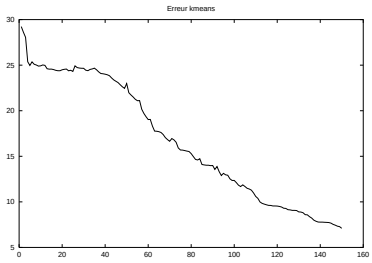
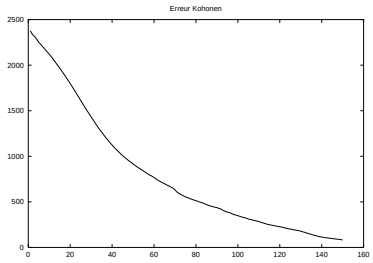


Figure 4.19 – Carte initiale et nuage des points sur le plan de l'AFTD

Sur les figures 4.22 et 4.23, nous présentons les courbes allant de la classe 1×1 à la classe 10×3. Dans chaque case figure les intervalles mensuels d'évolution de la température annuelle de l'individu référent du neurone. Nous remarquons que les neurones





$$E_{kmeans} = \sum_{z_i \in \Omega} \sum_{z_j \in a_{f(z_i)}} d^2(z_i, z_j) \qquad E_{koho} = \sum_{z_i \in \Omega} \sum_{r \in C} K^T(\delta(f(z_i), r)) \sum_{z_j \in a_r} d^2(z_i, z_j)$$

erreur kmeans — erreur de distorsion moyenne

Figure 4.20 – Les erreurs Kmeans et de distorsion moyenne

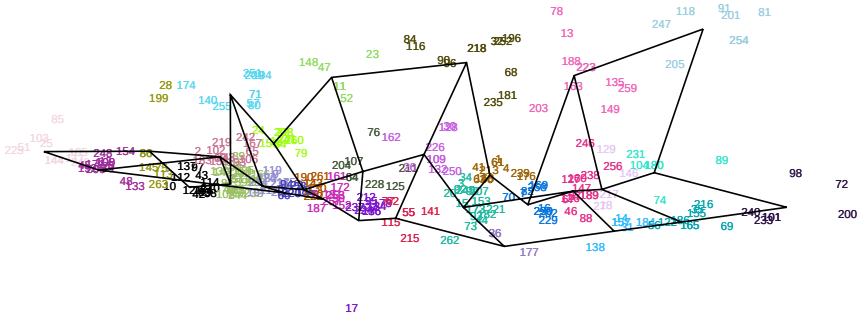


Figure 4.21 – Carte finale sur plan de l'AFTD

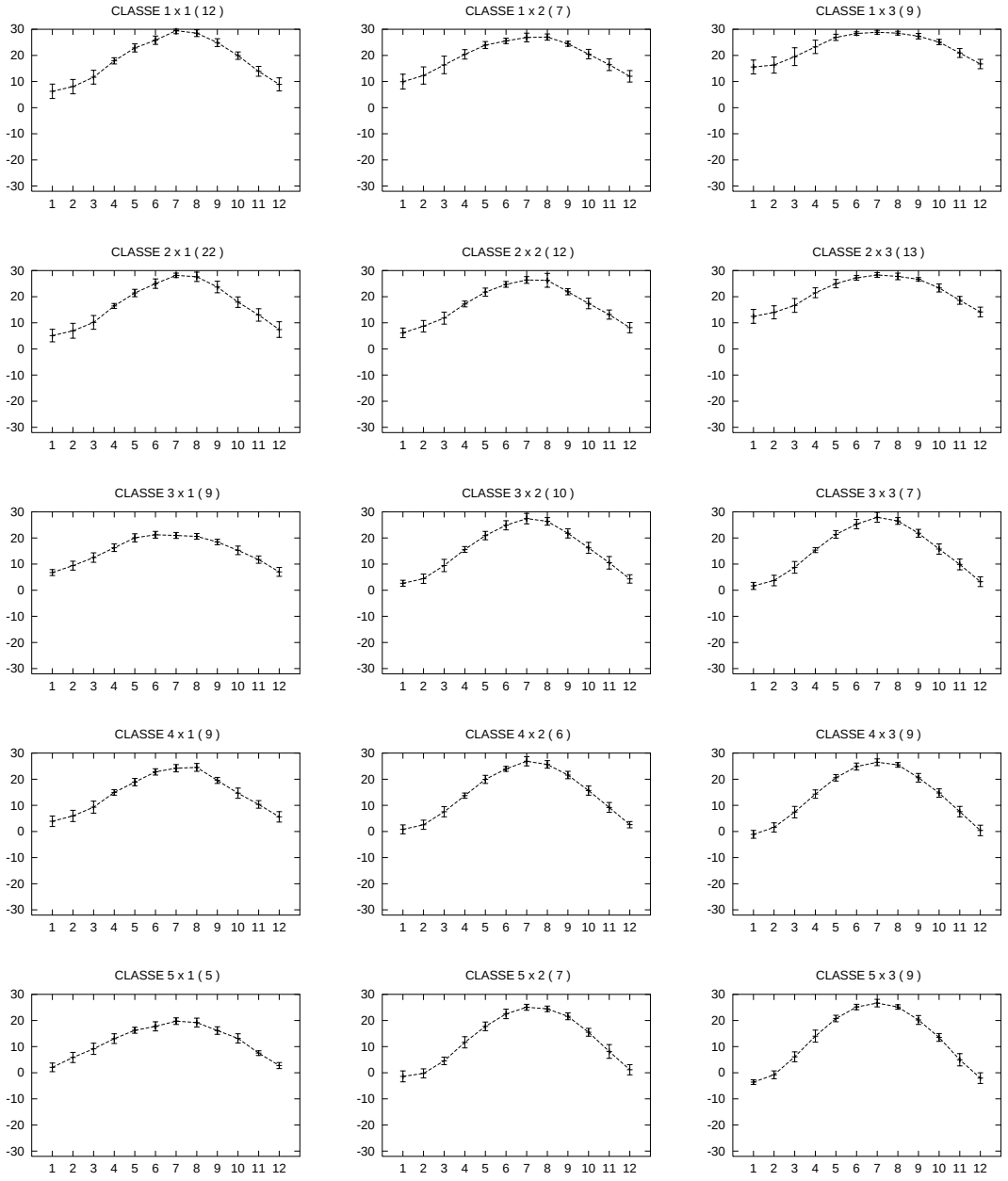


Figure 4.22 – Les 15 premiers individus référents de la carte (10×3). Pour chaque neurone on représente les intervalles mensuels d'évolution de la température annuelle

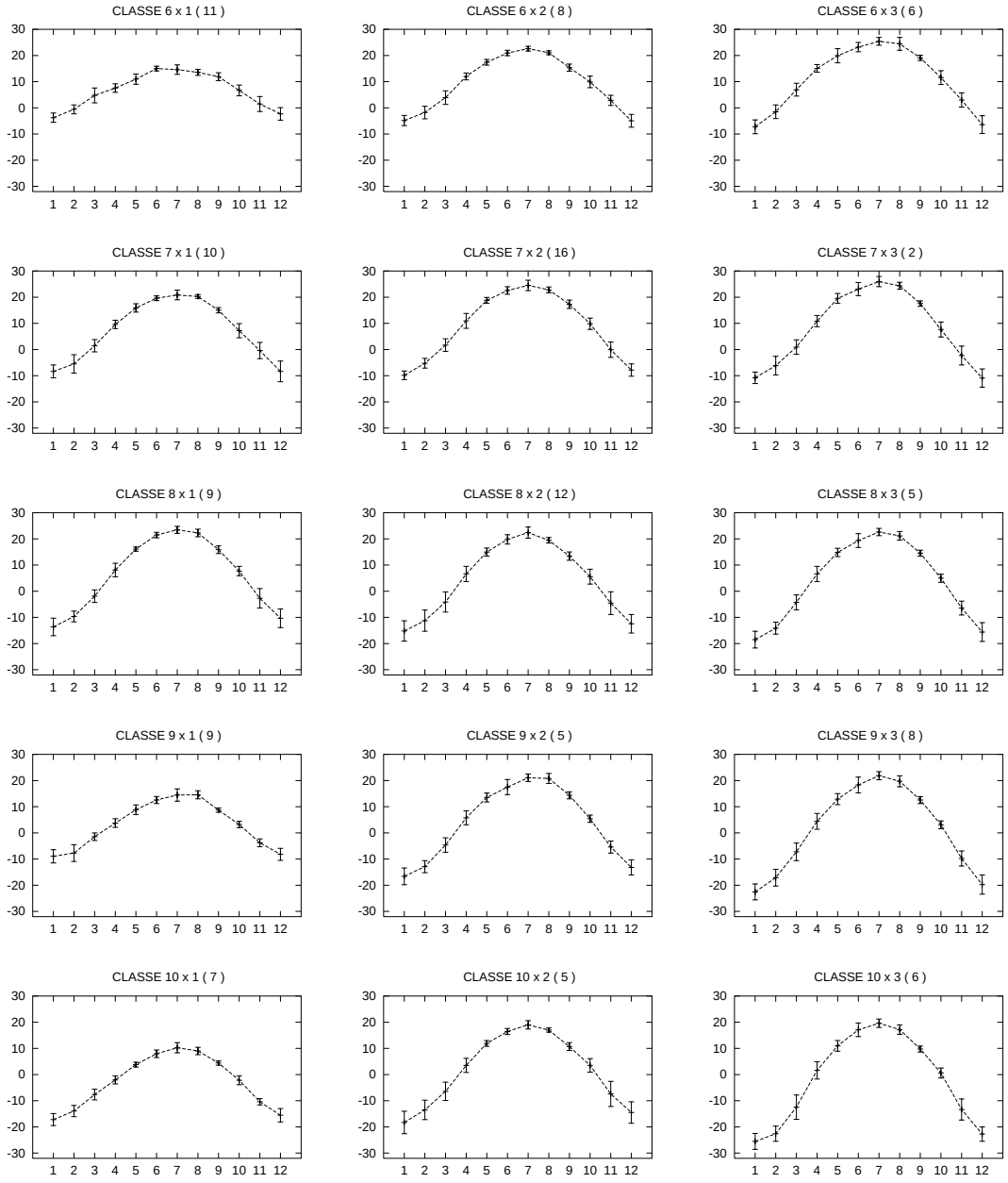


Figure 4.23 – Les 15 individus référents restant de la carte (10×3). Pour chaque neurone on représente les intervalles mensuels d'évolution de la température annuelle

voisins présentent une évolution de la température similaire avec des différences pour chaque neurone. Les courbes allant de la classe 1×1 à la classe 5×1, de la classe 1×2 à la classe 4×2 et de la classe 1×3 à la classe 4×3, représentent un climat plus chaud que le reste de la carte. Ces courbes correspondent à la partie Sud de la Chine. On voit bien que sur ces courbes les fins et débuts d'année sont assez chauds (températures au dessus de zéro), alors que les courbes allant de la classe 6×1 à la classe 10×1, de la classe 5×2 à la classe 10×2 et de la classe 5×3 à la classe 10×3, représentent des climats plutôt froids et correspondent à la partie Nord de la Chine.

Nous présentons maintenant sur la figure 4.24, la carte de la Chine avec les différentes classes obtenues. Chaque classe est représentée par ses individus rattachés à l'individu référent.

