Class-Add, une procédure de sélection de variables basée sur une troncature k-additive de l'information mutuelle et sur une classification ascendante hiérarchique en pré-traitement

Hélène Daviet

Université de Nantes

Laboratoire d'Informatique de Nantes Atlantique

ClassAdd, une procédure de sélection de variables basée sur une troncature k-additive de l'information mutuelle et sur une Classification Ascendante Hiérarchique en pré-traitement.

THÈSE

présentée et soutenue publiquement le 11 mars 2009

pour l'obtention du grade de

Docteur de l'Université de Nantes

Spécialité Informatique

par

Hélène Daviet

Composition du jury

Rapporteurs :	Younès Bennani, Professeur, LIPN-CNRS, Paris XIII André Hardy, Professeur, FUNDP, Namur
Examinateurs :	Yves Lechevallier, Directeur de recherche, INRIA Rocquencourt Israël-César Lerman, Professeur émérite, IRISA, Rennes
Directeur :	Pascale Kuntz, Professeur, LINA, équipe COD, Nantes
Co-encadrant :	Ivan Kojadinovic, Maître de conférences (HDR), Université d'Auckland

N° ED 503-040

École Doctorale Sciences et Technologies de l'Information et des Matériaux

Mis en page avec la classe thloria.

Remerciements

Je remercie Pascale Kuntz et Ivan Kojadinovic de m'avoir encadrée pendant ces années de thèse. Merci Pascale de m'avoir toujours transmis ta confiance en la réussite de ces travaux même pendant cette longue dernière année « loin » de Nantes.

Je remercie Younès Bennani et André Hardy d'avoir accepté d'être les rapporteurs de ce travail et Yves Lechevallier et Israël-César Lerman de faire partie du jury.

Je remercie l'entreprise PerformanSe de m'avoir accueillie sous contrat CIFRE pendant trois ans et particulièrement, Jacques et Vincent Philippé et Serge Baquedano à l'origine de ce partenariat.

Je remercie les collègues de PerformanSe dont la bonne humeur m'a permis de passer trois années très agréables là-bas avec une pensée spéciale pour le compagnon de galère thésard, Philippe.

Je remercie aussi les membres de l'équipe COD et ses thésards qui m'ont ouvert la voie : Julien, Jérôme, Manu, Nicolas et Bruno.

Enfin, je remercie Stéphane sans qui cette thèse n'aurait sûrement jamais été terminée et aussi mon petit Pierre qui a été bien mignon pour laisser sa maman travailler.

À Stéphane et Pierre.

Table des matières

Table des figures

Liste des tableaux

Introduction

La sélection de variables

Au début des années 90, la majorité des publications sur la sélection de variables portait sur des domaines souvent décrits par quelques dizaines de variables. Ces dernières années, de par l'accroissement des capacités de recueil, de stockage et de manipulation des données, la situation a beaucoup changé. Il n'est plus rare de rencontrer dans certains domaines, en particulier en bio-informatique ou en fouille de textes, des centaines voire des milliers de variables. Par conséquent, de nouvelles techniques de sélection de variables sont apparues pour tenter d'aborder ce changement d'échelle et de traiter notamment la prise en compte des variables redondantes et des variables non pertinentes. Cette problématique apparaît dans de nombreuses applications en apprentissage, notamment lorsque le processus est supervisé. Nous disposons généralement d'un ensemble d'apprentissage de taille fixe que ce soit pour les variables ou pour les individus. De cet ensemble, nous devons construire un modèle de classification des individus. Ce modèle est ensuite utilisé pour prédire la classe de nouveaux individus. Une première intuition pourrait laisser penser que le pouvoir discriminant d'un algorithme croît avec le nombre de variables. La situation n'est pas si simple puisqu'en parallèle, l'augmentation du nombre de variables peut entraîner une augmentation dramatique du temps d'exécution de l'algorithme. S'ajoutent à cette complexité algorithmique, des difficultés inhérentes au contenu même de l'information traitée : certaines variables sont redondantes et d'autres non pertinentes pour la prédiction de classes. De plus, beaucoup d'algorithmes d'apprentissage se basent sur l'estimation de la probabilité d'avoir telle classe en fonction de l'ensemble des variables, et, dès que le nombre de variables est trop élevé, la distribution devient alors bien souvent difficile à estimer.

Les deux principaux nouveaux domaines d'application qui ont stimulé le regain d'intérêt pour la problématique de la sélection de variables en vue d'une classification sont la bio-informatique – notamment, la génétique – et la fouille de données textuelles. Pour les données génétiques, les variables représentent généralement l'expression de gènes par leur séquence biologique de nucléotides (A, G, T, C) pour un certain nombre de patients. Une classification typique est la séparation des patients sains des patients atteints d'une certaine pathologie basée sur leur « profil génétique ». Dans ce type de jeu assez difficile à construire, on ne possède souvent guère plus de 100 patients pour constituer un jeu

d'apprentissage et un jeu de test ; en revanche, le nombre de variables manipulées peut varier de 6000 à 60000 (e.g. Hanczar et al., 2007). Un premier filtre est fréquemment utilisé pour ne garder qu'une centaine de variables à étudier. En ce qui concerne la fouille de textes, les individus manipulés sont des documents. Ils sont souvent décrits par un vecteur fonction de la taille du vocabulaire qui contient les mots les plus fréquents des documents. Des vocabulaires de centaines de milliers de mots ne sont pas rares. Mais là encore, un premier travail est effectué pour l'élagage des mots les moins fréquents pour ne garder que des collections de l'ordre de 15000 vocables. Dans ce type d'application, le nombre d'individus peut être très élevé : des bases de plusieurs dizaines de milliers de documents sont maintenant accessibles. Des applications désormais classiques sont la recherche automatique dans un répertoire web ou encore la détection de mails indésirables (e.g. Guyon and Elisseeff, 2003).

D'une façon générale, la sélection de variables peut avoir plusieurs intérêts : faciliter la compréhension ou la visualisation de données, réduire les besoins physiques en stockage et dimensionnement, réduire les temps d'utilisation, améliorer les performances de prédiction. Chaque méthode favorise un ou plusieurs de ces aspects mais l'architecture de la majorité des algorithmes de sélection de variables suit un même schéma.

D'un point de vue structurel, une procédure de sélection de variables est composée de deux éléments fondamentaux (Liu and Motoda, 1998) :

- une mesure de pertinence utilisée pour mesurer l'influence d'un sous-ensemble de variables sur la variable à expliquer,
- et un algorithme de recherche dont le rôle est de parcourir des sous-ensembles de variables afin de trouver le sous-ensemble optimal ou quasi-optimal au sens de la mesure de pertinence choisie.

La mesure de pertinence peut se définir dans deux grands contextes de procédures de sélection de variables :

- la construction et la sélection d'un sous-ensemble de variables permettant de construire un bon modèle de prédiction : les *méthodes dépendantes d'un modèle* (wrapper methods),
- la recherche de variables pertinentes puis éventuellement leur ordonnancement : les *méthodes filtres* (filter methods).

Dans l'approche dépendante du modèle, l'algorithme de sélection de variables fonctionne autour de l'algorithme d'apprentissage. Il recherche le meilleur sous-ensemble de variables en utilisant l'algorithme de construction du modèle comme fonction d'évaluation des sous-ensembles. L'algorithme d'apprentissage est alors vu comme une boîte noire (Kohavi and John, 1997). Il est exécuté sur les données, généralement partitionnées préalablement en un ensemble d'apprentissage et un ensemble de test. Ces données sont sont aussi partitionnées sur les variables et l'algorithme d'apprentissage est exécuté sur différents sous-ensembles de variables. Des évaluations de la qualité, telles que le taux d'erreur de classement, de l'algorithme sur les différents sous-ensembles permettent de conserver le plus pertinent.

L'approche « filtre » sélectionne un sous-ensemble de variables en pré-traitement

de la recherche de modèle. Un de ses avantages est d'être complètement indépendante du modèle de données que l'on cherche à construire. Elle propose un sous-ensemble de variables satisfaisant pour expliquer une classe et ce, quelle que soit la structure des données qui se cache : le sous-ensemble est indépendant de l'algorithme d'apprentissage choisi. On parle aussi de sélection de variables non supervisée (Guérif, 2008; Mitra et al., 2002; Bennani and Guérif, 2007). De plus, les procédures filtres sont généralement moins coûteuses en temps de calcul puisqu'elles évitent les exécutions répétées des algorithmes d'apprentissage sur différents sous-ensemble de variables. En revanche, leur inconvénient majeur est qu'elles ignorent l'impact du sous-ensemble choisi sur les performances de l'algorithme d'apprentissage qui est appliqué en deuxième étape.

Contexte applicatif

Cette thèse s'est déroulée dans le cadre d'une convention CIFRE avec l'entreprise PerformanSe. L'entreprise PerformanSe conçoit, développe et commercialise des solutions logicielles dédiées à l'évaluation et à la gestion des compétences comportementales, à l'intention des professionnels des Ressources Humaines. Son logiciel phare est l'outil d'évaluation Echo (Philippé et al., 2004). Il s'agit d'un questionnaire de 70 items ipsatifs portant sur des situations rencontrées en milieu professionnel. À la fin de la passation, la personne évaluée est décrite selon 20 caractéristiques comportementales (ou 10 caractéristiques bipolaires). Des extensions de ce logiciel permettent de décrire la personne selon d'autres critères : activités dans le travail, perception de soi, perception de soi vue par les autres, etc.

L'entreprise PerformanSe dispose de nombreux jeux de données où les individus sont décrits par des variables comportementales. Ces jeux sont construits à partir des nombreuses passations anonymisées et stockées sur des serveurs. Le nombre de variables n'est pas de l'ordre de ce que l'on rencontre en bio-informatique mais reste trop élevé pour que le décideur puisse interpréter tout ce qu'il souhaite. En effet, à partir de plusieurs dizaines de variables comportementales, il a besoin de déduire une capacité pour une activité dans le milieu professionnel, un temps de retour à l'emploi ou encore une succession de postes en interne, par exemple. Notre travail est donc de lui proposer une vision simplifiée des données en réduisant le nombre de variables à étudier tout en lui présentant les liens entre variables s'il y en a. L'expert a ainsi entre ses mains des sous-ensembles de variables de différentes tailles qui doivent l'aider à comprendre le comportement qu'il étudie. Le fait de lui présenter les liens entre les variables est une aide supplémentaire pour la compréhension. En effet, une variable n'est pas vue comme une entité seule mais comme une représentante d'un groupe de variables vraisemblablement proches.

Contributions de la thèse

Notre étude porte sur un algorithme de sélection de variables basé sur une approche filtre dans le cadre de variables discrètes. En effet, notre cadre applicatif ne nous permet pas de décider à l'avance d'un modèle à utiliser pour nos données. Les besoins de l'entreprise sont divers et peuvent donc donner lieu à plusieurs représentations des données. Notre contribution porte sur les deux éléments fondamentaux d'une procédure de sélection de variables. Concernant la mesure de pertinence, nous utilisons l'information mutuelle. En effet, la pertinence de ce choix a déjà été mis en évidence dans de nombreux travaux (Yang and Moody, 1999; Torkkola, 2003; Fleuret, 2004; Hutter and Zaffalon, 2005; Kojadinovic, 2005). De plus, cette mesure se décompose de manière additive avec l'entropie de Shannon. C'est ce dernier point qui nous intéresse plus particulièrement puisque nous proposons d'approximer l'information mutuelle avec sa troncature 2-additive. Cette approximation permet de réduire les coûts de calcul en considérant qu'il est possible de reconstruire une bonne part de l'information mutuelle entre plusieurs variables aléatoires en ne connaissant l'information mutuelle que sur les singletons et les paires. La pertinence d'un sous-ensemble devient ainsi peu coûteuse à estimer et donc aisée à mettre en œuvre.