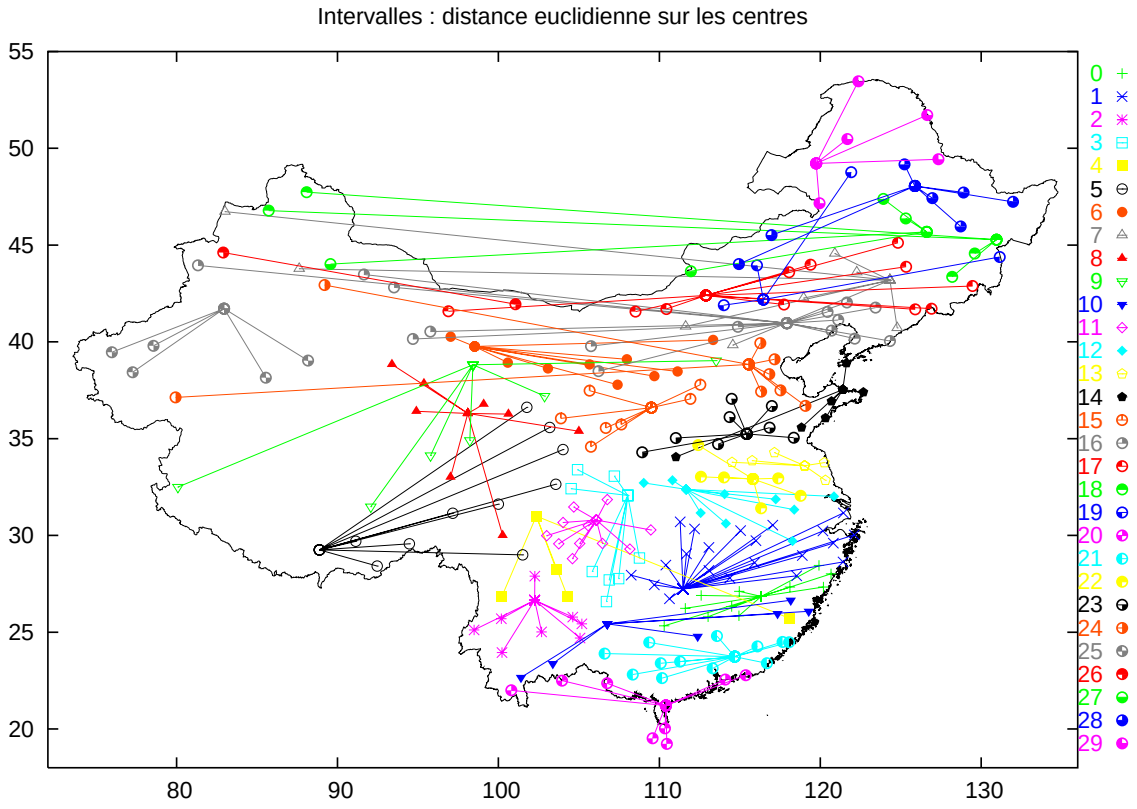


Figure 4.24 – La carte de la Chine et la classification obtenue avec la distance Euclidienne sur intervalle

4.3.3 Résultats et discussion

Nous avons appliqué la méthode avec d'autres mesures de dissimilarité, entre autre la distance de type sommet décrite au chapitre 1 (voir section 1.5.1). Comme nous disposons aussi des températures moyennes des stations (qui rappelons le n'est pas une représentation symboliques des températures), nous avons appliqué l'algorithme des cartes topologiques. Pour ce faire, nous avons calculé la matrice de distance euclidienne sur ces moyennes. Afin de pouvoir comparer tous les résultats obtenus avec ces différentes représentations et métriques, nous avons calculé les distorsions en longitude et en latitude des différentes classifications. La distorsion est définie comme l'erreur quadratique moyenne entre la station prototype et les stations qui lui sont affectées. Donc, la distorsion en longitude est définie ainsi :

$$(D_{long})^2 = \sum_{c \in C} \sum_{z_i \in c} \frac{1}{|c|} ||Lo_{z_i} - Lo_{f(z_i)}||^2$$

où $|c|$ est le cardinal de la classe c, $||Lo_{z_i} - Lo_{f(z_i)}||$ est la distance en longitude entre la station z_i et son prototype $f(z_i)$.
La distorsion en latitude est définie ainsi :

$$(D_{lati})^2 = \sum_{c \in C} \sum_{z_i \in c} \frac{1}{|c|} ||La_{z_i} - La_{f(z_i)}||^2$$

où $||La_{z_i} - La_{f(z_i)}||^2$ est la distance en latitude entre la station z_i et son prototype $f(z_i)$.

Dans le tableau 4.4, nous présentons les distorsions en latitude et en longitudes des différentes classifications obtenues avec les différentes métriques :

type des données	métrique utilisée	distorsion en longitude	distorsion en latitude
Intervalles	distance euclidienne	9.250688	1.993213
Intervalles	distance type sommet	8.625175	2.165838
Moyennes	distance euclidienne	7.656033	1.936692
Intervalles	distance de Hausdorff	7.38314	1.911461

Tableau 4.4 – Les valeurs des distorsions en longitude et en latitude pour les différentes métriques

Comme on le voit au tableau 4.4, la classification obtenue avec la métrique de Hausdorff induit les plus petites erreurs en latitude et en longitude. Ceci permet de conclure que les résultats obtenus avec la métrique de Hausdorff respectent au mieux l'aspect géographique du problème.

4.4 Conclusion

Au début de ce chapitre, nous avons proposé une adaptation de l'algorithme des cartes topologiques à des tableaux de dissimilarités. Cette adaptation est basée sur la version batch de l'algorithme initial, et permet de traiter aussi bien des données classiques que des données complexes. L'ordre topologique reste, comme dans l'algorithme initial, très sensible à l'ensemble des paramètres qui interviennent dans l'algorithme. Il n'existe pas de loi permettant de s'assurer de cet ordre.
Afin d'étudier le comportement de la méthode proposée, nous avons dans un premier temps appliqué notre algorithme à des données simulées. Les résultats obtenus son satisfaisants : quantification correcte de l'espace, bonne conservation de la topologie des données. Dans un deuxième temps, nous nous sommes intéressés à des données issues du domaine de la météorologie. Le problème traité s'est révélé très intéressant, car il a permis de comparer l'approche symbolique (modélisation par intervalles dans notre cas) à l'approche classique (utilisation de simples moyennes). La représentation par intervalles est pour cet exemple avantageuse, car elle permet de représenter de manière naturelle la variabilité associée aux données. Finalement, le traitement de données par tableaux de dissimilarités se révèle être une méthode souple, car elle permet facilement d'explorer différentes métriques (Hausdorff, euclidienne, sommet, etc).

Chapitre 5

Applications sur des données ayant une structure complexe

5.1 Introduction

Au chapitre 4, nous avons appliqué la méthode des cartes topologiques sur tableaux de dissimilarités à des données classiques quantitatives et aussi à des données symboliques. Rappelons que la méthode proposée dans nos travaux et décrite au chapitre 4, permet de traiter aussi bien les données classiques que les données complexes, car seule la définition d'une mesure de dissimilarité adéquate est nécessaire à la mise en œuvre de la méthode. Dans ce chapitre, nous allons présenter quelques applications de la méthode proposée sur des données fonctionnelles (des données de spectrométrie [RCG03] [GCGR04]) et des données Web.

5.2 Application aux données fonctionnelles : données de spectrométrie

Les données spectrométriques utilisées ont pour objectif le classement des échantillons de viandes selon une propriété physico-chimique qui n'est pas accessible directement et demande donc une analyse spécifique. Les données utilisées ont été obtenues grâce à un *Tecator Infratec food and feed Analyzer*[1] travaillant dans le proche infrarouge avec des

[1] les données sont disponible à l'URL http ://lib.stat.cmu.edu/datasets/tecator

longueurs d'onde comprises entre 850 et 1050 nanomètres (nm). La mesure est effectuée par transmission à travers un échantillon de viande finement hachée qui est ensuite analysé par un procédé chimique pour déterminer son taux de graisse. Les spectres obtenus correspondent à l'absorbance ($-log_{10}$ de la transmitance mesurée par l'appareil) pour 100 longueurs d'onde régulièrement réparties entre 850 et 1050 nm. À chaque spectre est associé une description de l'échantillon de viande : le pourcentage de graisse mais aussi de protéines et d'eau contenus dans l'échantillon. Le problème est alors de discriminer les spectres afin d'éviter une analyse chimique coûteuse et longue.

Le but de notre travail est d'obtenir une classification conforme au taux de graisse, afin de valider l'algorithme des cartes toplogiques sur tableau de dissimilarités.

5.2.1 Expériences avec la métrique euclidienne et une semi-métrique

Dans ce qui suit, nous allons présenter les résultats obtenus par notre méthode avec une distance euclidienne et une semi-métrique basée sur la dérivée seconde [GCGR04].

Les deux expériences ont été réalisées avec les mêmes paramètres, à savoir :

Paramètres	Valeurs
Ensemble d'apprentissage	215 spectres
Nombre d'itérations (N_{iter})	500
Nombre de neurones	16 : 8 * 2
Initialisation	"semi-aléatoire"
cardinal individus référents : q	1

5.2.1.1 Expérience avec la métrique euclidienne

Sur la figure 5.1, nous présentons sur le plan de l'AFTD (Analyse Factorielle sur Tableau de Distance) le nuage des points et la carte initiale (initialisée "semi-aléatoirement").

Sur la figure 5.2, nous présentons la courbe de l'erreur Kmeans et celle de la distorsion moyenne.

La figure 5.3 présente la carte finale obtenue après apprentissage.

La figure 5.4 représente une carte de (8×2). Chaque carré (est associé à une classe) représente une intensité calculée selon la moyenne des taux de graisse des spectres ap-

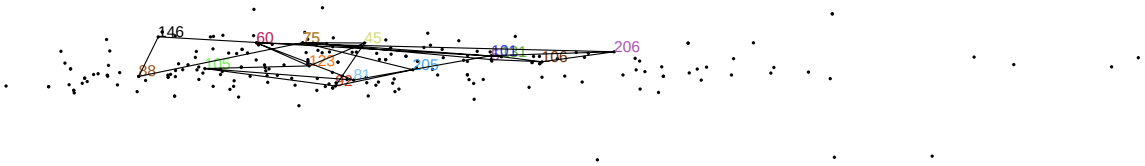
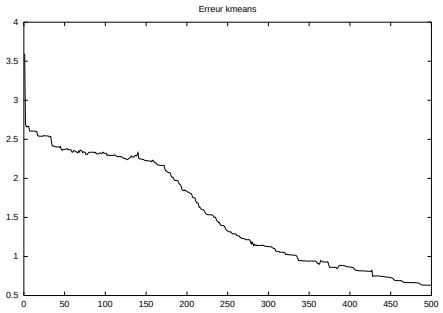
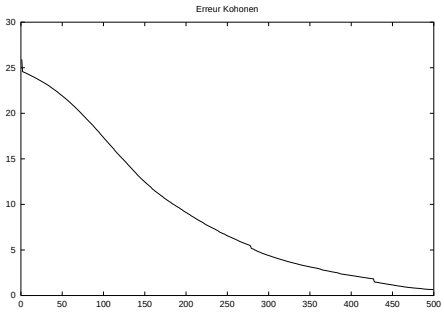


Figure 5.1 – Carte initiale et nuage des points sur le plan de l'AFTD





$$E_{kmeans} = \sum_{z_i \in \Omega} \sum_{z_j \in a_{f(z_i)}} d^2(z_i, z_j) \qquad E_{koho} = \sum_{z_i \in \Omega} \sum_{r \in C} K^T(\delta(f(z_i), r)) \sum_{z_j \in a_r} d^2(z_i, z_j)$$

erreur Kmeans erreur de distorsion moyenne

Figure 5.2 – Les erreurs Kmeans et de distorsion moyenne

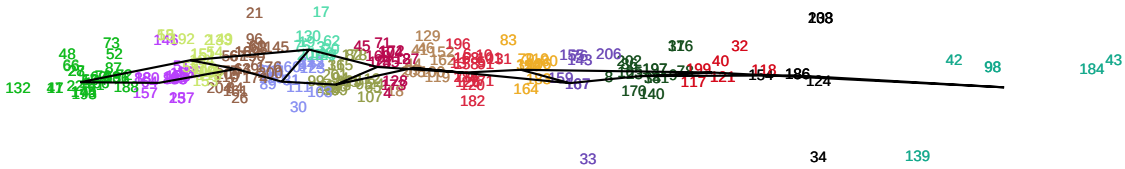


Figure 5.3 – Carte finale sur plan de l'AFTD

partenant à la classe (noir pour un taux de graisse faible et blanc pour un taux de graisse élevé).