En ce qui concerne l'algorithme de recherche, les principaux algorithmes utilisés dans la littérature sont des recherches pas à pas ascendante ou descendante. Une recherche incrémentale ascendante trop simpliste montre vite ses limites en ne prenant pas en compte les variables déjà sélectionnées ; les variables redondantes se trouvent souvent retenues. Pour palier cet inconvénient, certains travaux (Fleuret, 2004) proposent de ne retenir à chaque étape que la variable qui apporte le plus « d'information » pour comprendre la classe en plus des autres variables déjà sélectionnées. Ainsi, une variable qui n'apporte rien de plus que celles déjà sélectionnées ne sera pas retenue même si prise individuellement, elle est très porteuse d'information. Nous nous plaçons dans cette optique : nous proposons, pour chaque taille de sous-ensembles de réévaluer tous les sous-ensembles de cette taille proposés par une procédure de génération. Ainsi, une variable non sélectionnée dans le meilleur sous-ensemble de taille 2 peut très bien être retenue pour former le meilleur sous-ensemble de taille 3. En revanche, cette démarche appliquée *in extenso* se heurte à un problème de coût de calcul ; il n'est pas envisageable de pouvoir évaluer tous les sous-ensembles possibles de chaque taille d'un ensemble de variables donné.

Nous proposons donc de structurer l'ensemble des variables grâce à une Classification Ascendante Hiérarchique des variables du jeu de données. Cette classification nous permet d'obtenir des classes de variables *a priori* « homogènes ». Nous pouvons donc raisonnablement supposer que les variables sont suffisamment redondantes pour n'en garder qu'une seule par classe dans la procédure de sélection. Notre procédure de génération des sous-ensembles à évaluer ne propose donc à l'évaluation par la mesure de pertinence que des sous-ensembles composés d'au plus une variable par classe de la classification.

Organisation du document

Le plan du document est le suivant.

Dans le chapitre 1, nous présentons la démarche générale d'une procédure de sélection de variables et détaillons à titre illustratif quelques algorithmes parmi les plus classiques.

Dans le chapitre 2, nous rappelons la définition de l'information mutuelle et présentons différentes approximations exposées dans la littérature. Nous complétons cette présentation par la définition d'autres coefficients qui pourraient être également utilisés en tant que mesures de pertinence d'un sous-ensemble de variables.

Dans le chapitre 3, nous rappelons les principes généraux d'une méthode de Classification Ascendante Hiérarchique, puis nous nous focalisons sur son utilisation moins classique pour la classification de variables.

Dans le chapitre 4, nous détaillons notre algorithme de sélection de variables, ClassAdd, en définissant tous les points nécessaires à son implémentation.

Dans le chapitre 5, nous présentons les bases de tests que nous utilisons pour nos expérimentations numériques :

1. un jeu test construit « à la main » pour le calibrage et la validation de notre approche,
2. des données issues de la base *University of California at Irvine repository* (D.J. Newman and Merz, 1998) classiquement utilisées en apprentissage,
3. les données comportementales issues de notre cadre applicatif en gestion des ressources humaines.

Dans le chapitre 6, nous étudions la robustesse des différentes mesures que nous utilisons (mesure de pertinence et mesures de qualité d'une partition dans le cadre d'une Classification Ascendante Hiérarchique).

Le chapitre 7 présente les comparaisons numériques effectuées sur les différents jeux de données.

1

La sélection de variables

Résumé

Nous présentons dans ce chapitre les notions nécessaires à la construction d'un algorithme de sélection de variables que nous illustrons par la présentation d'algorithmes existants.

Nous définissons, tout d'abord, les notations employées dans la suite du document (section 1.1). Dans cette même partie, nous rappelons la définition de la monotonie d'une mesure, d'une variable redondante et d'une variable non pertinente.

Dans la section 1.2, nous nous attachons à faire un état de l'art des procédures de génération des sous-ensembles, c'est-à-dire, de la façon de générer les sous-ensembles de variables à estimer. Les trois grandes approches sont la génération complète, avec une heuristique et aléatoire. Les algorithmes les plus classiques sont avec une heuristique, telles que la génération par ajout de variables et par suppression de variables.

Nous détaillons en section 1.3 la notion d'évaluation d'un sous-ensemble. C'est dans cette partie que nous précisons les notions de méthodes filtres et de méthodes dépendantes du modèle. Ces notions sont illustrées. Puis nous présentons les grandes catégories de mesures d'évaluation trouvées dans la littérature : les mesures de consistance, les mesures de précision et l'information mutuelle.

La notion de critère d'arrêt est présentée en section 1.4. Cet élément a son importance puisqu'il permet de décider quand stopper l'algorithme de sélection de variables c'est-à-dire de savoir quand le sous-ensemble de variables construit est suffisamment satisfaisant.

Enfin, nous terminons ce chapitre par la présentation d'algorithmes de sélection de variables existants (section 1.5). Nous illustrons les différents cas que nous avons présentés. L'utilisation de l'information mutuelle est plus largement abordée puisque c'est sur cette mesure que notre choix s'est porté.

Sommaire

Introduction

Un algorithme de sélection de variables suit toujours le fonctionnement présenté dans la figure 1.1 :

1. Une procédure de génération propose un sous-ensemble de variables issu de l'ensemble des variables d'origine.
2. Ce sous-ensemble est évalué *via* une mesure d'évaluation.
3. Si l'évaluation de ce sous-ensemble satisfait un critère de sélection, l'ensemble est retenu. Sinon, un nouvel ensemble est généré par la procédure de génération.

Après une présentation des notations que nous utilisons, nous définissons les différentes procédures de génération (ou algorithme de recherche) que nous trouvons dans la littérature. Puis, nous classons les différentes mesures de pertinence rencontrées. Ensuite, nous définissons la notion de critère d'arrêt de l'algorithme qui permet de choisir quand doit s'arrêter la procédure de sélection de variables. Enfin, nous illustrons ces sections avec les principaux algorithmes de sélection de variables existants.

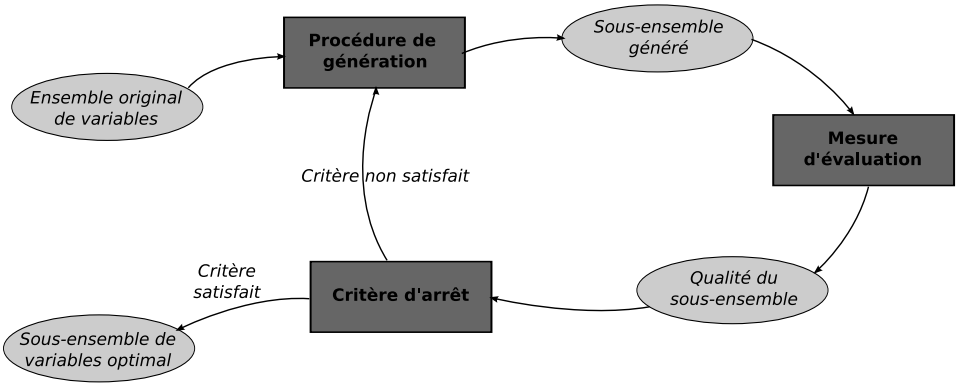


FIG. 1.1 – *Procédure de sélection de variables.*

1.1 Notations et prérequis

Soit $\aleph := \{X_1, \ldots, X_m\}$ un ensemble de variables discrètes potentiellement explicatives d'une variable aléatoire discrète Y dont les valeurs possibles sont $y_1, \ldots, y_l$. En pratique, cette variable Y correspond souvent à l'affectation des individus du jeu de données de n individus aux l classes d'un modèle. Un sous-ensemble de $\aleph$ de taille r se note $\mathbb{X}_r$, $r \leq m$. Chaque variable X_a, $a \in \{1, \ldots, m\}$, est une variable discrète ayant t_a modalités., Nous définissons $p(Y = y_i)$, $i \in \{1, \ldots l\}$, la probabilité *a priori* de y_i et $(p(\{X_a = x_{a_j}\}|y_i)$, $j \in \{1, \ldots, t_a\}$ et $a \in \{1, \ldots, m\}$, la probabilité conditionnelle d'avoir $\{X_a = x_{a_j}\}$ sachant que $Y = y_i$. La connaissance de ces deux probabilités suffit pour calculer les autres. En effet, grâce au théorème de Bayes, nous avons aussi :

$$p(y_i|\{X_a = x_{a_j}\}) = \frac{p(Y = y_i)p(\{X_a = x_{a_j}\}|y_i)}{p(\{X_a = x_{a_j}\})}$$

et

$$p(\{X_a = x_{a_j}\}) = \sum_{i \in \{1,\ldots,l\}} p(Y = y_i)(p(\{X_a = x_{a_j}\}|y_i)$$

Nous notons aussi ω une application de l'ensemble des parties de $\aleph$ dans $\mathbb{R}$ permettant l'évaluation d'un ensemble. Nous détaillons dans ce chapitre cette notion de mesure d'évaluation.

Définition 1.1.1 (Monotonie d'une mesure d'évaluation d'un ensemble) *Soit* $\omega : \mathcal{P}(\aleph) \rightarrow \mathbb{R}$*, une fonction d'évaluation d'un ensemble, soient* A *et* B *deux sous-ensembles de* $\aleph$ *tels que* $A \subset B$*,* ω *est monotone si et seulement si :*

$$\omega(A) \leq \omega(B)$$

Définition 1.1.2 (Variable non pertinente) *Soit la variable* $X_a \in \aleph$*,* $a \in \{1, \ldots, m\}$*,*

1. *X_a est une variable non pertinente si $\omega(X_a) = 0$.*

2. *X_a est une variable non pertinente conditionnellement à un ensemble $\mathbb{X} \subset \aleph$ si $\omega(\{X_a\} \cup \mathbb{X}) = \omega(\mathbb{X})$*

Une variable X_a est non pertinente si elle n'apporte rien pour expliquer Y ou si elle n'apporte rien de plus qu'un sous-ensemble donné de variables. Cette définition est équivalente à considérer que X_a et Y sont deux variables indépendantes dans le premier cas, et deux variables indépendantes conditionnellement à $\mathbb{X}$ dans le deuxième.

Définition 1.1.3 (Variable redondante) *La variable $X_a \in \aleph$, $a \in \{1, ..., m\}$ est redondante si*

$$\exists X_b \in \aleph, b \in \{1, ..., m\}, a \neq b, \omega(X_a|X_b) = 0$$

.
La notation $\omega(X_a|X_b)$ est utilisée pour représenter le fait que si l'on connaît la variable X_b alors la variable X_a n'est pas pertinente pour comprendre Y, elle n'apporte pas d'information en plus. Et plus généralement, $X_a \in \aleph$, $a \in \{1, ..., m\}$, est une variable redondante si

$$\exists \mathbb{Z} \subset \aleph - \{X_a\}, \mathbb{Z} \neq \emptyset, \omega(\mathbb{Z} \cup \mathbb{X}) = \omega(\{X_a\} \cup \mathbb{X}), \forall \mathbb{X} \subset \aleph - \{\mathbb{Z} \cup \{X_a\}\}$$

Une variable X_a est redondante si ce qu'elle apporte pour expliquer Y est déjà apporté par une autre variable.

1.2 Procédures de génération des sous-ensembles de variables candidats

Dans le cadre de la sélection de variables, la *procédure de génération* désigne la façon de générer l'ensemble de variables candidat à examiner (Liu and H.Motoda, 1998). Siedlecki and Sklansky (1988) parlent aussi de *procédure de recherche*. Le principe général consiste à générer successivement des sous-ensembles de variables à évaluer. Trois stratégies sont souvent considérées :

1. *ajout de variables* avec une initialisation avec l'ensemble vide,
2. *suppression de variables* avec une initialisation avec l'ensemble de toutes les variables,
3. tirage *aléatoire* d'un sous-ensemble de variables.

Pour cela, la procédure commence généralement avec un sous-ensemble vide et rajoute des variables, ou bien commence avec toutes les variables et en enlève, ou enfin commence avec un sous-ensemble aléatoire de variables.