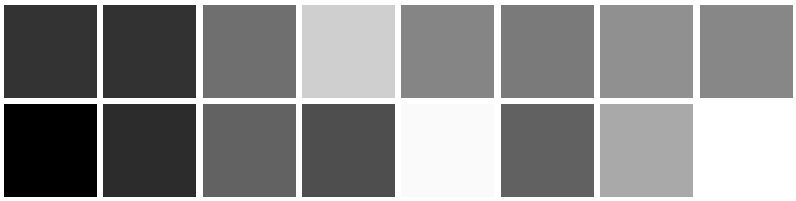


Figure 5.4 – La moyenne du taux de graisse pour chaque neurone (distance euclidienne)

Bien que la classification obtenue semble plutôt respecter la variable taux de graisse (taux de graisse faible à gauche et un taux de graisse élevé à droite), le résultat n'est pas tout à fait satisfaisant. En effet, comme on peut le voir sur la figure 5.4, il existe une classe avec un taux de graisse élevé entre deux classes à taux de graisse faible.

5.2.1.2 Expérience avec la semi-métrique basée sur la dérivée seconde

Nous présentons maintenant les résultats obtenus avec les mêmes données mais sur une semi-métrique basée sur la dérivée seconde des spectres :

$$||f||^2 = \int (f^{(2)}(t))^2 dt$$

Où $f^{(2)}$ est la dérivée seconde de f. Pour ces données, Ferraty et Vieu [FV03] indiquent que la dérivée seconde des spectres est généralement plus informative que les spectres eux-mêmes. Afin d'utiliser cette approche fonctionnelle, on dérive chaque spectre par un opérateur de différences finis. Finalement, on remplace le calcul exact des intégrales par l'évaluation de moyennes empiriques.

Sur la figure 5.5, nous présentons la carte initiale (initialisée "semi-aléatoirement").

Sur la figure 5.6, nous présentons la courbe de l'erreur Kmeans et celle de la distorsion moyenne.

La figure 5.7 présente la carte finale obtenue après apprentissage.

Sur la figure 5.8, on peut voir que la classification obtenue respecte parfaitement la variable taux de graisse. Ces deux exemples prouvent que notre algorithme dépend

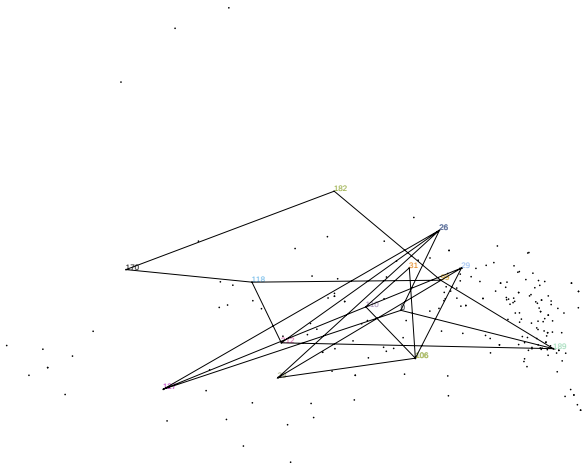
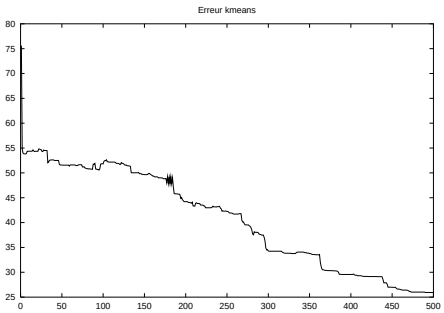
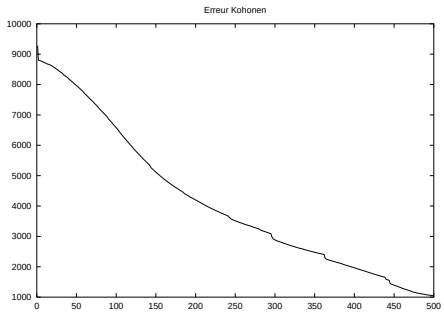


Figure 5.5 – Carte initiale sur le plan de l'AFTD





$$E_{kmeans} = \sum_{z_i \in \Omega} \sum_{z_j \in a_{f(z_i)}} d^2(z_i, z_j) \qquad E_{koho} = \sum_{z_i \in \Omega} \sum_{r \in C} K^T(\delta(f(z_i), r)) \sum_{z_j \in a_r} d^2(z_i, z_j)$$

erreur Kmeans erreur de distorsion moyenne

Figure 5.6 – Les erreurs Kmeans et de distorsion moyenne

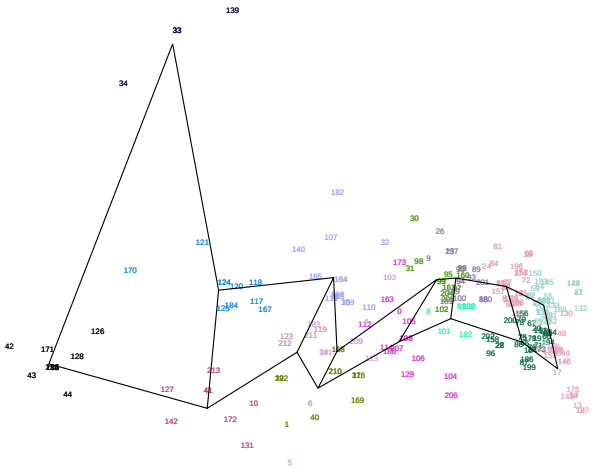


Figure 5.7 – Carte finale sur plan de l'AFTD

fortement de la métrique choisie. Avec une métrique adaptée, l'algorithme donne de bons résultats dans le sens où la topologie de la carte respecte la variable taux de graisse. L'application de l'algorithme classique des cartes topologiques sur ces données est aussi possible. Mais le but de cette application des données en spectrométrie est de montrer la validité de cette approche et aussi sa flexibilité.

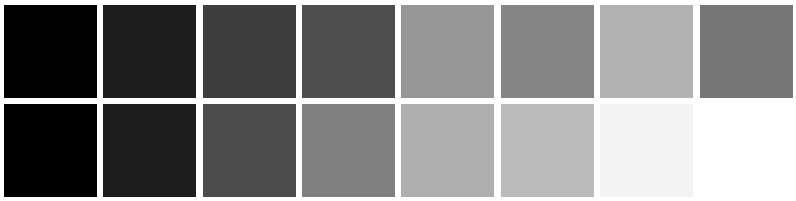


Figure 5.8 – La moyenne du taux de graisse pour chaque neurone (métrique basée sur la dérivée seconde)

5.3 Application aux données Web : analyse des fichiers Logs du site Web de l'INRIA

5.3.1 Le Web Mining

Le Web Mining comporte deux aspects complémentaires, le Web Content Mining et le Usage Mining. Le Web content Mining (le contenu du document) est considéré comme

une extension des méthodes du Text Mining à des documents contenant des structures de liens hypertextuels. Le Usage Mining (comportement des utilisateurs) est considéré comme des méthodes de traitement, de détection et d'analyse de comportement d'utilisateurs sur les sites Web.

L'objectif de mon projet de recherche (AxIS) est la conception, l'analyse et l'amélioration de systèmes d'informations dirigées par les usages. Bien qu'à court terme le projet s'oriente principalement sur les sites ou services Web, nous nous plaçons dans une optique globale de conception et d'évaluation de systèmes d'informations adaptatifs basés sur les standards du W3C. Une des applications de l'équipe est l'étude de l'activité du site Web de l'INRIA. Le premier objectif est la perception de l'activité de l'INRIA par les internautes via celle de son site. Le deuxième objectif est d'apporter des éléments significatifs en vue de l'amélioration de la qualité du site, et de la réponse qu'il apporte aux besoins des utilisateurs.

Une étude de ce type relève du "Web Usage Mining" dont les objectifs sont :

- connaître le profil des utilisateurs du site ;
- connaître les centres d'intérêt des utilisateurs, pour adapter le contenu du site ;
- connaître le comportement des utilisateurs, pour adapter l'ergonomie du site ;
- mesurer les performances du site pour améliorer son architecture.

Avant de commencer à détailler les étapes de notre application, nous présentons les principales notions définies dans [TT03] à partir de celles du W3C sur la terminologie de caractérisation du Web.

Définition 5.3.1. Ressource- d'après la spécification de W3C, une ressource R peut être "tout objet ayant une identité" [LN99]. Comme exemple de ressources, nous pouvons citer : un fichier HTML, une image ou un service Web.

Définition 5.3.2. Ressource Web- une ressource accessible par une version du protocole HTTP ou un protocole similaire (exp : HTTP-NG).

Définition 5.3.3. Serveur Web- un serveur qui donne accès à des ressources Web.

Définition 5.3.4. Requête Web- une requête pour une ressource Web, faite par un client (navigateur Web) à un serveur Web.

Définition 5.3.5. Page Web- ensemble des informations, consistant en une (ou plusieurs) ressource(s) Web identifiée(s) par un seul URI (Uniform Resource Identifier : "une chaine de caratères utilisée pour identifier une ressource abstraite ou physique" [LN99]). Exemple : un fichier HTML, un fichier image et un applet Java accessibles par un seul URI constituent une page Web.

Définition 5.3.6. Navigateur Web(Browser)- logiciel de type client chargé d'afficher des pages à l'utilisateur et de faire des requêtes HTTP au serveur Web.

Définition 5.3.7. Utilisateur- personne qui utilise un navigateur Web.

Définition 5.3.8. Navigation(s)- ensemble de clics utilisateurs sur un seul serveur Web (ou plusieurs lorsque on a fusionné leurs fichiers Logs) pendant une session utilisateur, qu'on appelle aussi *visite(s)*. Les clics de l'utilisateur peuvent être décomposés dans plusieurs navigations en calculant la distance temporelle entre deux requêtes HTTP consécutives : si cette distance excède un certain seuil, une nouvelle navigation commence.

Définition 5.3.9. Session utilisateur- un ensemble délimité de clics utilisateurs sur un (ou plusieurs) serveur(s) Web.

Définition 5.3.10. Vitesse de navigation (BS : Browsing Speed)-

$$BS(N_{ij}) = \frac{|N_{ij}|}{(\text{Durée de la navigation en seconde})}$$

avec $|N_{ij}|$ le nombre de pages visitées durant la navigation.

5.3.2 Les fichiers Logs et leur prétraitement

Cette section est un résumé des travaux de thèse de D. Tanasa sur les prétraitement de Logs Web [TT03], [LTTV03].

5.3.2.1 Les fichiers Logs

La principale source d'information des visiteurs d'un site Web provient des fichiers Logs qui listent toutes les requêtes HTTP des clients dans l'ordre de leurs visites. Chaque

requête HTTP est représentée par une ligne d'entrée dans le fichier Log dont le format est le suivant :

[Ip] [nom] [login] [date] [url] [statut] [taille] [referrer] [agent]

Ip	adresse électronique de l'utilisateur, cette adresse correspondant souvent au nom de domaine d'un serveur si l'utilisateur est connecté à Internet via un fournisseur d'accès ou une entreprise.
nom/login	supposent que l'utilisateur s'est lui même identifié.
date	date et heure précises de réception de la requête.
url	adresse de la page visitée sur le site (www.<...>).
statut	code retour qui indique si l'action s'est bien déroulée.
taille	indique la taille du fichier retourné.
referrer	signale l'adresse source de laquelle a été effectuée la requête.
agent	le navigateur et le type de système d'exploitation de l'utilisateur.

un exemple d'une ligne d'un fichier Log :

194.78.232.8 – [10/Jan/2003 :15 :33 :43 +0200] "Get /orion/liens.htm HTTP/1.1" 200 1893 "http ://www-sop.inria.fr/orion/index.html" "Mozilla/4.0(compatible ; MSIE 5.0b1 ; Mac_PowerPC)"

"194.78.232.8" étant l'Ip de l'utilisateur, "10/Jan/2003 :15 :33 :43 +0200" la date, "/orion/liens.htm HTTP/1.1" l'URL, "200" statut, "1893" la taille, "http ://www-sop.inria.fr/orion/index.html" le referrer et "Mozilla/4.0(compatible ; MSIE 5.0b1 ; Mac_PowerPC)" l'agent.

L'objectif de l'application en cours de réalisation au sein de l'équipe AxIS est d'analyser des fichiers Logs de trois sites WEB de l'INRIA, à savoir :

- *http ://www.inria.fr/*
- *http ://www-sop.inria.fr/*
- *http ://www-futurs.inria.fr/*

Une des étapes importantes du Web Usage Mining est l'étape de prétraitement des fichiers Logs. Cette étape a été réalisée par l'équipe AxIS à Sophia, [TT03] [LTTV03], et à partir d'un fichier brut archivé par l'INRIA. Ce fichier est composé de 673389 lignes de Logs relevées lors des quinze premiers jours de janvier 2003.

5.3.2.2 Prétraitement des fichiers Logs

Nettoyer, sélectionner et transformer les données est un processus fastidieux et complexe dû principalement à la grande quantité de données et à la faible qualité de l'information qu'on trouve dans les fichiers Logs.
L'INRIA est constitué de six unités de recherche dans toute la France. Il existe un site/serveur Web au niveau national et un pour chaque unité de recherche. Pour notre application, nous avons pris les Logs HTTP du serveur Web national et ceux du serveur d'INRIA Sophia Antipolis. Un utilisateur qui recherche de l'information, navigue parmi tous ces serveurs d'une façon relativement transparente car les pages de différents serveurs Web sont fortement liées entre elles. Il y a de fortes chances que le visiteur ne remarque même pas que le serveur Web a changé. Pour l'analyste du Web Usage Mining, ce changement est très important pour lui permettre d'analyser le comportement de l'utilisateur dans sa recherche de l'information. Ayant un fichier Log Web par serveur, l'analyste doit donc reconstituer le chemin suivi par l'utilisateur sur les différents serveurs sur lesquels ce dernier a navigué. Notre solution est de fusionner tous ces fichiers Logs Web, puis de reconstituer les visites des internautes [TT03].

Deux grandes étapes constituent le prétraitement, à savoir la transformation des données et le nettoyage des données

La transformation des données :

L'étape de transformation des données consiste à fusionner les fichiers Logs, rendre anonymes les Ip (ou les noms des domaines) dans le fichier Log obtenu et à grouper les requêtes par session (même Ip, même Agent). Ensuite, les sessions sont divisées en navigations en choisissant un seuil $\Delta t = 30min$.

1. Avant même de commencer le processus de nettoyage, nous avons dû fusionner les différents fichiers Logs. Les requêtes de tous les fichiers Logs ont été mises ensemble dans un seul fichier. Au préalable, le nom du serveur Web de la requête a été ajouté au nom de la ressource Web demandée. Pour les détails de l'algorithme accomplissant cette tâche, voir [TT03] ;
2. pour des raisons de confidentialité, nous avons remplacé le nom original ou l'adresse

Ip de la machine appelante avec un identificateur. Toutefois dans le codage de l'identificateur, nous gardons l'information sur l'extension du domaine (pays ou type d'organisation : .com, .org, .edu, ...). Pour les machines appelantes de l'INRIA, nous avons gardé certaines informations comme : le nom de l'unité de recherche et un identifiant pour les équipes de recherche ou les services pour une analyse sur l'usage du site par le personnel de l'INRIA ;

3. l'identification de l'utilisateur à partir du fichier Log n'est pas une tâche facile en raison de plusieurs facteurs comme : les serveurs proxy, les adresses dynamiques, le cas d'utilisateurs utilisant le même ordinateur (dans une bibliothèque, club Internet, etc.) ou celui d'un même utilisateur utilisant plus d'un navigateur web ou plus d'un ordinateur. En effet, en employant le fichier Log, nous connaissons seulement l'adresse de l'ordinateur (Ip) et l'agent de l'utilisateur. Il existe d'autres méthodes qui fournissent plus d'information. Les plus utilisées sont : les "Cookies", les pages dynamiques Web (avec un identifiant de session dans l'adresse URL), les utilisateurs enregistrés, les navigateurs modifiés, etc.
   Pour les fichiers Logs de l'INRIA, nous avons utilisé le couple (Ip, agent), pour l'identification de l'utilisateur. Pour ordonner la session multi-serveur de chaque utilisateur, nous avons ordonné le fichier Log par le couple (Ip, agent) et ensuite par le temps. Cette session multi-serveur contient toutes les requêtes de l'utilisateur dans la période analysée ;

4. nous avons ensuite, divisé la session en navigations. Pour cela, nous avons utilisé $\Delta t = 30$ *minutes* comme seuil temporel et largement utilisé comme standard.

Exemple 5.3.1 (Un exemple de construction de session). *Prenons par exemple ce fragment d'un fichier Log contenant 6 requêtes HTTP (unités élémentaires) :*

194.78.232.8 – [10/Jan/2003 :15 :33 :43 +0200] "Get ***/orion/liens.htm*** *HTTP/1.1" 200 1893 "http ://www-sop.inria.fr/orion/index.html" "Mozilla/4.0(compatible ;MSIE 5.0b1 ;Mac_ PowerPC)"*
lucy.ins.cwi.nl – [10/Jan/2003 :15 :34 :07 +0200] "Get ***/stacs2002/*** *HTTP/1.0" 200 1012 "[unknown origin]" "Mozilla/4.74[en](WinNT ;U)"*
lucy.ins.cwi.nl – [10/Jan/2003 :15 :34 :07 +0200] "Get ***/stacs2002/home.html*** *HTTP/1.0" 200 483 "[unknown origin]" "Mozilla/4.74[en](WinNT ;U)"*
194.78.232.8 – [10/Jan/2003 :15 :34 :09 +0200] "Get ***/orion/Telescope/Telescope.html*** *HTTP/1.1" 200 4433 "http ://www-sop.inria.fr/orion/liens.htm" "Mozilla/4.0(compatible ;MSIE 5.0b1 ;Mac_ PowerPC)"*
lucy.ins.cwi.nl – [10/Jan/2003 :15 :34 :10 +0200] "Get ***/stacs2002/cfp.html*** *HTTP/1.0" 200 10334 "http ://www-sop.inria.fr/stacs2002/home.html" "Mozilla/4.74[en](WinNT ;U)"*
194.78.232.8 – [10/Jan/2003 :15 :34 :23 +0200] "Get ***/orion/Telescope/Videosurveillance.html*** *HTTP/1.1" 200 2979 "http ://www-sop.inria.fr/orion/Telescope/Telescope.html" "Mozilla/4.0(compatible ;MSIE 5.0b1 ;Mac_ PowerPC)"*

Ce fragment nous permet de construire deux sessions en considérant le même (Ip, Agent) à savoir :

L'utilisateur provenant de ***194.78.232.8*** *avec l'agent* ***Mozilla/4.0 (compatible ; MSIE 5.0b1 ; Mac_ powerPC)***

- */orion/liens.htm*
- */orion/Telescope/Telescope.html*
- */orion/Telescope/Videosurveillance.html*

L'utilisateur provenant de ***lucy.ins.cwi.nl*** *avec l'agent* ***Mozilla/4.74 [en] (WinNT ; U)***

- */stacs2002/*
- */stacs2002/home.html*
- */stacs2002/cfp.html*

Nettoyage des données :

Le nettoyage des données pour les fichiers Logs consiste à supprimer les requêtes pour

les ressources Web qui ne font pas l'objet de l'analyse (les fichiers images par exemple) et les requêtes ou visites provenant des robots Web.
Pour les portails Web et les sites Web très populaires la dimension des fichiers Logs est comptée en gigabytes par heure. Bien nettoyer donc ces données avant toute analyse est crucial dans le Web Usage Mining. Par le filtrage de données inutiles, on gagne non seulement de l'espace disque, mais dans le même temps on rend plus efficaces les tâches qui suivent dans le processus du Web Usage Mining. Par exemple, dans le cas de notre application et donc des sites Web de l'INRIA, en supprimant les requêtes pour les images et les fichiers multimédia, nous avons réduit les dimensions des fichiers Logs à 40%-50% de la dimension initiale.

Lors de la suppression des requêtes pour des images, la carte du site Web doit être employée car, dans certains cas, ces images ne sont pas incluses dans les fichiers HTML. On peut avoir une image qui nécessite de cliquer sur un lien pour l'afficher. Dans un tel cas, nous devons maintenir la requête pour cette image dans le fichier Log car elle indique une action de l'utilisateur.

Les robots Web sont des logiciels utilisés pour balayer un site Web, afin d'extraire son contenu. Ils suivent automatiquement tous les liens d'une page Web. Les moteurs de recherche, comme Google, envoient régulièrement leurs robots pour extraire toutes les pages d'un site Web afin de mettre à jour leurs index de recherche. Le nombre de demandes d'un robot est en général supérieur au nombre de demandes d'un utilisateur normal. Dans le cas de l'INRIA, la taille des requêtes de Robots Web a représenté 46.5% de la taille du fichier obtenu après avoir supprimé les requêtes pour les images.
La suppression des entrées dans le fichier Log produites par les robots Web simplifie la tâche de fouille de données qui suivra et permet également de supprimer les sessions non intéressantes, en particulier en cas de reconception de site. Habituellement, un robot s'identifie en employant le champ "Ip, Agent" dans les fichiers Logs. Cependant, aujourd'hui il est presque impossible de connaître tous les agents qui représentent un robot car chaque jour apparaissent des nouveaux robots et ceci rend la tâche très difficile.
Nous avons utilisé trois heuristiques pour identifier les requêtes ou navigations issues des robots :

- Identifier les couples (Ip, agent) qui ont fait une requête pour la page 'robots.txt' ;
- utiliser les listes des agents connus comme étant des robots ;
- utiliser un seuil pour la vitesse de navigation, voir définition 5.3.10. Si par exemple, $BS(N_{ij}) > 2$ pages/seconde et $|N_{ij}|$(nombre de pages visitées)> 15 pages, alors la navigation N_{ij} vient d'un robot. Cette étape, doit être exécutée après l'identification des navigations. Une fois que toutes les requêtes/navigations venant des robots ont été identifiées, nous pouvons procéder à leur suppression.

5.3.3 Description de la base de données Log

La base de données relationnelle obtenue est le résultat d'un prétraitement où l'on a éliminé des requêtes inutiles (images,...), certains Robots et où l'on a construit des sessions et des navigations. La méthode de prétraitement implémentée au sein de l'équipe AxIS à Sophia, prend en entrée les fichiers Logs et les topologies des sites et construit en sortie une base de données relationnelle.
Nous décrivons dans ce qui suit les principales tables du schéma relationnel défini pour l'analyse des données d'usage sur le Web par [ALT+03].

5.3.3.1 La Table LOG

Nom	Description
IDRequest	Identifiant de la requête
IDNavigation	Identifiant de la navigation. Identifie un couple (IP, UserAgent) pour des requêtes séparées de moins de 30 minutes. Clé étrangère vers la table NAVIGATION
IDSession	Identifiant de session, identifie un couple (IP, UserAgent). Clé étrangère vers la table SESSION
IP	Adresse IP de la machine qui a initié la requête HTTP. Clé étrangère vers la table IP
UserAgent	Le type et la version du navigateur utilisé pour lancer la requête HTTP
ReqDate	Date de la requête HTTP
ReqTime	Heure de la requête HTTP
Duration	Durée écoulée entre deux requêtes successives de même IDNavigation
FileSize	Taille de la page demandée par la requête HTTP
Statut	Etat ou code retour de la requête HTTP
URL	URL de la page demandée par la requête HTTP. Clé étrangère vers la table URL
referer	Adresse de la page d'où vient la requête

5.3.3.2 La Table URL

La table décrivant les URLs de connexions de tous les navigateurs. Elle contient les URLs, les noms des fichiers qui décrivent l'accès au site, le nom du site et la définition de l'extension du fichier d'arrivée :

Nom	Description
IDUrl	Identifiant de l'URL (arbitraire)
Url	URL
Rubrique1	Rubrique Principale
Rubrique2	Rubrique secondaire
Site	Site de l'URL
FileExtension	Extension du fichier

$$http://\underbrace{www}.inria.fr/\underbrace{rrrt}/\underbrace{tryg}/rr-3378.\underbrace{html}$$

$$\nearrow \qquad \nearrow \quad \nearrow \qquad \nearrow$$

$$Site \quad Rubrique1 \quad Rubrique2 \quad FileExtension$$

5.3.3.3 La Table SESSION

La table SESSION, décrit les sessions. Elle contient les numéros des sessions, le nombre de navigations par session, la durée totale de connexions.