Pour un ensemble de m variables, le nombre de sous-ensembles de variables candidats est $2^m - 1$. Même pour un nombre de variables raisonnable, le nombre de sous-ensembles à étudier est donc considérable. Pour affronter ce problème de taille de l'espace de recherche, les algorithmes se divisent en trois grandes familles : la recherche complète, la recherche avec une heuristique et la recherche aléatoire.

1.2.1 Recherche complète

Les procédures de génération complètes effectuent une recherche *complète* pour trouver le sous-ensemble de variables optimal au sens de la mesure d'évaluation choisie. Une méthode est dite complète ou exacte si elle assure de retourner toujours le sous-ensemble optimal. Il est important de distinguer une recherche complète d'une recherche exhaustive. En effet, une recherche exhaustive est toujours complète puisque qu'elle consiste à parcourir tous les sous-ensembles possibles. Ainsi, le ou les meilleur(s) sous-ensemble(s) est(sont) toujours évalué(s) et donc choisi(s). En revanche, la réciproque est fausse : dans certains cas, une recherche complète n'est pas exhaustive. Par exemple, si la mesure d'évaluation est monotone (rajouter une variable à un sous-ensemble ne peut pas faire diminuer la qualité de celui-ci, voir aussi la définition 1.1.1), nous n'aurons pas besoin de regarder tous les sous-ensembles (d'être exhaustif) pour retourner le sous-ensemble optimal. La recherche complète est donc encore coûteuse (de l'ordre de $O(2^m)$ (Dash and Liu, 1997)) mais elle évalue toujours moins de sous-ensembles que l'exhaustivité.

L'ensemble des sous-ensembles à évaluer peut être vu comme un treillis (Figure 1.2). Les deux méthodes de recherche exhaustive les plus courantes sont donc le parcours en profondeur d'abord et le parcours en largeur d'abord.

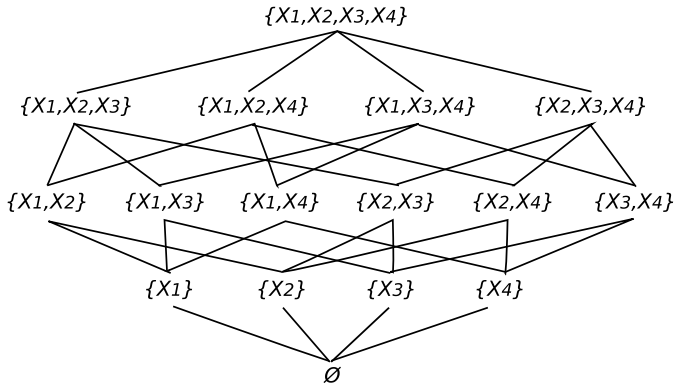


FIG. 1.2 – *Sous-ensembles de variables possibles à partir d'un ensemble de 4 variables.*

En ce qui concerne les recherches complètes, l'algorithme le plus classique trouvé dans la littérature est l'algorithme « Branch and Bound » de Narendra and Fukunaga (1977). C'est un algorithme de recherche en largeur d'abord auquel un seuil β est ajouté. Si un nœud de l'arbre de tous les sous-ensembles possibles dépasse β (respectivement

est inférieur à β), dans le cas d'une recherche descendante (respectivement ascendante) alors la recherche dans cette branche s'arrête et les nœuds fils du nœud courant sont supprimés de l'arbre des sous-ensembles possibles. Cet élagage de branches utilise la monotonie de la mesure d'évaluation.

1.2.2 Recherche avec une heuristique

Les parcours complets ou exhaustifs sont évidemment très coûteux en temps de calcul. L'utilisation d'une heuristique suit une stratégie de recherche que l'on sait non optimale mais qui assure la découverte d'une solution rapidement, que l'on souhaite proche de la solution optimale.

Les algorithmes classiques de recherche complète intègrent tous une heuristique pour les rendre opérationnels. Pour l'algorithme en largeur d'abord, l'heuristique « le meilleur en premier » consiste à ne prendre à chaque niveau de l'arbre que le meilleur sous-ensemble. C'est une stratégie qui ne prend en compte la notion de meilleur qu'à un seul niveau de l'arbre (c'est-à-dire qu'elle ne garde que le meilleur sous-ensemble de chaque taille). Augmenter le nombre de niveaux sur lequel on retient le meilleur sous-ensemble augmente aussi le coût en temps pour trouver une solution optimale. Une amélioration de cette heuristique consiste à prendre en compte le fait qu'un meilleur sous-ensemble à un niveau i de l'arbre ne provient pas forcément du meilleur sous-ensemble du niveau $i-1$ (le meilleur sous-ensemble de taille $i-1$ n'engendre pas forcément le meilleur sous-ensemble de taille i). L'heuristique « beam » (rayon) consiste donc à garder pour chaque niveau plusieurs sous-ensembles potentiellement « meilleurs » (Cover, 1974).

En ce qui concerne l'algorithme « Branch and Bound », Siedlecki and Sklansky (1988) ont proposé une heuristique : le but est de faire fonctionner l'algorithme dans le cas d'une mesure d'évaluation non monotone et non d'améliorer les performances en temps. Pour cela, le seuil β est conservé mais une valeur α lui est ajoutée pour traduire « l'intensité » de la non-monotonie. Ainsi, à chaque étape seuls les nœuds en dessous de $\beta+\alpha$ sont évalués. Pour la plupart des autres algorithmes rencontrés dans la littérature, la génération des sous-ensembles est incrémentale.

Génération par ajout de variables

L'algorithme 1 présente l'algorithme général de génération séquentielle ascendante. Les travaux autour de cet algorithme portent essentiellement sur le choix de la variable à ajouter à l'ensemble en construction. Par exemple, Liu and Wen (1993) ont travaillé sur la recherche des deux meilleurs suivants dans le cas où l'on suppose que deux variables ensemble ont plus d'influence que chacune prise séparément.

Génération par suppression de variables

Algorithme 1 L'algorithme de sélection de variables par ajout de variables.

Entrées:

$\aleph = \{X_1, \ldots, X_m\}$: l'ensemble des m variables potentiellement discriminantes
$r \leq m$: le nombre de variables à sélectionner
ω : une mesure de pertinence

Sorties:

$S \subset \aleph$:l'ensemble des variables sélectionnées

$S \leftarrow \emptyset$
pour $a = 1, \ldots, r$ **faire**
  $X_a \leftarrow \max_{X \in \aleph} \omega(X)$
  $S \leftarrow S \cup \{X_a\}$
  $\aleph \leftarrow \aleph \setminus \{X_a\}$
fin pour

Algorithme 2 L'algorithme de sélection de variables par suppression de variables.

Entrées:

$\aleph = \{X_1, \ldots, X_m\}$: l'ensemble des m variables potentiellement discriminantes
$r \leq m$: le nombre de variables à sélectionner
ω : une mesure de pertinence

Sorties:

$S \subset \aleph$:l'ensemble des variables sélectionnées

$S \leftarrow \aleph$
pour $a = 1, \ldots, m - r$ **faire**
  $X_a \leftarrow \min_{X \in \aleph} \omega(X)$
  $S \leftarrow S \setminus \{X_a\}$
  $\aleph \leftarrow \aleph \setminus \{X_a\}$
fin pour

Si le nombre de variables pertinentes, r, est inférieur à $m/2$ alors la démarche ascendante est plus rapide, sinon c'est la descendante la plus efficace (algorithme 2). Cependant, étant donné que r est rarement connu, le choix est rendu délicat. De plus, la principale limite de la génération séquentielle incrémentale (ascendante ou descendante) est le risque de tomber dans un minimum local duquel on ne pourra pas sortir puisque les variables enlevées ne peuvent être rajoutées et celles sélectionnées ne peuvent être enlevées. C'est pourquoi, des algorithmes hybrides ont été mis en oeuvre. Ils consistent à chaque itération à avoir le choix entre enlever une variable ou ajouter une variable au sous-ensemble en construction. Dans la pratique, ce choix est très dépendant de la sémantique des variables et ces algorithmes sont difficiles à automatiser complètement.

1.2.3 Génération aléatoire

Les algorithmes précédemment présentés sont déterministes. Une part de stochastique peut être introduite dans ces approches pour éviter un des principaux écueils des algorithmes précédents à savoir tomber dans un optimal local. L'algorithme RAND sélectionne à chaque tour un sous-ensemble généré aléatoirement s'il satisfait un critère de qualité et si sa cardinalité est inférieure au meilleur ensemble courant (Boddy and Dean, 1994). Le problème de ce genre d'algorithme reste le critère d'arrêt. Pour cela, on peut choisir de limiter le nombre d'itérations ou bien de s'arrêter dès lors que l'on a obtenu un ensemble ayant pour cardinalité le minimum que l'on s'était fixé (mais rien n'assure que l'algorithme ne se termine dans ce dernier cas). Siedlecki and Sklansky (1988) ont proposé d'appliquer un algorithme génétique ou bien une méthode de recuit simulé pour générer un meilleur sous-ensemble de variables pertinentes. Setiono and Liu (1997) ont utilisé un modèle de réseau de neurones avec élagage des branches de l'arbre. Le réseau de neurones et le recuit simulé retournent une solution unique ce qui n'est pas le cas de l'algorithme génétique ni de l'algorithme RAND.

Plusieurs implémentations de génération aléatoire de sous-ensembles de variables sont présentées dans Press et al. (1992). Ce point n'est pas à négliger puisque la performance de l'algorithme est étroitement liée à la qualité du générateur aléatoire.

1.3 Fonctions d'évaluation des sous-ensembles

La *fonction d'évaluation* permet de mesurer quantitativement la qualité d'un sous-ensemble généré par la procédure de recherche choisie (Liu and H.Motoda, 1998). Elle doit permettre de mesurer la qualité d'une variable ou d'un sous-ensemble de variables pour expliquer la variable que l'on cherche à comprendre. Les mesures généralement utilisées sont les mesures de consistance, les mesures de précision et les mesures basées sur l'information mutuelle.

1.3.1 Les méthodes filtres et les méthodes dépendantes du modèle

Langley (1994) a proposé de structurer les méthodes de sélection de variables en deux grands groupes : les méthodes *filtres* et les méthodes *dépendantes du modèle* (« wrapper »). Cette séparation est basée sur la dépendance ou non de l'algorithme de sélection de variables avec l'algorithme d'induction utilisé par le sous-ensemble sélectionné. Les méthodes filtres sont indépendantes du modèle que l'on choisit pour nos données. Elles retournent un ensemble de variables qui peut être utilisé pour construire n'importe quel modèle de données contrairement aux méthodes dépendantes du modèle. Celles-ci utilisent le modèle de données que l'on cherche à construire pour évaluer la qualité d'un sous-ensemble. Elles retournent donc le sous-ensemble optimal pour un modèle donné, comme par exemple un réseau de neurones (Leray and Gallinari, 1999).

La plupart des approches filtres classent les variables selon leur pouvoir individuel de prédiction de la classe qui peut être estimé de divers moyens tels que le score de Fisher (Furey et al., 2000), le test de Kolomogorov-Smirnov, le coefficient de corrélation de Pearson (Miyahara and Pazzani, 2000) ou encore l'information mutuelle (Torkkola, 2003; Battiti, 1994). Ce type de sélection basée sur le classement n'assure pas de dépendance faible entre variables : les sous-ensemble sélectionnés peuvent tout à fait comporter des variables redondantes et donc être moins informatifs. Prendre en compte la dépendance entre les variables semble donc être nécessaire. Par exemple, Ratanamahatana and D.Gunopulos (2003) ont proposé un critère basé sur les arbres de décisions. En effet, les attributs qui apparaissent dans un arbre binaire de type ceux construits par l'algorithme C4.5 sont soit individuellement porteurs d'information (variables en haut de l'arbre) soit conditionnellement porteurs d'information (variables plus bas dans l'arbre).