Nom	Description
IDSession	Identifiant de la session
DureeTotal	Durée de la session (en seconde). Elle est définie par la différence entre les dates de la dernière et de la première requête de la session
NbNavigations	Nombre total de navigations dans la session

5.3.3.4 La Table NAVIGATION

La table navigation décrit les navigations. Une navigation est un ensemble de requêtes pseudo-continues émises par une même session. La pseudo-continuité est vrai entre 2 requêtes quand l'intervalle temps est inférieure à 30 minutes. Cette table contient les numéros des navigations, le nombre de connexions à une page, le nombre de connexions à une date et une heure précises, le nombre de connexions à des pages différentes du site

et la durée totale de connexions.

Nom	Description
IDNavigation	Identifiant de la navigation
NbRequest	Nombre de requêtes dans la navigation
NbInstance	Nombre d'instances dans la navigation. Une instance est une action de l'utilisateur déclenchant une ou plusieurs requêtes HTTP. Cette information étant absente du protocole HTTP, on l'approxime en considérant qu'une instance rassemble des requêtes initiées à la même heure (à la seconde près)
NbRequestdif	Nombre d'URL différentes demandées dans la navigation
DureeTotale	Durée totale de la navigation

5.3.3.5 La Table IP

Cette table contient les informations sur les adresses IP des postes connectés. Elle contient l'identifiant des adresses IP, l'adresse IP, le pays contenu dans l'adresse IP, le numéro du domaine, l'unité de recherche de l'INRIA, l'identificateur du projet et l'identificateur du service.

Nom	Description
IDIP	Identifiant de l'adresse IP.
IP	Adresse IP de la machine qui a initié la requête HTTP
Country	Pays de l'IP
IDDomain	Domaine de l'IP
UniteRecherche	Si l'IP est dans le domaine de l'INRIA : unité de recherche
IDProjet	Si l'IP est dans le domaine de l'INRIA et rattachée à un projet : identifiant du projet
IDService	Si l'IP est dans le domaine de l'INRIA et rattachée à un service : identifiant du service

Sur la figure 5.9, nous représentons une partie du schéma relationnel de la base de données obtenue.

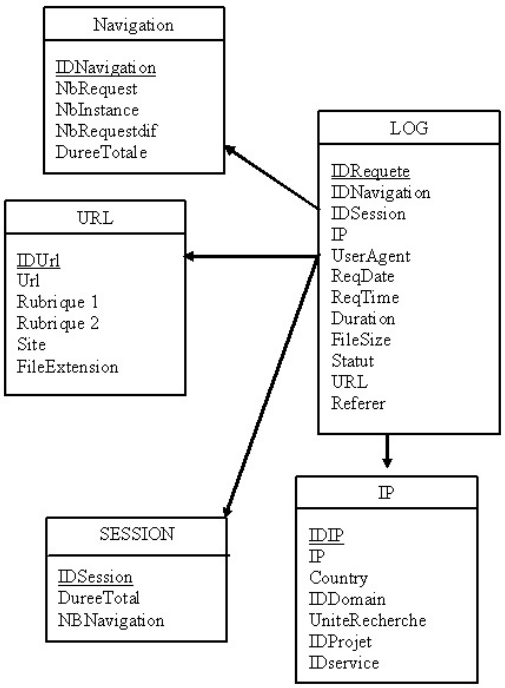


Figure 5.9 – Une partie du schéma relationnel de la base de données Log

5.3.4 Traitements et analyses

À partir de la base de données obtenue, nous avons décidé de sélectionner les navigations d'une durée supérieure à soixante secondes. Nous avons aussi éliminé les pages dont le code statut représente une erreur. Dans nos traitements, nous avons choisi d'analyser les navigations des sites du siège (www) et de Sophia (SOP), l'equivalent de 300 000 pages visitées.

Nous avons crée une taxonomie sur les "rubrique 1". En effet, chaque rubrique 1 appartient à une rubrique sémantique. Par exemple : les rubriques "AxIS", "Sinus", "Sloop", sont des projets de l'INRIA Sophia et donc appartiennent à la rubrique sémantique "projet". Nous avons donc crée une table RubSemamtique qui à chaque rubrique

fait correspondre sa rubrique sémantique.

Sur la figure 5.10, nous représentons la topologie du site du siège. On constate sur cette figure que les rubriques 1 sont groupées par rubriques sémantiques.

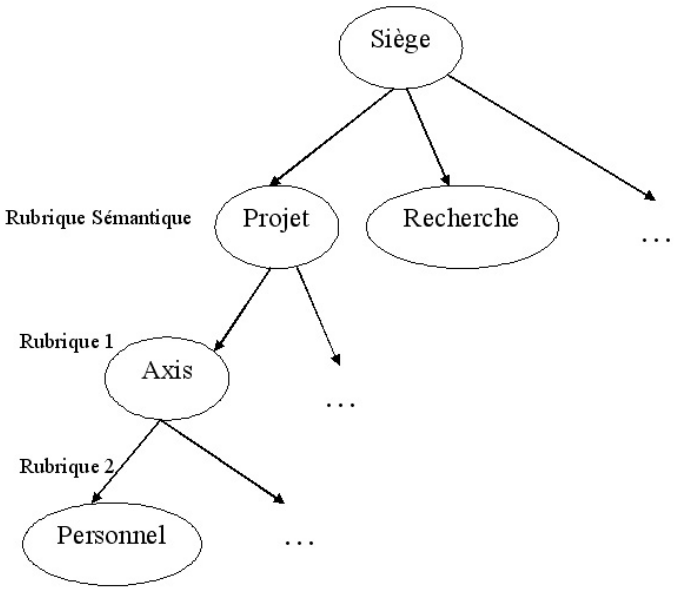


Figure 5.10 – La topologie du site du siège

Dans un premier temps, nous avons sélectionné par requête SQL, le tableau suivant (NavigRub), décrivant les pages visitées des navigations :

Select Id, Navigation.ID_Navig, URL.Site, URL.Rubrique 1

From Navigation, LOG, URL

Where (Navigation.ID_Navig=LOG.IDNavigation) and (Site.IDUrl=LOG.URL)

Id	ID_Navig	Site	Rubrique 1
0	N_1958	www	Robovis
1	N_1958	SOP	Robovis
2	N_1958	www	JGI2002
⋮	⋮	⋮	⋮
282 539	N_158887	www	freesoft
282 540	N_158887	SOP	freesoft

À partir de cette table (NavigRub), résultat d'une requête SQL, nous avons utilisé l'opérateur de généralisation symbolique, décrit au chapitre 2. Dans un premier temps, nous avons construit les descriptions des navigations du siège (www). La requête SQL est la suivante :

Select Id, ID_Navig, Rubrique 1 AS wwwRubrique1,

From NavigRub

Where Site="www"

Cette requête nous permet d'obtenir le tableau symbolique 5.1, décrivant les navigations visitant les pages du siège. La variable symbolique correspondante à la variable "Rubrique 1" est modale. Les modalités correspondent aux différentes rubriques, dotées des poids (ici le nombre de visites).

ID Navig	wwwRubrique1
N_1958	Robovis(10), JGI2001(0),...
N_14564	valorisation(50), semir(80),...
⋮	⋮

Tableau 5.1 – Le tableau symbolique des navigations du siège (www)

Dans un deuxième temps, nous avons construit les descriptions des navigations de Sophia (SOP). La requête SQL est la suivante :

Select Id, ID_Navig, Rubrique 1 As SOPRubrique1,

From NavigRub

Where Site="SOP"

Cette requête nous permet d'obtenir le tableau symbolique 5.2, décrivant les navigations visitant les pages de Sophia. La variable symbolique correspondante à la variable "Rubrique 1" est modale. Les modalités correspondent aux différentes rubriques, dotées des poids (ici le nombre de visites).

Nous obtenons ainsi, deux tableaux symboliques décrivant les mêmes objets par deux ensembles de variables différentes. Nous avons utilisés ensuite, la jointure symbolique (voir chapitre 2, section 2.2.4) afin de concatener les deux tableaux 5.1 et 5.2. Le tableau

ID Navig	SOPRubrique1
N_1958	Robovis(15), thesard(36),...
N_14564	interne(64), saga(18),...
⋮	⋮

Tableau 5.2 – Le tableau symbolique des navigations de Sophia (SOP)

résultat (tableau 5.3) contient 3969 objets navigations visitant donc les pages du siège et les pages de Sophia. Ces navigations sont décrites par les rubriques du siège (95 modalités) et les rubriques de sophia (101 modalités).

ID Navig	wwwRubrique1	SOPRubrique1
N_1958	Robovis(10), JGI2001(0),...	Robovis(15), thesard(36),...
N_14564	Rapports(63), projet(98),...	interne(64), saga(18),...
⋮	⋮	⋮

Tableau 5.3 – Le tableau symbolique des navigations obtenues après généralisation et jointure

À partir du tableau 5.3, on peut déduire par exemple que durant la navigation N_1958, la page "Robovis" (qui est un projet à l'INRIA) disponible sur le site du siège a été visitée 10 fois alors que la page "Robovis" disponible sur le site de Sophia a été visitée 15 fois, la page dont la rubrique est "JGI2001" (qui correspond à une manifestation disponible sur le site du siège) n'a pas été visitée, etc.

Nous allons à présent analyser le tableau symbolique 5.3 en utilisant l'adaptation des cartes topologiques auto-organisatrices de Kohonen aux tableaux de dissimilarités. Pour ce faire, nous avons utilisé des mesures de dissimilarités adéquates à leur formalisme. Notre analyse porte sur deux traitements différents à savoir :

- le traitement des navigations pour trouver des comportements types ;
- le traitement des rubriques pour analyser la perception des sites par les internautes.

5.3.4.1 Traitements des navigations

Ayant le tableau 5.3, nous avons calculé la dissimilarité d'affinité entre deux navigations N_A et N_B, définie au chapitre 1 section 1.5.1.3 et dont la définition est la suivante :

$$d^2(N_A, N_B) = 2 * (1 - a(N_A, N_B))$$

avec a, le coefficient d'affinité entre les différentes distributions de fréquences δ_{N_A} et δ_{N_B} correspondantes aux différentes rubriques des navigations. Le coefficient d'affinité a est défini comme suit : $a(N_A, N_B) = \sum_{j=1}^{p} \sum_{r \in rubrique} \sqrt{\delta_{N_A^j}(r) \times \delta_{N_B^j}(r)}$, j correspond aux différentes variables symboliques.

Nous disposons donc, du tableau de dissimilarités entre les 3969 navigations. Nous avons appliqué la méthode des cartes topologiques sur ce tableau. Les paramètres utilisés pour le déroulement de notre algorithme sont les suivants :

Paramètres	Valeurs
Dissimilarité	affinité
Ensemble d'apprentissage	3969
Nombre d'itérations (N_{iter})	150
Nombre de neurones	$20 : 5 \times 4$
Initialisation	"semi-aléatoire"
cardinal individus référents : q	1

Les résultats obtenus sont assez intéressants. Sur la figure 5.11, nous représentons les individus référents des classes obtenues. Ces individus référents constituent des comportements types de navigations. À chaque case, nous représentons les rubriques 1 visitées. Nous représentons en gras les similarités entre les neurones voisins.
Si nous prennons la classe 1 par exemple, les rubriques 1 en gras sont : "Recherche", "Valorisation", "rrrt" et "rapportactivite". Les voisins directs de la classe 1 sont la classe 2 et la classe 6. Les rubriques "Recherche" et "rapportactivite" ont été visitées par ces trois classes. Les rubriques "rrrt" et "Valorisation" sont communes à la classe 1 et la classe 6.