Nous présentons deux exemples d'algorithmes de Las Vegas proposées par Setiono and Liu (1997) (on rappelle qu'un algorithme de Las Vegas est un algorithme probabiliste qui résout exactement un problème avec une complexité moyenne polynômiale). L'un est une approche filtre et l'autre est une approche dépendante du modèle, et pourtant les deux algorithmes sont très proches. LVF (algorithme 3) est un algorithme Las Vegas pour une approche filtre de la sélection de variables. L'algorithme génère un sous-ensemble de variables aléatoirement et le taux d'inconsistance (définition 1.3.3) comme procédure d'évaluation permet de décider si le sous-ensemble est satisfaisant.

LVW (algorithme 4) est, lui, un algorithme Las Vegas pour l'approche dépendante du modèle de la sélection de variables. La mesure d'évaluation est la précision de l'algorithme d'apprentissage choisi (section 1.3.3).

Considérant cette séparation des méthodes de sélection de variables, Dash and Liu (1997) ont rangé les fonctions d'évaluation en cinq grandes catégories : les mesures de divergence, les mesures d'information, les mesures de dépendance, les mesures de consistance et les mesures de précision. Les trois premières catégories seront détaillées dans le chapitre 2.

Algorithme 3 L'algorithme LVF, Las Vegas Filter (Setiono and Liu, 1997).

Entrées:

$\aleph = \{X_1, \ldots, X_m\}$: l'ensemble des m variables potentiellement discriminantes
U : le taux d'inconsistance pour mesure de pertinence
max : le nombre maximum d'itérations possibles
γ : un seuil d'inconsistance maximale

Sorties:

L : une liste de bons sous-ensemble équivalents

$L \leftarrow \emptyset$
$C_{best} \leftarrow m$
pour $i = 1, \ldots, max$ **faire**
  $S \leftarrow$ un sous-ensemble de variables aléatoire
  $C \leftarrow card(S)$
  si $C \leq C_{best}$ et $U(S) \leq \gamma$ **alors**
    $S_{best} \leftarrow S$
    $C_{best} \leftarrow C$
    $L \leftarrow S$
  sinon
    si $C = C_{best}$ et $U(S) \leq \gamma$ **alors**
      $L \leftarrow L \cup S$
    finsi
  finsi
fin pour
Renvoyer$L$

Algorithme 4 L'algorithme LVW, Las Vegas Wrapper (Setiono and Liu, 1997).

Entrées:

$\aleph = \{X_1, \ldots, X_m\}$: l'ensemble des m variables potentiellement discriminantes
AA : un algorithme d'apprentissage
max : le nombre maximum d'itérations possibles

Sorties:

L : une liste de bons sous-ensemble équivalents

$L \leftarrow \emptyset$
$A_{best} \leftarrow$ précision de AA en utilisant $\aleph$
pour $i = 1, \ldots, max$ **faire**
  $S \leftarrow$ un sous-ensemble de variables aléatoire
  $A \leftarrow$ précision de AA en utilisant S
  si $A \geq A_{best}$ **alors**
    $S_{best} \leftarrow S$
    $A_{best} \leftarrow A$
    $L \leftarrow S$
  sinon
    si $A = A_{best}$ **alors**
      $L \leftarrow L \cup S$
    finsi
  finsi
fin pour
Renvoyer$L$

1.3.2 Les mesures de consistance

Avant de présenter *le taux d'inconsistance*, la principale mesure de consistance (Liu and H.Motoda, 1998), nous précisons la notion d'inconsistance pour deux individus.

Définition 1.3.1 (Inconsistance) *Deux individus sont inconsistants s'ils ont la même description sur chaque variable mais qu'ils appartiennent à des classes différentes.*

Par exemple, $o_1 = \{0101\}$ et $o_2 = \{0100\}$ sont deux individus inconsistants si l'on considère que la dernière variable représente leur classe : les trois premières variables ont la même valeur mais pas la dernière. Les individus ayant la même description sur chaque variable sont rassemblés dans un groupe d'inconsistance.

Définition 1.3.2 (Quantité d'inconsistance) *La quantité d'inconsistance, pour un groupe d'individus inconsistants donné, est définie comme la différence entre le nombre d'individus de ce groupe et la cardinalité de la classe contenant le plus d'individus inconsistants.*

Par exemple, si l'on a un groupe de 30 individus inconsistants et parmi eux, 5 appartiennent à la classe 1 et 25 appartiennent à la classe 2, alors le nombre d'inconsistance est égal à $30 - 25 = 5$. Un jeu de données est généralement composé de plusieurs groupes d'individus inconsistants.

Définition 1.3.3 (Taux d'inconsistance) *Le taux d'inconsistance est défini par la somme de toutes les quantités d'inconsistance divisée par le nombre total d'individus du jeu de données.*

Le taux d'inconsistance est inversement proportionnel au pouvoir discriminant : en effet, un sous-ensemble de variables ayant un taux d'inconsistance élevé signifie que ces variables ne permettent pas de bien prédire la classe et donc que ce sous-ensemble n'est pas un bon ensemble discriminant. Le calcul du taux d'inconsistance a une complexité en $O(n)$ avec n le nombre d'individus (Liu and H.Motoda, 1998).

1.3.3 Les mesures de précision

Les mesures de précisions sont utilisées lorsque l'on définit *a priori* un modèle des données. Dans ce cas là, la sélection de variables sert à optimiser le processus en simplifiant les calculs par une diminution du nombre de variables à prendre en compte dans le modèle. En théorie, un algorithme d'apprentissage doit être doté d'une mesure de précision permettant d'évaluer la qualité du modèle construit. Les mesures de précision sont généralement utilisées avec un algorithme de type recherche séquentielle descendante (Algorithme 2). A chaque étape, une variable est enlevée et l'on vérifie que le modèle est toujours suffisamment précis. Ce type d'algorithme s'arrête soit quand il n'y a plus

de variables à enlever soit quand la précision est jugée trop mauvaise. Une mesure de précision classique est le *le taux d'erreur* des données reconstruites par le modèle (e.g. Liu and H.Motoda, 1998), c'est-à-dire le nombre d'individus mal classés.

1.3.4 L'information mutuelle

L'information mutuelle est une mesure issue de la théorie de l'information. Elle mesure à la fois l'information qu'apporte une variable aléatoire sur une autre et la réduction d'incertitude sur une variable aléatoire grâce à la connaissance d'une autre. Elle se note généralement I ($\hat{I}$ est sa version estimée à partir des données) et est détaillée dans le chapitre 2.

1.4 Un critère d'arrêt

Le *critère d'arrêt* permet à la procédure de sélection de variables de s'arrêter. En effet, la plupart des fonctions d'évaluations rencontrées dans la littérature sont monotones. Ainsi, sans critère d'arrêt, le meilleur sous-ensemble serait toujours l'ensemble complet des variables de départ.

Le critère d'arrêt peut être lié à la procédure de recherche ou bien à la mesure d'évaluation (Dash and Liu, 1997). Dans le premier cas, le critère d'arrêt est soit la taille prédéfinie du sous-ensemble à sélectionner, soit un nombre fixé d'itérations de l'algorithme de sélection de variables. Dans le deuxième cas, un critère d'arrêt lié à la mesure d'évaluation est soit une différence de qualité entre deux ensembles non significative (l'ajout ou la suppression d'une variable n'améliore pas la qualité du sous-ensemble), soit un seuil pour la fonction d'évaluation à atteindre. Si la distribution empirique de la mesure d'évaluation est connue, un bon critère d'arrêt est alors l'invraisemblance de la valeur de l'évaluation. Cette invraisemblance est mesurée grâce à un test statistique.

1.5 Algorithmes existants

Nous présentons ici quelques algorithmes de sélection de variables de la littérature qui nous paraissent illustrer la variété des stratégies mises en œuvre. Pour avoir plus de détails, les travaux de Liu and H.Motoda (1998) et de Dash and Liu (1997) offrent un classement précis des algorithmes de base. Dans la littérature, les algorithmes de sélection de variables se divisent en deux grandes catégories : les algorithmes d'ordonnancement des variables et les algorithmes de construction du plus petit ensemble de variables.

1.5.1 Les algorithmes d'ordonnancement de variables

Ces algorithmes retournent un classement des variables selon une mesure d'évaluation qui évalue chaque variable individuellement. La complexité de ce type d'algorithme est en $O(m * n + m^2)$ où m est le nombre de variables et n le nombre d'individus (Liu and H.Motoda, 1998). L'algorithme d'ordonnancement le plus cité dans la littérature est l'algorithme Relief de Kira and Rendell (1992b) (Algorithme 5). Cet algorithme choisit aléatoirement un individu dans un échantillon de taille fixée par l'utilisateur. Pour chaque individu tiré, l'algorithme recherche l'individu le plus proche de lui appartenant à sa classe (« near hit ») et l'individu le plus proche n'étant pas dans la même classe que lui (« near miss »). L'idée intuitive de cet algorithme est qu'une variable est d'autant plus pertinente qu'elle distingue bien un individu et son « near miss » et d'autant moins pertinente qu'elle distingue bien un individu et son « near hit ». À partir de là, un poids est donné à chaque variable. À la fin de l'algorithme, à chacune des variables est donc associée une valeur qui permet de classer ces variables. L'algorithme garde toutes les variables dont le poids est supérieur à un seuil fixé par l'utilisateur.

Un autre critère d'arrêt usuel pour les algorithmes d'ordonnancement de variables est de fixer le nombre de variables, r, que l'on souhaite garder et de ne conserver ainsi les r premières variables du classement. La principale limite des algorithmes d'ordon-

Algorithme 5 L'algorithme Relief (Kira and Rendell, 1992b)

Entrées:
$\aleph = \{X_1, \ldots, X_m\}$: l'ensemble des m variables potentiellement discriminantes
τ : un seuil de pertinence
ech :un échantillon des données de taille $tailleEch$

Sorties:
w :un vecteur de poids pour les variables de taille m
$w \leftarrow 0$
pour $i = 1, \ldots, tailleEch$ **faire**
  $I \leftarrow$ un individu de ech choisi aléatoirement
  $H \leftarrow$ le « near-hit » de I
  $J \leftarrow$ le « near-miss » de I
  pour $j = 1, \ldots, m$ **faire**
    $w(j) \leftarrow w(j) - \frac{diff(j,I,H)^2}{tailleEch} + \frac{diff(j,I,J)^2}{tailleEch}$
    /* diff calcule la différence entre les valeurs des variables pour deux individus donnés */
  fin pour
fin pour
Renvoyer les variables dont le poids dans w est supérieur à τ

nancement de variables est qu'ils ne prennent pas en compte les variables redondantes.

En effet, deux variables redondantes vont être à la suite dans le classement des variables (la quantité d'information qu'elles apportent est très proche) et donc être choisies successivement.

1.5.2 Les algorithmes de construction du plus petit sous-ensemble de variables

En pratique, on ne connaît pas toujours le nombre de variables pertinentes et donc l'application d'algorithmes de classement est délicate. C'est pour cela que les algorithmes de construction du plus petit sous-ensemble de variables retournent un ensemble minimal de variables pertinentes et aucune différenciation n'est faite entre les variables.

Les méthodes complètes

L'algorithme FOCUS proposé par Almuallim and Dietterich (1991) est un algorithme avec recherche complète (Algorithme 6). Il considère tous les sous-ensembles possibles en partant des ensembles de plus petite taille, les singletons. Dès que FOCUS trouve un ensemble qui satisfait la mesure de consistance, il s'arrête. FOCUS retourne donc le plus petit sous-ensemble qui suffit pour déterminer la classe des individus. La complexité en temps est de l'ordre de $O(n)$, avec n le nombre d'individus. Dès que le nombre de variables pertinentes est supérieur à $n/2$, le coût est exhorbitant. La version de base telle que nous la présentons n'autorise aucun bruit sur les données mais des heuristiques ont été proposées par la suite (Almuallim and Dietterich, 1994).

Algorithme 6 L'algorithme FOCUS (Almuallim and Dietterich, 1991).

Entrées:

$\aleph = \{X_1, \ldots, X_m\}$: l'ensemble des m variables potentiellement discriminantes
U : le taux d'inconsistance pour mesure de pertinence

Sorties:

$S \subset \aleph$:le plus petit ensemble qui satisfait le taux d'inconsistance nul

$S \leftarrow \emptyset$
pour $i = 1, \ldots, m$ **faire**
  pour chaque sous-ensemble S de taille i **faire**
    si $U(S) = 0$ **alors**
      Renvoyer$S$
    finsi
  fin pour
fin pour

L'algorithme ABB de Liu and H.Motoda (1998) est une version automatique de l'algorithme « Branch and bound » (Algorithme 7). On parle d'automatique car le seuil

est déterminé automatiquement et non prédéfini. L'algorithme débute avec l'ensemble complet des variables. On enlève une variable à la fois en utilisant un parcours en profondeur d'abord jusqu'à ce qu'aucune des variables ne puisse plus être supprimée puisque le critère d'inconsistance est satisfait.

Algorithme 7 L'algorithme ABB, Automatic Branch and Bound (Liu and H.Motoda, 1998)

Entrées:
$\aleph = \{X_1, \ldots, X_m\}$: l'ensemble des m variables potentiellement discriminantes
U : le taux d'inconsistance pour mesure de pertinence
S_1, S_2 : des sous-ensembles de $\aleph$
Q : une pile vide
L : une liste pour stocker les ensembles satisfaisant le taux d'inconsistance

Sorties:
$S \subset \aleph$:le plus petit ensemble de L qui satisfait le taux d'inconsistance

$L \leftarrow \aleph$
$\delta \leftarrow U(\aleph)$
pour chaque variable X_i de $\aleph$ **faire**
  $S_1 \leftarrow \aleph - X_i$
  empiler S_1 dans Q
fin pour
Tant que Q n'est pas vide **faire**
  dépiler Q dans S_2
  si $U(S_2) \leq \delta$ **alors**
    $L \leftarrow L + S_2$
    $ABB(S_2)$
  finsi
fin tant que
$S \leftarrow$ le plus petit ensemble de L
Renvoyer$S$

Les méthodes avec une heuristique

Cette catégorie de méthodes de sélection de variables est sans doute la catégorie la plus fournie. En effet, la plupart des travaux trouvés dans la littérature consistent à prendre un algorithme existant de sélection de variables et à l'améliorer en terme de performance de calcul ou d'évaluation de la qualité d'un sous-ensemble grâce à une heuristique. L'algorithme Relief(Algorithme 5), par exemple, a été de nombreuses fois amélioré (Kononenko, 1994; Kira and Rendell, 1992a). Nous allons donc ne présenter qu'un très bref aperçu des algorithmes de sélection de variables utilisant une heuris-

tique. L'algorithme DTM (Cardie, 1993) se sert des arbres de décision pour sélectionner les variables (Algorithme 8). L'algorithme C4.5 (Quinlan, 1993) est appliqué sur un ensemble d'apprentissage et les variables apparaissant dans l'arbre généré sont sélectionnées. De nombreuses autres approches utilisent les variables apparaissant dans un arbre de décision comme heuristique : les travaux de Pudil et al. (1994) ou bien de Poggi and Tuleau (2006).

Algorithme 8 L'algorithme DTM, Decision Tree Method (Cardie, 1993)

Entrées:
$\aleph = \{X_1, \ldots, X_m\}$: l'ensemble des m variables potentiellement discriminantes

Sorties:
$T \subset \aleph$: un sous-ensemble de $\aleph$

$T \leftarrow \emptyset$
$C4.5$ sur un ensemble d'apprentissage
$T \leftarrow$ les variables apparaissant dans l'arbre
Renvoyer T

Les méthodes non déterministes

Les algorithmes LVF et LVW déjà présentés (Algorithmes 3 et 4) sont deux algorithmes classiques non déterministes. Ces algorithmes sont efficaces et très simples à implémenter. De plus, les expérimentations numériques ont montré que le résultat est généralement proche du sous-ensemble optimal dès lors que les ressources matérielles le permettent. Les méthodes d'algorithmes génétiques et de recuit simulé ont également été testées pour la sélection de variables (e.g. Siedlecki and Sklansky, 1988).

Enfin, de nombreux travaux existent sur la construction d'algorithmes hybrides mixant plusieurs des algorithmes précédents en essayant de ne garder que les avantages de chacun.

1.5.3 L'utilisation de l'information mutuelle pour l'évaluation des sous-ensembles

Dans cette section, nous nous focalisons sur la mesure d'évaluation basée sur l'information mutuelle qui est utilisée dans le reste de la thèse. Nous présentons des algorithmes de sélection de variables utilisant cette mesure.

L'algorithme de Fleuret (2004)

Les travaux de Fleuret proposent un algorithme de sélection de variables basé sur l'information mutuelle conditionnelle (Algorithme 9). C'est une approche itérative par ajout de variables. La particularité de cet algorithme est la prise en compte des variables déjà sélectionnées. Une variable est considérée comme bonne si elle apporte suffisamment d'information sur la variable à expliquer et si cette information n'est apportée par aucune des variables déjà choisies. Plus formellement, une variable X' est bonne si l'information mutuelle entre X' et Y sachant X est suffisamment grande pour chaque variable X déjà choisie. La sélection de variables redondantes est ainsi évitée. En revanche, les interactions entre plus de deux variables ne sont pas étudiées. L'algorithme a été implémenté pour des variables booléennes.

Algorithme 9 L'algorithme proposé par Fleuret (2004)

Entrées:
$\aleph = \{X_1, \ldots, X_m\}$: l'ensemble des m variables booléennes potentiellement discriminantes
Y : la variable à expliquer

Sorties:
$T \subset \aleph$: un sous-ensemble de $\aleph$ de taille K

$T \leftarrow \emptyset$
$T \leftarrow X_a$ tel que $a = argmax_n(\hat{I}(Y; X_n))$
pour $k = 2, \ldots, K$ **faire**
  $T \leftarrow T + X_a$ tel que $a = argmax_n(min_{l<k}\hat{I}(Y; X_n|X_l))$
fin pour
Renvoyer T

Le critère $\hat{I}(Y; X_n|X_l)$ est faible si X_n n'apporte pas beaucoup d'information sur Y (variable non pertinente) ou bien si cette information est déjà apporté par X_l (variable redondante). Prendre le X_n maximisant ce critère minimisé assure que la nouvelle variable sera à la fois informative et non redondante en terme d'information apportée pour expliquer Y. Cet algorithme n'est pas très coûteux puisque le calcul nécessite au plus un tableau de contingence pour trois variables.

L'algorithme de Koller and Sahami (1996)

La méthode de Koller and Sahami (1996) se base sur l'idée qu'une variable, qui apporte peu ou pas du tout d'information en plus de celle apportée par un ensemble d'autres variables déjà sélectionnées, est soit redondante soit non pertinente. Par conséquent, elle doit être éliminée. Pour cela, les auteurs ont utilisé les chaînes de Markov. Un

sous-ensemble S est une chaîne de Markov pour la variable X_a si, connaissant S, X_a est conditionnellement indépendante de la variable à expliquer Y et de toutes les variables n'appartenant pas à S.

L'algorithme MIFS (Battiti, 1994)

Battiti (1994) a proposé d'utiliser l'information mutuelle dans son algorithme de sélection ascendante de variables : Mutual Information based Feature Selection. La probabilité jointe de X et Y, deux variables aléatoires, est obtenue grâce à l'algorithme de Fraser and Swinney (1986). Cet algorithme ne permet de calculer que l'information mutuelle entre un couple de variables et la variable à expliquer. L'algorithme étant ascendant, il est nécessaire de calculer l'information mutuelle entre une variable X_a et l'ensemble des variables déjà sélectionnées, $SelectVara - 1$. L'algorithme MIFS simplifie le calcul de l'information mutuelle d'un ensemble en choisissant une variable représentative de l'ensemble. Le nombre de variables est fixé à l'avance et à chaque étape, on choisit la variable qui maximise l'information mutuelle entre elle, l'ensemble des variables déjà sélectionnées et la variable à expliquer.

L'algorithme de Yang and Moody (1999)

Les travaux de Yang and Moody (1999) portent sur un algorithme de sélection de variables basé sur l'information mutuelle jointe. Ils utilisent l'information mutuelle conditionnelle. Pour chaque variable potentielle, l'algorithme calcule le gain d'information qu'elle apporte, c'est à dire l'information mutuelle de cette variable avec la variable à expliquer conditionnellement aux variables déjà sélectionnées. Une variable X_a est indépendante de la variable à expliquer Y si son information mutuelle conditionnée par les variables déjà choisies est nulle. À chaque étape, l'algorithme classe donc les variables restantes en terme d'information mutuelle conditionnelle et sélectionne la variable apportant le plus d'information en plus de celles déjà choisies.

L'algorithme de Hutter and Zaffalon (2005)

Hutter and Zaffalon (2005) utilisent leur approximation de l'information mutuelle dans un cadre bayésien (section 2.3.2)pour une procédure de sélection de variables par filtre. Ils proposent deux approches : le filtre descendant (BF) et le filtre ascendant (FF) dans une méthode pas à pas.

Le filtre descendant supprime une variable X_a si $p(I(X_a;Y) < \epsilon|n) > \rho$ où ϵ est un seuil arbitraire bas et ρ est une probabilité arbitraire haute. Littéralement, l'algorithme BF rejette la variable X_a si l'hypothèse « $I(X_a;Y)$ petit » est très probable.

Le filtre ascendant, à l'inverse, ajoute une variable X_a si $p(I(X_a;Y) > \epsilon|n) > \rho$ où ϵ est un seuil arbitraire bas et ρ est une probabilité arbitraire haute. Une variable X_a est retenue par l'algorithme AF si l'hypothèse « $I(X_a;Y)$ grand » est très probable.

Conclusion

Dans ce chapitre, après avoir présenté théoriquement ce qu'est un processus de sélection de variables, nous avons illustré notre propos avec des algorithmes de sélection de variables proposés dans la littérature.

Nous voyons qu'une multitude de pistes de recherche s'offrent dans les algorithmes de sélection de variable. Ne voulant pas présumer d'un modèle sur nos données, nos travaux portent sur une procédure filtre de sélection de variables. De plus, devant la taille des données dont nous disposons, une méthode exhaustive n'est pas envisageable. Nous proposons donc d'utiliser une heuristique de recherche afin de limiter les coûts en temps de calcul. L'algorithme ClassAdd que nous proposons dans ce document est une procédure filtre de sélection de variables basée sur une heuristique de recherche que nous détaillerons dans la suite.

2

L'information mutuelle en tant que mesure de dépendance

Résumé

Ce chapitre définit la notion d'information mutuelle dans le cadre de la mesure de liaison entre variables.

Nous définissons, tout d'abord, les notations employées dans la suite du document (section 2.1).

Dans la section 2.2, nous rappelons le fondement de la mesure d'information mutuelle à partir de la notion d'incertitude et de l'entropie de Shannon dans le cadre de la théorie de l'information. En fin de partie, nous présentons la notion de généralisation de l'information mutuelle qui permet de mesurer l'information mutuelle entre plus de deux variables aléatoires. Cette notion est importante pour comprendre l'approximation k-additive de l'information mutuelle que nous faisons par la suite.

Nous abordons dans la section 2.3, les deux estimations de l'information mutuelle que nous comparons dans nos expérimentations. L'estimation classique se calcule à partir du tableau de contingence. L'estimation par une approche bayésienne est plus détaillée car moins utilisée.