[1] Le prefixe SOP- signifie que le projet a été consulté à partir du site de Sophia

Recherche, inria, SOP-act-recherche, Travailler, SOP-axis *Classe 16*	**Recherche, Travailler, SOP-act-recherche, SOP-axis, SOP-semir, actu** *Classe 17*	**Recherche, Travailler, SOP-act-recherche, SOP-semir, SOP-actu, SOP-DR, interne, SOP-interne**, services, DR, Relation-ext *Classe 18*	**interne, SOP-interne, SOP-DR, SOP-semir** *Classe 19*	**interne, SOP-DR, SOP-interne, SOP-semir**, semir, DR *Classe 20*
Recherche, Valorisation, inria, rrrt, actualite, rapportactivite, cgi-bin, act-recherche, Travailler personnel, sinus, SOP-sinus, SOP-miaou, SOP-omega, SOP-smash, SOP-caiman *Classe 11*	**Recherche, Travailler, inria,** SOP-lemme, SOP-Oasis *Classe 12*	**DR, Recherche, Travailler, inria, interne**, personnel, cermics, SOP-cermics, SOP-caiman *Classe 13*	agos-sophia, acacia, SOP-axis *Classe 14*	**publication, dias, SOP-cgi-bin, SOP-dias, SOP-interne**, SOP-actu *Classe 15*
Recherche, Valorisation, rrrt, rapportactivite, SOP-epidaure *Classe 6*	**Recherche, rapportactivite, rrrt, inria**, Robovis, SOP-Robovis, SOP-Odyssee *Classe 7*	**rapportactivite, rrrt,** Prisme, SOP-Prisme *Classe 8*	**rrrt, Publications, SOP-cgi-bin, SOP-dias** *Classe 9*	**Publication, SOP-cgi-bin, dias, SOP-dias** *Classe 10*
Recherche, Valorisation, rrrt, rapportactivite, Travailler, presse, personnel, inria, publications, actualite, multimedia, fonctions, SOP-robovis[1], SOP-lemme, SOP-mistral *Classe 1*	**Recherche, rapportactivite,** icare, SOP-icare, RA95 *Classe 2*	caiman, SOP-caiman, SOP-glaad, SOP-Safir, SOP-cgi-bin *Classe 3*	chir, SOP-chir, SOP-Saga *Classe 4*	rrrt, icons, SOP-coprin *Classe 5*

Figure 5.11 – La carte des navigations : représentation des rubriques 1 (sop et siège) des individus référents

Cette première analyse succinte permet de conclure que l'adaptation des cartes topologiques aux tableaux de dissimilarités a bien fonctionné car il y a eu conservation de la topologie. En effet, les classes voisines partagent les même rubriques.

5.3.4.2 Traitements des rubriques

Nous avons choisi ensuite, de nous intéresser à la classification des rubriques afin de trouver des associations. Pour cela, ayant les 3969 navigations grâce à la base de données relationnelle, nous avons construit un tableau décrivant chaque navigation par la liste des "rubriques 1" consultées. À partir de ce tableau on construit un tableau binaire dont les individus sont les 196 "rubriques 1" et les variables sont les navigations : une navigation N_i visitant la rubrique R_j et pas la rubrique R_k sera codée respectivement par 1 pour R_j et 0 pour R_k dans le tableau (voir tableau 5.4).

Navigations / Rubriques	N_1	N_2	...	N_{3969}
R_1	0	1	...	0
R_2	1	0	...	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
R_{196}	0	0	...	0

Tableau 5.4 – Tableau binaire décrivant les 196 rubriques visitées (1) ou pas (0) par une navigation

Dissimilarité : Ayant deux vecteurs binaires R_1 et R_2, pour définir une similarité ou une dissimilarité spécifique, il est nécessaire d'introduire les quatres quantités suivantes :

- soit a le nombre de fois où $R_1^j = R_2^j = 1$;
- soit b le nombre de fois où $R_1^j = 0$ et $R_2^j = 1$;
- soit c le nombre de fois où $R_1^j = 1$ et $R_2^j = 0$;
- soit d le nombre de fois où $R_1^j = R_2^j = 0$;

R_2 \ R_1	1	0
1	a	b
0	c	d

La similarité choisie dans notre cas entre les rubriques est la suivante :

$$S(R_1, R_2) = \frac{a}{a+b+c}$$

Ceci correspond à l'indice de similarité de Jaccard. Cet indice indique la probabilité de visite de la rubrique R_1 et la rubrique R_2 sachant qu'on a visité au moins une des deux. Ayant donc, le tableau de dissimilarités entre les 196 rubriques, nous avons appliqué la méthode des cartes topologiques sur ce tableau. Les paramètres utilisés pour le déroulement de notre algorithme sont les suivants :

Paramètres	Valeurs
Dissimilarité	$1 - S(R_1, R_2) = 1 - \frac{a}{a+b+c}$
Ensemble d'apprentissage	196
Nombre d'itérations (N_{iter})	150
Nombre de neurones	$12 : 4 \times 3$
Initialisation	"semi-aléatoire"
cardinal individus référents : q	1

Les résultats obtenus sont assez intéressants. Dans la classification obtenue nous nous sommes intéressé à la rubrique sémantique "projet" et les classes obtenues sont relativement fidèles à l'organisation des sites des projets de l'INRIA. En effet, en 2003 les projets de l'INRIA été groupés sur les sites par "Thème", il existait 4 thèmes à savoir :

- Thème 1 : Réseaux et systèmes
- Thème 2 : Génie logiciel et calcul symbolique
- Thème 3 : Interaction homme-machine, images, données, connaissances
- Thème 4 : Simulation et optimisation de systèmes complexes

En prenant les individus référents des classes et en se référant aux rubriques sémantiques de chacun de ces individus référents, on obtient la carte suivante (voir figure 5.12) :

manifestation	Projet(Thème 1)	Projet(Thème 3)	inria
manifestation	Projet(Thème 1)	Projet(Thème 4)	Projet(Thème 2)
Projet(Thème 2)	Projet(Thème 4)	Projet(Thème 4)	Projet(Thème 4)

Figure 5.12 – La carte (4× 3) obtenue : représentation de la correspondance sémantique des individus référents (pour les projets on représente le thème auquel ils sont attachés)

Nous constatons tout d'abord une conservation de la topologie. En effet, les projets de thème 1 appartiennent à des classes voisines, de même pour les projets de thème 4 ainsi que les manifestations.

Et pour mieux expliquer les résultats obtenus et voir les associations entre les projets, voici le détail des classes obtenues pour la rubrique sémantique "projet". Nous représentons les individus référents en gras :

Thème 1 meije Thème 2 Koala, croap Thème 3 odyssee Thème 4 Opale **Classe 9**	Thème 1 SOP-mistral [1], SOP-Mimosa, SOP-sloop, SOP-rodeo, rodeo, mascotte, **SOP-mascotte**, sloop, SOP-planete, SOP-oasis **Classe 10**	Thème 3 robovis, epidaure, ariana, acacia, orion, aid, SOP-robovis , SOP-epidaure, SOP-odyssee, SOP-acacia, **SOP-orion**, SOP-ariana, SOP-aid, SOP-axis, SOP-visa **Classe 11**	**Classe 12**
Thème 1 tropics Thème 3 reves Thème 4 Omega **Classe 5**	Thème 1 Mimosa, tick, **SOP-tick** **Classe 6**	Thème 4 **comore**, mefisto, miaou, SOP-mefisto, SOP-smash **Classe 7**	Thème 2 Prisme, SOP-Prisme, SOP-lemme, **SOP-galaad**, SOP-cafe, SOP-saga, SOP-safir **Classe 8**
Thème 2 cafe, lemme, **certilab** Thème 4 Chir, Fractales, opale **Classe 1**	Thème 1 Mistral, planete, SOP-meije Thème 2 oasis, saga, safir, SOP-Koala Thème 4 **caiman**, sinus **Classe 2**	Thème 4 icare, SOP-sinus, **SOP-icare**, SOP-miaou, SOP-caiman **Classe 3**	Thème 1 SOP-tropics Thème 2 SOP-certilab Thème 3 SOP-reves Thème 4 **SOP-Omega**, SOP-sysdys **Classe 4**

[1] Le prefixe SOP- signifie que le projet a été consulté à partir du site de Sophia

Comme on peut le voir sur la carte détaillée, aucun projet n'a été affecté à la classe 12, classe représentée par la rubrique sémantique "inria". Ceci permet de déduire que la classe 12 est assez homogène.

Nous constatons aussi que les classes 3, 6, 7, 8, 10 et 11 sont composées uniquement de projets appartenant aux même thèmes ce qui permet de déduire que l'adaptation du SOM effectue une bonne quantification de l'espace. Nous pouvons en outre constater pour la classe 11, constituée de projets appartennant au thème 3, la présence simultanée du projet *Aid* et du projet *Axis*. Il faut savoir que le projet *Axis* a remplacé le projet *Aid* au sein de l'INRIA. La visite de l'un entraine donc très souvent la visite de l'autre, car il un lien mutuel existe entre les deux pages. Nous retrouvons le même comportement pour le projet *Odyssee* et le projet *Robovis*.

Nous constatons la présence de projets dans les classes 5 et 9. En effet, ces deux classes sont représentées par des manifestations et donc la présence des projets permet de déduire que les manifestations sont liées à ces projets. Ce qui explique la visite des pages des projets.

Un dernier point intéressant, si nous revenons à la première carte 5.12, nous constatons que les projets de thème 2 appartiennent à des classes très éloignées sur la carte. Pour expliquer ce phénomène, il faut savoir que :

- tout projet à l'INRIA a son propre site Web localisé sur le serveur local de l'unité de recherche à laquelle il est rattaché ;
- de plus, tout projet à l'INRIA possède une page descriptive sur le serveur national du siège.

Prenons l'exemple du projet "cafe", qui est un projet de l'unité de recherche de Sophia-Antipolis. Son site Web est donc localisé sur le serveur de l'unité de recherche de Sophia, que nous avons noté **SOP-cafe** sur la carte. La page descriptive du projet "cafe" est quant à elle localisée sur le serveur national du siège, que nous avons noté **cafe** sur la carte. Nous nous attendons à ce que ces deux rubriques, **cafe** et **SOP-cafe**, apparaissent dans la même classe ou dans des classes voisines, car elles sont liées sémantiquement. Or, comme on peut le constater sur la carte, la rubrique **cafe** appartient à la classe 1 et la rubrique **SOP-cafe** appartient à la classe 8 qui ne sont pas voisines. Ce phénomène peut

être expliqué par l'absence de lien dans la page descriptive vers le site Web du projet "cafe". Donc au cours d'une même navigation, l'internaute peut difficilement passer de la page descriptive vers le site Web du projet. Ce qui démontre un défaut de conception du site. Nous remarquons le même comportement pour le projet lemme.

5.4 Conclusion

Dans ce chapitre, nous avons présenté deux applications sur des données à structure complexe. La première application concerne les données fonctionnelles et a permis de confirmer que le choix d'une mesure de dissimilarité adaptée est important pour la méthode. Dans la deuxième application, nous avons proposé une démarche pour l'exploitation des fichiers Logs Web. L'originalité de notre approche est d'utiliser dans un premier temps, le formalisme des données symboliques. En effet, à partir de plusieurs fichiers Logs Web provenant de plusieurs sites, nous avons construit une base de données. L'exploitation des données de la base a été réalisée grâce aux outils de généralisation de l'analyse de données symboliques. La définition ensuite d'une mesure de dissimilarité adéquate a permis d'appliquer notre adaptation des cartes topologiques sur tableaux de dissimilarités et d'obtenir des résultats.

Conclusions et perspectives

Au cours de ce travail de recherche, nous nous sommes fixés les deux objectifs suivants :

1. améliorer la méthode de généralisation symbolique, qui a fait ses preuves dans la modélisation des données complexes ;
2. apporter une contribution en classification des données à structure complexe.