Dans la dernière partie (section 2.4), nous rappelons les mesures classiques de liaison entre variables de la littérature : la notion de corrélation, de mesure de divergence et de mesure de dépendance. Cette section se conclut par la présentation d'une normalisation de l'information mutuelle afin qu'elle puisse être utilisée comme mesure de dépendance, ce que nous faisons dans nos travaux.

Sommaire

Introduction

L'information mutuelle est une mesure classique de liaison entre variables dans les problèmes de sélection de variables. Son utilisation en tant que mesure de pertinence a déjà été considérée à de nombreuses reprises dans la littérature (e.g. Hutter and Zaffalon, 2005). Elle est utilisée sous plusieurs formes : information mutuelle classique ou information mutuelle conditionnelle pour la prise en compte des variables préalablement choisies.

Dans la première section, nous définissons les notations utilisées par la suite. En seconde section, nous présentons les bases de la théorie de l'information, de l'entropie de Shannon à la définition de l'information mutuelle. Dans la troisième section, nous étudions les façons d'estimer l'information mutuelle. Enfin, en quatrième section, nous proposons une vue sur les principales autres mesures de liaisons entre variables trouvées dans la littérature.

2.1 Notations et prérequis

Nous introduisons tout d'abord des notations concernant les variables aléatoires qui sont utilisées dans la suite.

– Nous considérons X,Y et Z trois variables aléatoires discrètes prenant un nombre fini de valeurs dans respectivement $\{1,\ldots,t\}$,$\{1,\ldots,l\}$ et $\{1,\ldots,h\}$. X a donc t modalités, Y a l modalités et Z a h modalités.

- Les ensembles $\mathcal{X} = \{x_1, \ldots, x_t\}$, $\mathcal{Y} = \{y_1, \ldots, y_l\}$ et $\mathcal{Z} = \{z_1, \ldots, z_h\}$ désignent, respectivement, l'ensemble des états possibles de X, Y et Z.
- Les notations x_j, $j \in \{1, \ldots, t\}$, y_i, $i \in \{1, \ldots, l\}$ et z_k, $k \in \{1, \ldots, h\}$ représentent un état possible, respectivement, de X, Y et Z. Si nous avons besoin de désigner plusieurs états d'une variable aléatoire, nous indiçons les i, j, k par $1, 2, 3, \ldots$.
- Les distributions marginales de X, Y et Z sont notées respectivement p_X,p_Y et p_Z. Par exemple, la distribution p_X est un vecteur ligne $p_X = (p_1, ..., p_t)$ où $p_j = p(X = x_j)$ pour j allant de 1 à t.
- La distribution jointe de X et Y est notée $p_{(X,Y)}$ et $p_{j,i} = p((X = x_j) \cap (Y = y_i))$ pour j allant de 1 à t et i allant de 1 à l.
- La distribution de X conditionnellement à Y est notée $p_{(X|Y)}$ et on pose $p_{j|i} = p((X = x_j)|(Y = y_i))$ pour j allant de 1 à t et i allant de 1 à l.

Toutes les notations précédentes sont utilisées avec « ' » si nous avons besoin de variables ayant le même nombre de modalités. Par exemple, X' désigne une variable aléatoire ayant le même nombre de modalités, t, que X. Enfin, nous supposons un jeu de données ayant n individus. Les valeurs des variables X, Y et Z pour un individu u, $u \in 1, \ldots, n$, se notent respectivement u_X, u_Y et u_Z. Nous renvoyons à l'annexe A pour les rappels des propriétés de convexité et de convergence qui nous sont utiles dans la suite.

2.2 Les bases de la théorie de l'information

2.2.1 La notion d'incertitude

Considérons la variable aléatoire discrète X de distribution de probabilité p_X. L'incertitude sur la prédiction de son état dépend directement de sa distribution p_X. Par exemple, si p_X est une masse de Dirac sur $\mathcal{X}$, c'est-à-dire qu'il existe un état x_j, $j \in \{1, ..., t\}$, tel que $p_j = 1$, alors il n'existe aucune incertitude concernant la réalisation de X. En revanche, si p_X est la loi uniforme, tous les états de X sont équiprobables et l'incertitude sur la réalisation de X est donc maximale. L'état de X n'est alors pas du tout prévisible.

La notion d'incertitude sur la réalisation d'un évènement aléatoire peut aussi être reliée à la *quantité d'information* qu'apporte la réalisation de cet évènement. En effet, si l'on reprend les deux exemples précédents, dans le cas d'une masse de Dirac, la réalisation de l'évènement $\{X = x_j\}$ est certaine et elle n'apporte donc aucune information sur la variable aléatoire X. En revanche, dans le cas où p_X est une loi uniforme, la réalisation d'un évènement apporte beaucoup d'information sur la variable aléatoire puisque son état n'était pas du tout prévisible.

La notion d'incertitude sur la réalisation d'un évènement aléatoire ou d'information apportée par la réalisation d'un évènement est donc directement liée à la probabilité pour cet évènement d'être réalisé. Il est donc naturel de définir une mesure d'incertitude comme une application de l'ensemble des distributions de probabilités dans l'ensemble des réels. De plus, elle doit satisfaire trois grandes propriétés :

1. l'incertitude liée à un évènement doit être d'autant plus grande que la probabilité de réalisation de cet évènement est faible,
2. l'incertitude liée à l'évènement certain est nulle,
3. l'incertitude liée à deux évènements indépendants est la somme des incertitudes liées à chacun d'entre eux.

Une telle mesure estime *a priori* l'incertitude qui règne sur la réalisation d'un évènement et *a posteriori* l'information apportée par la réalisation de cet évènement. Ces deux points de vue sont aussi pertinents l'un que l'autre. La mesure classique, l'entropie de Shannon (1948) d'une variable aléatoire X, est en fait une moyenne des incertitudes calculées sur les évènements $\{X = x_j\}$, $j \in \{1, ..., t\}$.

2.2.2 L'entropie de Shannon

Le concept d'*entropie d'une distribution de probabilité* a été initialement proposé par Shannon (1948). Dans le cas de variables aléatoires discrètes, la notion d'entropie se confond à la fois avec celle d'*incertitude* liée à la variable et avec celle de *quantité d'information* apportée par une réalisation de cette variable. En effet, si nous voulons observer l'état de la variable aléatoire X, son entropie s'interprète comme l'incertitude quant à l'état dans lequel elle va se trouver. En revanche, si l'état de X est connu, l'entropie permet alors de quantifier l'information que nous avons obtenue en observant la variable aléatoire X.

Définition 2.2.1 (Entropie) *L'entropie H d'une variable aléatoire discrète X est définie par*

$$H(X) = -\sum_{j=1}^{t} p_j \log(p_j)$$

La convention $0log0 = 0$ est utilisée. Elle se justifie par le prolongement par continuité en 0 de la fonction $x \mapsto log(x)$. Nous retrouvons que l'entropie de Shannon ne dépend pas uniquement des valeurs prises par la variable aléatoire X mais bien des probabilités pour la variable X d'avoir telle valeur.

L'entropie de Shannon s'interprète aussi souvent comme l'*information moyenne* sur X. En effet, elle correspond à l'espérance mathématique de $-\log(p_j)$. Définissons la fonction I_X, pour tout $j \in \{1, ..., t\}$ par $I_X(x_j) = -\log(p_j)$; I_X peut être vue comme une fonction mesurant l'incertitude ou l'information relative aux états de la variable aléatoire X. En effet, I_X vérifie bien les propriétés attendues pour une mesure d'incertitude :

1. Si x_j a une probabilité très faible alors $I_X(x_j)$ est très élevée, la réalisation de cet état est très informative.
2. Si x_j est un état de probabilité 1, $I_X(x_j) = -\log(1) = 0$.
3. Si nous considérons deux états indépendants de X, x_{j1} etx_{j2},

$$\begin{aligned} I_X(x_{j1} \cap x_{j2}) &= -\log(p_{j1j2}) \\ &= -\log(p_{j1}.pj2) \\ &= -\log(p_{j1}) - \log(p_{j2}) \\ &= I_X(x_{j1}) + I_X(x_{j2}) \end{aligned}$$

L'entropie H correspond ainsi à une moyenne des incertitudes sur les états de la variable X :

$$H(X) = E[I_X].$$

La définition de l'entropie d'une seule variable aléatoire s'étend naturellement à une paire de variables aléatoires. Cette extension est naturelle puisqu'un couple de variables aléatoires peut être vu comme un seul vecteur aléatoire à deux dimensions.

Définition 2.2.2 (Entropie jointe) *L'entropie jointe H de deux variables aléatoires discrètes X et Y est définie par*

$$H(X,Y) = -\sum_{j=1}^{t}\sum_{i=1}^{l} p_{j,i}\log(p_{j,i})$$

ce qui donne sous la forme d'une espérance,

$$H(X,Y) = -E[\log(p_{(X,Y)})]$$

Définition 2.2.3 (Entropie conditionnelle) *L'entropie conditionnelle se définit aussi naturellement par*

$$H(Y|X) = \sum_{ji=1}^{t} p_j H(Y|\{X = x_j\})$$

L'entropie mesure l'incertitude sur la réalisation d'une variable aléatoire. Ainsi, l'incertitude sur la réalisation de deux variables aléatoires peut se voir comme l'incertitude sur la première à laquelle on ajoute l'incertitude sur la deuxième sachant la première. Nous rappelons donc le théorème suivant :

Théorème 2.2.1

$$H(X,Y) = H(X) + H(Y|X)$$

Preuve. Ce théorème se démontre facilement en utilisant la définition de l'entropie.

$$\begin{aligned} H(X,Y) &= -\textstyle\sum_{j=1}^{t}\sum_{i=1}^{l} p_{j,i}\log(p_{j,i}) \\ &= -\textstyle\sum_{j=1}^{t}\sum_{i=1}^{l} p_{j,i}\log(p_j p_{i|j}) \\ &= -\textstyle\sum_{j=1}^{t}\sum_{i=1}^{l} p_{j,i}\log(p_j) - \sum_{j=1}^{t}\sum_{i=1}^{l} p_{j,i}\log(p_{i|j}) \\ &= -\textstyle\sum_{j=1}^{t} p_j\log(p_j) - \sum_{j=1}^{t}\sum_{i=1}^{l} p_{j,i}\log(p_{i|j}) \\ &= H(X) + H(Y|X) \end{aligned}$$

□

Ainsi, si les deux variables X et Y sont indépendantes, l'incertitude sur la réalisation du couple (X, Y) est la somme des incertitudes. En effet, $H(Y|X) = H(Y)$.

Nous rappelons maintenant quelques propriétés intéressantes de l'entropie de Shannon.

Propriété 2.2.1 *L'entropie de Shannon est toujours positive :*

$$H(X) \geq 0$$

En effet, $\forall j \in \{1, \ldots, t\}, 0 \leq p_j \leq 1$ puisque p_j est une probabilité. Il s'ensuit naturellement que $\log(p_j) \leq 0$ et donc que $H(X) \geq 0$. De plus, $H(X) = 0$ si et seulement si p_X est une masse de Dirac.

Propriété 2.2.2 *L'entropie de Shannon est bornée :*

$$H(X) \leq \log(t)$$

De plus, l'entropie est maximale lorsque p_X suit une loi uniforme et alors $H(X) = \log(t)$. Dans ce cas là, l'entropie de Shannon se confond avec l'information de Hartley (1928) qui est définie comme le logarithme du nombre d'états possibles de X dont la probabilité est non nulle.

Propriété 2.2.3 *L'entropie conditionnelle est majorée :*

$$H(X|Y) \leq H(X)$$

avec égalité si et seulement si X et Y sont indépendantes.

L'entropie conditionnelle se généralise et en considérant trois variables aléatoires X, Y et Z, nous avons

$$H(X, Y|Z) = H(X|Z) + H(Y|X, Z)$$

Théorème 2.2.2 (Règle de chaînage pour l'entropie) *Soient $X_1, \ldots, X_m$, m variables aléatoires, alors*

$$H(X_1, \ldots, X_m) = \sum_{a=1}^{m} H(X_a|X_{a-1}, \ldots, X_1)$$

Preuve. La preuve s'obtient par réécriture successive du théorème 2.2.1

$$\begin{aligned}
H(X_1, X_2) &= H(X_1) + H(X_2|X_1) \\
H(X_1, X_2, X_3) &= H(X_1) + H(X_2, X_3|X_1) \\
&= H(X_1) + H(X_2|X_1) + H(X_3|X_2, X_1) \\
&\vdots \\
H(X_1, \ldots, X_m) &= H(X_1) + H(X_2|X_1) + \cdots + H(X_m|X_{m-1}, \ldots, X_1) \\
&= \textstyle\sum_{a=1}^{m} H(X_a|X_{a-1}, \ldots, X_1)
\end{aligned}$$

□

Une application directe de cette règle de chaînage est le théorème suivant :

Théorème 2.2.3 (Majoration de l'entropie) *Soient $X_1, \ldots, X_m$, m variables aléatoires, alors*

$$H(X_1, \ldots, X_m) \leq \sum_{a=1}^{m} H(X_a)$$

Preuve. D'après le théorème 2.2.2, nous avons

$$\begin{array}{rcl} H(X_1, \ldots, X_m) & = & \sum_{a=1}^{m} H(X_a | X_{a-1}, \ldots, X_1) \\ & \leq & \sum_{a=1}^{m} H(X_a) \end{array}$$

par la propriété 2.2.3. Nous avons l'égalité si et seulement si X_a est indépendante de $X_{a-1}, \ldots, X_1$ pour tout $a \in 1, \ldots, m$, c'est-à-dire, si et seulement si les X_a sont indépendants. □

2.2.3 L'entropie relative

L'entropie d'une variable aléatoire permet donc de mesurer l'incertitude que l'on a sur cette variable aléatoire. Elle traduit aussi la quantité d'information nécessaire pour décrire la variable aléatoire. L'entropie relative est un concept directement lié à cette notion d'information. En effet, elle mesure la distance entre deux distributions de probabilité. L'entropie relative $D(p_X||q_X)$ permet de mesurer l'erreur faite par l'hypothèse que la distribution de la variable aléatoire X est q_X alors que la vraie distribution est p_X.

Définition 2.2.4 (Entropie relative) *L'entropie relative ou distance de Kullback-Leibler entre deux distributions de probabilités p_X et q_X est définie par*

$$D(p_X||q_X) = \sum_{j=1}^{t} p_j \log \frac{p_j}{q_j}$$

avec la convention $0 \log \frac{0}{q} = 0$ et $0p \log \frac{p}{0} = \infty$.

Définition 2.2.5 (Entropie relative conditionnelle) *L'entropie relative conditionnelle $D(p_{(Y|X)}||q_{(Y|X)})$ est la moyenne des entropies relatives entre les distributions de probabilités conditionnelles $p_{(Y|X)}$ et $q_{(Y|X)}$ pondérée par la distribution de probabilité p_X :*

$$D(p_{(Y|X)}||q_{(Y|X)}) = \sum_{j=1}^{t} p_j \sum_{i=1}^{l} p_{i|j} \log \frac{p_{i|j}}{q_{i|j}}$$

Théorème 2.2.4 (Règle de chaînage pour l'entropie relative)

$$D(p_{(X,Y)}||q_{(X,Y)}) = D(p_X||q_X) + D(p_{(Y|X)}||q_{(Y|X)})$$

Preuve.

$$\begin{aligned} D(p_{(X,Y)}||q_{(X,Y)}) &= \sum_{j=1}^{t}\sum_{i=1}^{l} p_{j,i}\log\frac{p_{j,i}}{q_{j,i}} \\ &= \sum_{j=1}^{t}\sum_{i=1}^{l}\log\frac{p_j p_{i|j}}{q_j q_{i|j}} \\ &= \sum_{j=1}^{t}\sum_{i=1}^{l}\log\frac{p_j}{q_j} + \sum_{j=1}^{t}\sum_{i=1}^{l}\log\frac{p_{i|j}}{q_{i|j}} \\ &= D(p_X||q_X) + D(p_{(Y|X)}||q_{(Y|X)}) \end{aligned}$$

□

Théorème 2.2.5 (Inégalité de l'information) *L'entropie relative est toujours positive*

$$D(p_X||q_X) \geq 0$$

et elle est nulle si et seulement si $p_j = q_j$, $\forall j \in \{1,\ldots,t\}$.

Preuve.

$$\begin{aligned} -D(p_X||q_X) &= -\sum_{j=1}^{t} p_j\log\frac{p_j}{q_j} && (1)\\ &= \sum_{j=1}^{t} p_j\log\frac{q_j}{p_j} && (2)\\ &\leq \log\sum_{j=1}^{t} p_j\frac{q_j}{p_j} && (3)\\ &= \log\sum_{j=1}^{t} q_j && (4)\\ &\leq \log 1 && (5)\\ &= 0 && (6) \end{aligned}$$

L'inégalité (3) s'obtient à partir de l'inégalité de Jensen. En effet, la fonction $t \mapsto -\log t$, est strictement convexe. Il y a donc égalité si et seulement si $\frac{q_j}{p_j} = 1$, $\forall j \in \{1,\ldots,t\}$, c'est-à-dire, si et seulement si $p_j = q_j$, $\forall j \in \{1,\ldots,t\}$. Nous avons donc bien $D(p_X||q_X) = 0$ si et seulement si $p_j = q_j$, $\forall j \in \{1,\ldots,t\}$. □

Corollaire 2.2.1

$$D(p_{(Y|X)}||q_{(Y|X)}) \geq 0$$

et elle est nulle si et seulement si $p_{i|j} = q_{i|j}$, $\forall i \in \{1,\ldots,l\}$ *et* $j \in \{1,\ldots,t\}$, $p_j > 0$.

2.2.4 L'information mutuelle

L'information mutuelle mesure la quantité d'information qu'apporte une variable aléatoire sur une autre. C'est la réduction d'incertitude sur une variable aléatoire grâce à la connaissance d'une autre.

Définition 2.2.6 (Information mutuelle) *Soient X et Y deux variables aléatoires, l'information mutuelle entre X et Y, $I(X;Y)$, est l'entropie relative entre la distribution jointe $p_{(X,Y)}$ et le produit des distributions marginales p_X et p_Y.*

$$\begin{aligned} I(X;Y) &= \sum_{j=1}^{t}\sum_{i=1}^{l} p_{j,i}\log\frac{p_{j,i}}{p_j p_i} \\ &= D(p_{(X,Y)}||p_X p_Y) \\ &= E[\log(\tfrac{p_{(X,Y)}}{p_X p_Y})] \end{aligned}$$

En considérant l'information mutuelle comme la *H-information* obtenue à partir de l'entropie de Shannon, la définition 2.2.6 peut se récrire de la façon suivante :

Définition 2.2.7 (Information mutuelle)

$$I(X;Y) = H(Y) - H(Y|X)$$

L'information mutuelle est symétrique : $I(X;Y)$ est la réduction d'incertitude sur X due à la connaissance de Y et elle est égale à la réduction d'incertitude sur Y due à la connaissance de X, $I(Y;X)$. De plus, nous avons vu que $H(X,Y) = H(X) + H(Y|X)$ (Théorème 2.2.1), nous pouvons donc en déduire

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

Nous retrouvons bien que l'information mutuelle d'une variable aléatoire avec elle-même est l'entropie de la variable aléatoire :

$$I(X;X) = H(X) + H(X) - H(X,X) = H(X)$$

C'est pour cette raison que l'entropie est parfois désignée comme la *self-information* d'une variable aléatoire.

Définition 2.2.8 (Information mutuelle conditionnelle) *L'information mutuelle conditionnelle, $I(X;Y|Z)$, se définit comme la réduction d'incertitude sur X grâce à la connaissance de Y sachant que Z, une troisième variable aléatoire, est réalisée :*

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z)$$

L'information mutuelle satisfait la règle de chaînage.

Théorème 2.2.6 (Règle de chaînage pour l'information mutuelle) *Soient $\{X_1, \ldots, X_m\}$, m variables aléatoires alors*

$$I(X_1, \ldots, X_m; Y) = \sum_{a=1}^{m} I(X_a; Y | X_{a-1}, X_{a-2}, \ldots, X_1)$$

Preuve.

$$\begin{array}{rcl} I(X_1, ..., X_m; Y) & = & H(X_1, X_2, \ldots, X_m) - H(X_1, X_2, \ldots, X_m|Y) \\ & = & \sum_{a=1}^{m} H(X_a|X_{a-1}, \ldots, X_1) - \sum_{a=1}^{m} H(X_a|X_{a-1}, \ldots, X_1, Y) \\ & = & \sum_{a=1}^{m} I(X_a; Y|X_1, X_2, \ldots, X_{a-1}) \end{array}$$

□

Nous présentons maintenant quelques propriétés de l'information mutuelle.

Propriété 2.2.4 *L'information mutuelle entre deux variables aléatoires X et Y est positive :*

$$I(X;Y) \geq 0$$

et elle est nulle si et seulement si X et Y sont indépendantes.

Cette propriété découle directement du théorème 2.2.5 sur l'inégalité de l'information et nous avons pour les mêmes raisons :

Propriété 2.2.5 *Considérons trois variables aléatoires, X, Y et Z, alors*

$$I(X;Y|Z) \geq 0$$

avec l'égalité si et seulement si X et Y sont indépendantes conditionnellement à Z.

2.2.5 Généralisation de l'information mutuelle

Dans cette partie, nous introduisons une nouvelle notation qui aide à généraliser l'information mutuelle à un ensemble de variables. L'information mutuelle est indicée par le nombre de variables aléatoires qu'elle prend en paramètres. Par exemple, l'information mutuelle classique I entre deux variables aléatoires s'écrira I_2. La généralisation pour $r \geq 2$ variables aléatoires est notée I_r.

Définition 2.2.9 (Information mutuelle entre trois variables aléatoires) *Cette définition a été proposée par Abramson (1963). L'information mutuelle entre trois variables aléatoires X, Y et Z est définie par :*

$$\begin{aligned} I_3(X;Y;Z) = H(X) + H(Y) + H(Z) \\ - H(X,Y) - H(X,Z) - H(Y,Z) + H(X,Y,Z). \end{aligned}$$

Et plus généralement,

Définition 2.2.10 (Information mutuelle entre plus de deux variables aléatoires) *Soient $r \geq 2$ et $X_1, \ldots, X_r$, r, variables aléatoires, l'information mutuelle entre $r \geq 2$ variables aléatoires est définie par :*

$$I_r(X_1;\ldots;X_r) = \sum_{k=1}^{r} \sum_{\{a_1,\ldots,a_k\} \subseteq \{1,\ldots,r\}} (-1)^{k+1} H(X_{a_1},\ldots,X_{a_k}).$$

2.3 Estimation

Dans toute cette partie, les versions estimées de l'information mutuelle (respectivement des distributions), à partir des données seront notées $\hat{I}$ (respectivement $\hat{p}$).

2.3.1 Estimation classique

En considérant la définition 2.2.6 de l'information mutuelle, il apparaît clairement que l'information mutuelle de deux variables aléatoires X et Y est fonction de leur distribution jointe $p(X, Y)$. Cette distribution est classiquement estimée par un maximum de vraisemblance (proportions). Un estimateur naturel de l'information mutuelle est alors

$$\hat{I}(X;Y) = D(\hat{p}_{X,Y}||\hat{p}_X\hat{p}_Y)$$

2.3.2 Estimation bayésienne

Nous venons de voir que l'information mutuelle entre deux variables aléatoires X et Y peut se calculer avec la probabilité jointe de ces deux variables. Classiquement, la probabilité empirique est utilisée. Ce calcul présente deux grandes limites :

- l'estimation classique $\hat{I}(X;Y)$ ne donne aucune information sur sa précision,
- $\hat{I}(X;Y) = 0$ peut avoir deux origines que sont une dépendance réelle entre les deux variables aléatoires ou bien une fluctuation due à l'échantillon de données de taille finie.