Concernant le premier point, nous avons introduit le formalisme et plusieurs indices de proximité entre descriptions symboliques. Nous avons aussi introduit la méthode de généralisation symbolique qui permet de construire un tableau d'assertions à partir de la sélection et de la structuration des informations provenant d'une base de données relationnelle. Cette généralisation étant une étape cruciale dans la suite des traitements, notre travail a tout d'abord porté sur son amélioration. En effet, la généralisation symbolique étant supervisée, lorsque les données que l'on souhaite généraliser sont très hétérogènes, les descriptions obtenues incluent des individus virtuels. Nous avons donc proposé une décomposition basée sur un algorithme divisif de classification. Les groupes étant choisis d'une manière supervisée, une décomposition est alors appliquée à chaque groupe hétérogène, groupe dont le critère de densité est faible. Les avantages de cette décomposition au niveau de la généralisation sont les suivants :

- la décomposition des groupes hétérogènes se fait sur des données quantitatives ou qualitatives. Les critères utilisés sont donc basés sur le calcul des centres de gravité, ce qui accélère les calculs ;
- la méthode proposée traite les données manquantes ;
- les résultats obtenus sont de meilleure qualité. En effet, les descriptions des groupes

sont plus homogènes ;

- les résultats de la décomposition sont modélisés par des données symboliques ;
- la méthode permet d'extraire des connaissances supplémentaires sur les groupes. Ces connaissances pourront être utiles pour des analyses ultérieures.

Nous avons aussi proposé une méthode de généralisation symbolique basée sur l'algorithme des cartes topologiques de Kohonen. L'avantage de cette méthode est de réduire les données d'une manière non supervisée et de modéliser les groupes homogènes obtenus par des données symboliques.
Ces méthodes de généralisation supervisée ou non supervisée, possèdent l'avantage d'une représentation symbolique des données synthétisées. Elles constituent non seulement un outil descriptif pour l'utilisateur mais aussi une étape intermédiaire et importante permettant d'autres analyses. Bien que cette représentation se soit révélée fructueuse, d'autres représentations peuvent être envisagées.

Après cette étape de généralisation, la principale motivation fut de définir une méthode de classification capable de traiter aussi bien les représentations symboliques que d'autres représentations des données à structure complexe et de fournir une interprétation des classes obtenues. La méthode proposée est une adaptation des cartes topologiques aux tableaux de dissimilarités. L'avantage que procure cette approche est son application à tout type de données. En effet, en analyse de données symboliques, les méthodes déjà développées ne traitent que les données assertions. Suite à l'amélioration du processus de généralisation, nous avons proposé une modélisation en Multi-assertions ou en hordes. Une définition de mesures de dissimilarités adéquates à ces types de données permettra donc l'application de la méthode de classification proposée. Il serait donc intéressant dans un travail ultérieur de définir des mesures de dissimilarités à tous les types du formalisme symbolique.

D'autre part, dans le cas des données symboliques, la représentation des individus référents associée à une classe s'interprète plus facilement que la description extensive de la classe (i.e. la liste de ses membres). Cependant, dans nos expériences nous nous sommes intéressés qu'au cas d'un espace de représentation L_c d'un neurone par un seul individu (q=1). Une représentation par plusieurs individus, peut être plus intéressante

et plus souple mais n'a pas été abordée par faute de temps. Ce gain en souplesse, se paie bien sûr en coût de calcul.

Le passage par un tableau de dissimilarités inter-individuelles peut être très coûteux quand le nombre d'individus est très important. Une solution est de travailler sur un échantillon d'individus. À la fin de la phase d'apprentissage, la fonction de voisinage étant faible, l'algorithme se comporte comme un algorithme des nuées dynamiques. On affecte alors les individus restants aux individus référents en fonction des dissimilarités initiales. Une autre perspective intéressante, sur le plan de la complexité, serait d'optimiser la phase de représentation en limitant la recherche des individus référents d'un neurone aux individus des neurones voisins.

Avant d'exploiter plus en avant les différentes possibilités offertes par la méthode proposée, il sera intéressant d'étudier la robustesse de cette méthode. Cette étude peut être réalisée via une comparaison des résultats obtenues par la méthode des cartes topologiques sur tableaux de dissimilarités avec une méthode combinant l'algorithme des Kmeans et le multi-dimensional scaling (MDS) par exemple.

Annexe A

Implémentation

L'implémentation est une étape primordiale dans ce travail. Au cours de cet annexe, nous allons donc présenter les détails des deux contributions de cette thèse. La première concerne la méthode de décomposition intégrée au processus d'extraction de données symboliques (programme méthode de décomposition), et la seconde concerne l'adaptation des cartes topologiques de Kohonen aux tableaux de dissimilarités (programme DissSOM).

A.1 Programme méthode de décomposition

La méthode de décomposition a été implémentée en C/C++ puis intégrée au processus d'extraction de données symboliques à partir des bases de données relationnelles (DB2SO) qui existe dans le logiciel SODAS. En effet, le logiciel SODAS (Symbolic Object Data AnalysiS) comme son nom l'indique, est un logiciel de traitement et d'analyse de données symboliques. Ce logiciel regroupe quelques méthodes de discrimination, de régression, de classification et de visualisation de données symboliques ainsi que le processus DB2SO d'extraction de données symboliques. En effet, le module DB2SO accède à une base de données relationelle, par un lien ODBC, et récupère les données via des requêtes SQL. Les tuples décrivant les groupes sont résumés sous forme de données symboliques. Le programme méthode de décomposition réalise une décomposition de chaque groupe en un nombre fixe de sous groupes. Les données en entrée de la méthode sont les tuples décrivant le groupe et le nombre de classes fixé par l'utilisateur. En sortie, le

programme produit un fichier SODAS (d'extension .sds ou .xml) qui comprend, pour l'essentiel, un dictionnaire des variables, un dictionnaire des individus et une matrice de descriptions symboliques (individus $\times$ variables).

A.2 Programme DissSOM

Le programme DissSOM a été écrit en C/C++. Le code source est directement compilable sous Windows par Visual C++, ainsi que sous Unix avec g++. La méthode prend en entrée soit un fichier SODAS (d'extension .sds ou .xml) contenant la matrice de dissimilarité (individus $\times$ individus) soit un fichier texte (.txt) contenant la matrice de dissimilarité triangulaire. Initialement l'utilisateur devra fixer les paramètres, à savoir :

- le nombre total d'itérations N_{iter} ;
- le type d'initialisation (initialisation "semi aléatoire" ou initialistion par AFTD) ;
- les dimensions de la carte $xdim$, $ydim$ pour une initialisation "semi-aléatoire" et le nombre de neurones m pour une initialisation par AFTD ;
- le type de voisinage (voisinage bubble ou voisinage gaussien) voir chapitre 1, session 1.2.3.3 ;
- et les valeurs T_{max} et T_{min}.

Il a été décidé de ne pas coder d'interface graphique propriétaire, où propre à un type d'architecture où de système d'exploitation. Au lieu de cela, le programme fournit automatiquement les résultats sous forme d'un rapport d'une dizaine de pages écrit en LaTeX. Certaines des figures contenues dans ce rapport sont générées directement en PostScript, tandis que d'autres telles que les diverses courbes décrivant l'évolution des critères d'erreur, sont obtenues en lançant des scripts GnuPlot. L'intérêt d'une sortie de ce type est qu'elle permet de fournir à l'utilisateur de nombreuses informations concernant les paramètres de l'algorithme, ainsi que les résultats. Il est par exemple possible d'obtenir les équations des divers critères de contrôle de la convergence, ainsi que les fonctions de voisinage, le type d'initialisation, etc.

A.2.1 Etude de la complexité algorithmique

Nous présentons dans ce qui suit l'étude de complexité de l'algorithme DissSOM. Pour ce faire, nous considérons les paramètres suivants :

- **n** : nombre total d'individus appartenant à l'ensemble d'apprentissage ;
- **m** : nombre total de neurones de la carte, avec $m < n$;
- $\mathbf{N_{iter}}$: nombre total d'itérations.

L'étape d'initialisation de l'algorithme DissSOM, permet d'associer à chaque neurone une représentation. Nous distinguons deux cas :

1. une initialisation "semi-aléatoire", voir chapitre 4 session 4.2.4. Dans ce cas, nous affectons aléatoirement les référents de chacun des m neurones parmi l'ensemble d'apprentissage. Ainsi, le coût de l'initialisation "semi-aléatoire" est de l'ordre de (m), soit $\mathbf{O(m)}$;
2. une initialisation par AFTD, voir chapitre 4 session 4.2.4. Le coût de l'AFTD est de l'ordre de $O(n^2)$. Après obtention des coordonnées des m neurones dans l'espace de projection grâce à l'AFTD, chaque neurone sera représenté par un des n individus de l'ensemble d'apprentissage le plus proche au sens de la distance euclidienne dans l'espace de projection. Ainsi, le coût de l'initialisation par AFTD est de l'ordre de $(mn + n^2)$, soit $\mathbf{O(n^2)}$.

La structure de l'algorithme DissSOM nécessite N_{iter} itérations et chaque itération comporte deux étapes. Le coût de chaque étape est définie comme suit :

1. la première étape qui est la phase d'affectation est itérative. Celle-ci considère chaque élément z_i de l'ensemble d'apprentissage (contenant (n) individus) et l'affecte au neurone de la carte (contenant (m) neurones) le plus proche au sens de la fonction d'adéquation d^T (voir chapitre 4 équation 4.2.2). Or la fonction d'affectation f_{z_i} considère chaque neurone et itère sur tous les autres neurones de la carte. Par conséquent, le coût de l'affectation d'un individu au neurone le plus proche est de l'ordre de (m^2). Ainsi, le coût de la phase d'affectation est de l'ordre de (nm^2), soit $\mathbf{O(nm^2)}$;
2. la deuxième étape qui est la phase de représentation est itérative. Celle-ci considère

chaque neurone r de la carte, i.e. (m) et trouve l'individu parmi les n éléments de l'ensemble d'apprentissage, qui minimise la fonction E_r (voir chapitre 4 équation 4.2.3). Or la fonction E_r considère chaque individu et itère sur tous les éléments de l'ensemble d'apprentissage. Par conséquent le coût de la représentation d'un neurone est de l'ordre (n^2), Ainsi, le coût de la phase de représentation est de l'ordre de (mn^2), soit $\mathbf{O(mn^2)}$.

Ainsi, nous avons :

Une initialisation qui est de l'ordre de $\max(\mathbf{O(m)}, \mathbf{O(n^2)})$.

Un parcours itératif de l'ordre de $\mathbf{O(N_{iter})}$, qui comporte :

1. l'affectation de tout l'ensemble d'apprentissage à la carte qui est de l'ordre de $\mathbf{O(nm^2)}$;
2. la recherche des nouveaux représentants des neurones, qui est de l'ordre de $\mathbf{O(mn^2)}$.

Le coût C de l'algorithme DissSOM est calculé comme suit :

$C = \max(O(m), O(n^2)) + O(N_{iter})(O(nm^2) + O(mn^2))$

$\rightsquigarrow O(n^2) + O(N_{iter})(O(mn^2))$

$\rightsquigarrow O(N_{iter}mn^2)$

Dans le cas pratique, souvent $N_{iter}m \simeq n$. Nous en déduisons que le coût C de l'algorithme DissSOM est de type polynomial et qu'il est de l'ordre de $O(n^3)$.

Bibliographie

[ALT+03] Mireille Arnoux, Yves Lechevallier, Doru Tanasa, Brigitte Trousse, and Rosanna Verde, *Automatic clustering for the web usage mining*, Proceedings of the 5th Intl. Workshop on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC03) (Dana Petcu, Daniela Zaharie, Viorel Negru, and Tudor Jebeleanu, eds.), Editura Mirton, Timisoara, 1-4 October 2003, pp. 54–66.

[ASBT00] Christophe Ambroise, Geniève Sèze, Fouad Badran, and Sylvie Thiria, *Hierarchical clustering of self-organizing maps for cloud classification.*, Neurocomputing 30., Elsevier Science, 2000, pp. 47–52.

[Aub94] Jean-Pierre Aubin, *Initiation à l'analyse appliquée*, ch. Analyse Multivoque, pp. 304–307, Masson, 1994.

[BD99a] Hans H. Bock and Edwin Diday, *Analysis of symbolic data, exploratory methods for extracting statistical information from complex data*, Springer, 1999.

[BD99b] ———, *Analysis of symbolic data, exploratory methods for extracting statistical information from complex data*, ch. Similarity and dissimilarity, pp. 139–197, Springer, 1999.

[BN85] H. Bacelar-Nicolau, *The affinity coefficient in cluster analysis*, Methods of operations research **53** (1985), 507–512.

[BN00] ———, *Analysis of symbolic data : exploratory methods for extracting statistical information from complex data*, ch. Similarity and Dissimilarity, pp. 160–165, H. H. Bock and E. Diday, 2000.

[Car94] F. De Carvalho, *New approches in classification and data analysis*, ch. Proximity coefficients between Boolean symbolic objects, pp. 387–394, Springer-verlag, 1994.

[Car96] ———, *Histogrammes et indices de proximités en analyse de données symboliques*, Actes de l'école d'été sur l'analyse de données symboliques. LISE-CEREMADE, Université de Paris IX-Dauphine, Paris (1996), 101–127.

[CCD99] A. Chouakria, P. Cazes, and E. Diday, *Analysis of symbolic data, exploratory methods for extracting statistical information from complex data*, ch. Symbolic principal Component Analysis, pp. 200–211, Springer, 1999.

[CCLG01] Marie Chavent, Antonio Ciampi, Yves Lechevallier, and Aicha El Golli, *Classification automatique en deux étapes : modélisation probabiliste des neurones d'une carte topologique suivie par une classification divisive*, Société Francophone de Classification SFC (Décembre 2001).

[CCLV03] M. Chavent, F. De Carvalho, Y. Lechevallier, and R. Verde, *Trois nouvelles méthodes de classification automatique de données symboliques de type intrevalle*, Revue Statistique Appliquée **4** (2003), 5–29.

[CF87] M. Cottrell and J.C. Fort, *Etude d'un processus d'auto-organisation*, Ann. Inst. Henri Poincaré **23** (1987), no. 1, 1–20.

[CF97] ———, *Theoretic aspects of the som algorithm*, Prépublication du SAMOS, vol. 73, Centre de recherche SAMOS, Avril 1997.

[CGGR04] Brieuc Conan-Guez, Aïcha El Golli, and Fabrice Rossi, *Clustering functional data with som algorithm*, ESANN (European Symposium on Artificial Neural Networks), Avril 2004, pp. 305–312.

[CGK78] K. Chidananda-Gowda and G. Krishna, *Disaggregative clustering using the concept of mutual nearest neighborhood.*, IEEE Transactions on Systems, Man, and Cybernetics 8., 1978, pp. 888–895.

[Cha95] Marie Chavent, *Choix de bases pour un partitionnement d'objets symboliques*, Actes du IIIème congrès de la société Francophone de Classification (1995).

[Cha97] ———, *Analyse de données symboliques une méthode divisive de classification*, Thèse de doctorat, Université Paris Dauphine, 1997.

[Cha98] ———, *a monothetic clustering method*, pattern recognition letters **19** (1998), 989–996.

[Cha00] ———, *Criterion-based divisive clustering for symbolic objects*, Analysis of symbolic data : exploratory methods for extracting statistical information from complex data, Springer, 2000, pp. 299–311.

[Cho98] Ahlame Chouakria, *Extension des méthodes d'analyse factorielle à des données de type intervalle*, Thèse de doctorat, Université Paris Dauphine, 1998.

[CL00] Antonio Ciampi and Yves Lechevallier, *Clustering large, multi-level data sets : An approach based on kohonen self organizing maps*, principales of Data Mining and knowledge discovery, 4^{th} European conference, PKDD, 2000, pp. 353–358.

[CS99] Marie Chavent and Véronique Stephan, *From generalization to clustering in the relational database context*, Proceedings of the conference on Knwoledge Extraction and Symbolic Data Analysis : KESDA'98, Eurostat's collection, 1999, pp. 105–117.

[DB89] Edwin Diday and Paula Brito, *Symbolic cluster analysis*, Conceptual and numerical analysis of data (Otto opitz, ed.), 1989.

[DcG+89] Edwin Diday, Gilles celeux, Gérard Govaert, Yves Lechevallier, and H. Ralambondrainy, *Classification automatique des données*, DUNOD informatique, 1989.

[Did71] Edwin Diday, *La méthode des nuées dynamiques*, Revue statistique appliquée **XIX** (1971), no. 2, 19–34.

[Did89] ———, *Introduction à l'approche symbolique en analyse de données*, RAIRO (Revue, d'Automatique, d'informatique et de Recherche Opérationnelle) **23** (1989), no. 2.

[Did98] ———, *L'analyse des données symboliques, un cadre théorique et des outils*, cahiers du CEREMADE, Université Paris Dauphine (1998), no. 9821, 26–28.

[DK91] Edwin Diday and Yves Kodratoff, *des objets de l'analyse de données à ceux de l'analyse des connaissances*, Induction symbolique et numerique à partir des données, Éditions Cépaduès, 1991, pp. 9–76.

[DLPT82] E. Diday, J. Lemaire, J. Pouget, and F. Testu, *Eléments d'analyse de données*, Dunod, 1982.

[dR02] Aurélien de Reyniès, *Classification de données symboliques : une extension de la méthode des nuées dynamiques*, Actes du IXème congrès de la société Francophone de Classification (2002), 177–180.

[dR03] ———, *Classification et discrimination en analyse de données symboliques*, Thèse de doctorat, Université Paris Dauphine, 2003.

[Ele99] Olivier Elemento, *Initialisation, convergence et validation de cartes topologiques de kohonen*, Rapport de dea, Université Paris Dauphine, 1999.

[FV03] Frédéric Ferraty and Philippe Vieu, *Curves discriminations : a nonparametric functional approach*, Computational statistics and Data Analysis **44** (2003), 161–173.

[GCG03] Aicha El Golli and Brieuc Conan-Guez, *Adaptation des cartes topologiques auto-organisatrices aux tableaux de dissimilarités*, Xèmes Rencontres de la Société Francophone de Classification (Septembre, 2003), 99–102.

[GCGR04] Aicha El Golli, Brieuc Conan-Guez, and Fabrice Rossi, *a self-organizing map for dissimilarity data*, accepté à IFCS (International Federation of Classification Societies), 2004.

[GG90] O. Gascuel and A. Guénoche, *Approche symbolique-numérique en apprentissage*, actes des journées nationales du PRC-GRECO IA **Hermes** (1990).

[GL02] Aicha El Golli and Yves Lechevallier, *Une méthode divisive de classification*, Société Francophone de Classification, SFC Toulouse (Septembre 2002), 193–196.

[GL03] ———, *Extraction de classes homogènes et création d'objets symboliques*, XXXVèmes journées de statistique (juin, 2003).

[GO99] Thore Graepel and Klaus Obermayer, *A stochastic self-organizing map for proximity data*, Neural Computation, vol. 11, 1999, pp. 139–155.

[Gol02] Aicha El Golli, *Bases de données relationnelles : construction d'objets symboliques par généralisation*, Extraction des connaissances et apprentissage EGC2002 **1** (Janvier 2002), 422.

[HD98] G. Hébrail and A. Debregeas, *Interactive interpretation of kohonen maps applied to curves*, Proceedings of the 4^{th} International Conference on Knowledge Discovery and Data Mining, 1998, pp. 179–183.

[JD88] Anil K. Jain and Richard C. Dubes, *Algorithms for clustering data*, Prentice Hall, 1988.

[KD91] Y. Kodratoff and E. Diday, *Préambule : approche numérique et approche symbolique*, ch. Induction Symbolique et Numérique à partir de données, Cepadues, 1991.

[Koh82a] Teuvo Kohonen, *Analysis of a simple self-organizing process*, Biol. cybern. **44** (1982), 135–140.

[Koh82b] ———, *Self-organized formation of topologically correct feature map*, Biol. Cybern **43** (1982), 59–69.

[Koh97] ———, *Self-organizing maps*, Springer Verlag, New York, 1997.

[LN99] Brian Lavoie and Henrik Frystyk Nielsen, *Web characterization terminology & definitions sheet*, http ://www.w3c.org/1999/05/WCA-terms/, May 1999.

[LTTV03] Yves Lechevallier, Doru Tanasa, Brigitte Trousse, and Rosanna Verde, *Classification automatique : Application au web mining*, 10ème rencontres de la société Francophone de classification (2003), 157–160.

[Mac65] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, Proc.of the Fifth Berkeley Symposium on Math., Stat. and Prob., vol. 1, 1965, pp. 281–297.

[Mat51] K. Matusita, *Decision rules based on distance for problems of fit, two samples and estimation*, Ann. Math. Stat. **3** (1951), 1–30.

[Mat55] ———, *On the theory of statistical decision functions*, Ann. Math. Stat. **26** (1955), 631–640.

[MDS82] R. Michalski, E. Diday, and R. Stepp, *A recent advance in data analysis : Clustering objects into classes characterized by conjunctive concepts*, Progress in pattern recognition (1982), 33–55.

[MS64] MacNaughton-Smith, *Dissimilarity analysis : A new technique of hierarchical subdivision.*, Nature 202, Éditions Cépaduès, 1964, pp. 1034–1035.

[MS83] R. Michalski and R. Stepp, *Learning from observations : Conceptual clustering*, Machine learning : An artificial intelligence approach (1983), 331–363.

[Mur95] F. Murtagh, *Interpreting the kohonen self-organizing feature map using contiguity-constrained clustering.*, Pattern Recognition Letters 16, Elsevier Science, April 1995, pp. 399–408.

[RCG03] Fabrice Rossi and Brieuc Conan-Guez, *Un modèle semi-paramètrique neuronal pour la régression et la discrimination sur données fonctionnelles*, Soumis RSA (2003).

[RS97] Jim Ramsay and Bernard Silverman, *Functional data analysis*, Springer Series in Statistics, Springer Verlag, June 1997.

[Rus70] E.M. Ruspini, *Numerical methods for fuzzy clustering.*, Information Science **2** (1970), 319–350.

[Sap90] Gilbert Saporta, *Probabilités, analyse des données et statistique*, Technip, 1990.

[SHL00] Véronique Stephan, Georges Hébrail, and Yves Lechevallier, *Analysis of symbolic data : exploratory methods for extracting statistical information from complex data*, ch. Generation of symbolic objects from relational databases, pp. 78–105, Springer, 2000.

[SPA83] SPAD, *Système portable pour l'analyse des données*, Editions DECISIA, France, 1983.

[Ste98] Véronique Stephan, *Construction d'objets symboliques par synthèse des résultats de requêtes sql*, Thèse de doctorat, Université Paris Dauphine, 1998.

[Tea02] Sylvie Thiria and Gérard Dreyfus et al., *Réseaux de neurones méthodologie et applications*, Eyrolles, Paris, 2002.

[TLGC97] Sylvie Thiria, Yves Lechevallier, Olivier Gascuel, and Stéphane Canu, *Statistique et méthodes neuronales*, Dunod, Paris, 1997.

[Tor52] W. S. Torgerson, *Multidimensional scaling : I. theory and method*, Psychometrika **17** (1952), 401–419.

[TT03] Doru Tanassa and Brigitte Trousse, *Le prétraitement des fichiers log web dans le web usage mining multi-sites*, journée Francophones de la toile (2003).

[US90] A. Ultsch and H.P. Siemon, *Kohonen's self organizing feature maps for exploratory data analysis.*, Proc. Intern. Neural Networks, Kluwer Academic Press, Paris 1990, pp. 305–308.

[War63] J.H. Ward, *Hierchical grouping to optimize an objective function*, JASA, 58, 1963.

[WWL+99] Jason Tsong-Li Wang, Xiong Wang, King-Ip Lin, Dennis Shasha, Bruce A. Shapiro, and Kaizhong Zhang, *Evaluating a class of distance-mapping algorithms fordata mining and clustering*, Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, 1999, pp. 307–311.

[ZyL93] Xuegong Zhang and yanda Li, *Self-organizing map as a new method for clustering and data analysis.*, International Joint Conference on Neural Networks, 1993, pp. 2448–2451.