Le cadre bayésien permet de répondre à ces problèmes par l'utilisation d'une probabilité *a priori* du second ordre $\pi(\hat{p})$ qui prend en compte une certaine incertitude sur $\hat{p}$. De cette probabilité *a priori*, le calcul de la probabilité *a posteriori* $\pi(\hat{p}|n)$ est possible et ainsi $\pi(I(X;Y)|n)$ peut être obtenue.

Distribution de l'information mutuelle

Dans le cadre d'une approche bayésienne, nous utilisons une fonction de probabilité *a priori*, π, du second ordre pour représenter la probabilité d'avoir $\hat{p}$. De là, nous pouvons calculer $\pi(\hat{p}|n) \propto \pi(\hat{p}) \prod_{ij} \hat{p}_{ij}^{n_{ij}}$ car les n_{ij} sont distribués selon une loi multinomiale. Ainsi se définit une probabilité *a posteriori* telle que (Hutter and Zaffalon, 2005) :

$$\pi(I|n) = \int \delta(\hat{I}(X;Y) - I(X;Y)\pi(\hat{p}|n)d^{tl}\hat{p}$$

La distribution δ permet de restreindre l'intégrale sur les $\hat{p}$ tels que $\hat{I} = I$.

Espérance et variance sous une distribution de Dirichlet

Nous n'avons pas d'information explicite sur le comportement de $\hat{p}$. Dans ces cas là, une probabilité *a priori* non-informative $\pi(\hat{p})$ est souvent utilisée. Ce type de probabilité *a priori* mène généralement à l'utilisation d'une distribution de Dirichlet (annexe B.1) *a posteriori* (Agresti, 2002). Les travaux de Hutter and Zaffalon (2005) permettent de définir l'espérance, E, et la variance, V, de l'information mutuelle sous des probabilités

a posteriori de Dirichlet :

$$E[I(X;Y)] \simeq J + \frac{(t-1)(l-1)}{2(n+1)}$$

$$V[I(X;Y)] \simeq \frac{1}{n+1}(K - J^2)$$

avec

$$J = \sum_{j=1}^{t}\sum_{i=1}^{l} \frac{n_{ij}}{n} \log \frac{n_{ij}n}{n_i n_j}$$

et

$$K = \sum_{j=1}^{t}\sum_{i=1}^{l} \frac{n_{ij}}{n} \left(\log \frac{n_{ij}n}{n_i n_j}\right)^2$$

Approximation de l'information mutuelle

Maintenant que nous avons l'espérance et la variance de l'information mutuelle, il nous reste à trouver une loi pour approximer l'information mutuelle. Trois choix sont possibles. Choisir une approximation par une loi normale est le premier choix possible. En effet, le théorème central-limite (annexe A.0.5) assure que $\pi(I|n)$ converge vers une loi normale d'espérance $E[I]$ et de variance $V[I]$. Cependant, sachant que I est non négative, nous pouvons donc plutôt considérer l'approximation de $\pi(I|\hat{p})$ par une loi Gamma (loi proche de la loi du χ^2). Enfin, étant donné que I peut être normalisée et majorée par 1, la distribution Beta (annexe B.2) est une autre candidate naturelle. Les travaux de Hutter and Zaffalon (2005) proposent une comparaison de l'approximation de l'information mutuelle par chacune de ces lois. Les trois approximations sont très correctes avec une petite préférence pour l'approximation par la loi Beta. Nous utilisons donc cette dernière dans nos expérimentations sur l'estimation de l'information mutuelle.

2.4 D'autres mesures de liaison entre variables

2.4.1 Généralités sur les corrélations

Définition 2.4.1 (Corrélation entre deux variables) *Il y a corrélation entre les variables aléatoires X et Y s'il y a dépendance en moyenne : à $X = x_j$ fixé, $j \in \{1, \ldots, t\}$, la moyenne de Y est fonction de x_j. Rappelons que la non corrélation n'est pas forcément l'indépendance.*

Si la liaison entre la moyenne de Y et x_j est approximativement linéaire, nous sommes dans le cas de la corrélation linéaire entre deux variables. Le coefficient classique mesurant le caractère plus ou moins linéaire de cette liaison est le coefficient de Pearson.

Définition 2.4.2 (Coefficient de Bravais-Pearson) *Le coefficient de Bravais-Pearson pour les deux variables aléatoires X et Y, r_{XY}, est défini par :*

$$r_{XY} = \frac{\frac{1}{n}\sum_{u=1}^{n}\left(u_X - \overline{X}\right)\left(u_Y - \overline{Y}\right)}{s_X s_Y}$$

où $\overline{X}$, respectivement $\overline{Y}$, désignent les moyennes observées sur les n individus de X, respectivement Y, et s_X et s_Y sont les écarts-types observées de X et Y.

Nous rappelons que r ne mesure que le caractère linéaire d'une liaison. Un faible coefficient r traduit seulement une non corrélation linéaire mais pas forcément une absence de corrélation.

Une autre étude de corrélation classique est l'étude de la corrélation des rangs. Elle permet de quantifier les liaisons du type « les deux variables varient dans le même sens ». Les deux principaux coefficients sont les coefficients de Spearman et de Kendall.

Définition 2.4.3 (Coefficient de corrélation des rangs de Spearman) *Le coefficient de Spearman, r_s, est défini par :*

$$r_s = \frac{cov\left(r_X, r_Y\right)}{s_{r_X} s_{r_Y}}$$

où r_X, respectivement r_Y, sont des vecteurs de taille n, le nombre d'individus, contenant le classement des n individus selon les variables X, respectivement Y. s_{r_X} et s_{r_Y} sont les écarts-types de ces deux vecteurs de classement.

Définition 2.4.4 (Coefficient de corrélation des rangs de Kendall) *Le coefficient de Kendall, τ, est défini par :*

$$\tau = \frac{2S}{n\left(n-1\right)}$$

Pour calculer S, il nous faut regarder les $\frac{n(n-1)}{2}$ couples distincts d'individus. Pour chaque couple, si ses valeurs sur la variable X sont dans le même ordre que ses valeurs sur la variable Y alors on ajoute 1 à S sinon, si les deux classement discordent, on retire 1.

L'indice de Kendall est un indice relationnel. Dans le cadre de la méthode AVL, il y a toute une famille de coefficients d'association entre variables relationnelles probabilistes (Lerman, 1992).

2.4.2 Les mesures de divergence

Les *mesures de divergence* sont aussi connues sous le nom de mesures de séparabilité, mesures de discrimination ou encore mesures de dissimilarités entre distributions de probabilités.

Par exemple, dans un problème à deux classes représentées par la variable à expliquer Y, la variable X sera préférée à la variable Z, si la variable X discrimine mieux les deux classes de Y que la variable Z ne le fait. Dans le contexte d'ordonnancement des variables, nous présentons les deux principales mesures de divergence permettant de le faire. La divergence directe, DD, d'une variable aléatoire X avec une variable de classes Y, est définie par :

$$DD(X,Y) = \sum_{j\in\{1,\ldots,t\}} p_j \sum_{i\in\{1,\ldots,l\}} p(y_i|\{X=x_j\}) \log \frac{p(y_i|\{X=x_j\})}{p(y_i)}$$

La divergence directe traduit ce qu'apporte la variable X pour prédire la classe de l'individu. Il en est de même de la variance, V, définie comme suit :

$$V(X) = \sum_{j\in\{1,\ldots,t\}} p(X=x_j) \sum_{i\in\{1,\ldots,l\}} p(y_i) \left(p(y_i \mid X=x_j) - p(y_i)^2 \right)$$

Ces deux mesures de divergence sont étroitement liées au problème de la sélection de variables. Mais, plus généralement, la notion de divergence est très utilisée en statistiques et probabilités pour mesurer la dissimilarité entre deux distributions de probabilités. Une des premières propositions de mesure de distance entre probabilités est attribuée à Pearson (1904). Soient p et q, les deux distributions de probabilités de deux variables aléatoires discrètes X et X' ayant chacune t modalités : $(x_1,\ldots,x_t)$ et $(x'_1,\ldots,x'_t)$

$$\chi^2(p,q) = \sum_{j=1}^{t} \frac{(p(X=x_j) - q(X'=x'_j))^2}{q(X'=x'_j)}$$

avec la convention $\frac{(0-0)^2}{0} = 0$. La divergence du χ^2 est à l'origine du test statistique classique du χ^2. L'autre grande mesure de dissimilarité entre distributions de probabilités est la divergence de Kullback and Leibler (1951), KL, que nous avons déjà présentée sous le nom d'entropie relative (Définition 2.2.4) :

$$KL(p,q) = \sum_{j=1}^{t} p(X=x_j) \log \frac{p(X=x_j)}{q(X'=x'_j)}$$

avec la convention $0 \log \frac{0}{0} = 0$.

Ces deux mesures de divergence sont parfois abusement qualifiées de distances alors qu'elles ne sont pas symétriques et ne vérifient pas l'inégalité triangulaire.

2.4.3 Les mesures de dépendance

Les mesures de dépendance permettent de mesurer le degré de *dépendance fonctionnelle* entre deux variables aléatoires.

		X				
		x_1	x_2	...	x_t	
Y	y_1	n_{11}	n_{12}	...	n_{1t}	$n_{1.}$
	y_2	n_{21}	n_{22}	...	n_{2t}	$n_{2.}$
	...	...	...	...	...	...
	y_l	n_{l1}	n_{l2}	...	n_{lt}	$n_{l.}$
		$n_{.1}$	$n_{.2}$	...	$n_{.t}$	n_{lt}

TAB. 2.1 – *Tableau de contingence de deux variables aléatoires X et Y.*

Définition 2.4.5 (Dépendance fonctionnelle) *Deux variables aléatoires X, à t modalités, et Y, à l modalités, sont fonctionnellement dépendantes s'il existe une fonction f de $\mathbb{R}^m$ dans $\mathbb{R}^l$ telle que pour tout couple (x_j, y_i), $j \in \{1, \ldots, t\}, i \in \{1, \ldots, l\}$, $y_i = f(x_j)$.*

La notion de dépendance fonctionnelle a été précisée par Joe (1989b). Une mesure de dépendance fonctionnelle, d, entre deux variables aléatoires, X (t modalités) et Y (l modalités), a pour principales propriétés :

- *Indépendance* : $d(X, Y) = 0$ si et seulement si X et Y sont deux variables aléatoires indépendantes.
- *Dépendance fonctionnelle* : si X et Y sont fonctionnellement dépendantes alors $d(X, Y) = 1$.
- *Normalisation* : $d(X, Y)$ est compris entre 0 et 1.
- *Invariance* : pour toutes bijections f_X de $\mathbb{R}^t$ dans $\mathbb{R}^t$ et f_Y de $\mathbb{R}^l$ dans $\mathbb{R}^l$, $d(X, Y) = d(f_X(X), f_Y(Y))$.

La notion de dépendance fonctionnelle correspond en fait à la notion d'*écart à l'indépendance*. La mesure de dépendance la plus utilisée est le χ^2 d'écart à l'indépendance. Cette mesure est en fait la divergence du χ^2 dans le cas particulier où p représente la distribution jointe de X et Y et q représente la distribution jointe de X et Y si X et Y étaient indépendantes. Considérons le tableau de contingence de deux variables aléatoires Y et X où n_{ij} désigne le nombre d'individus possédant les propriétés $Y = y_i$ et $X = x_j$ (Tableau 2.1). L'indépendance empirique entre Y et X se traduit par $n_{ij} = \frac{n_{i.}n_{.j}}{n}$ avec n le nombre total d'individus et $n_{i.}$, respectivement $n_{.j}$, le nombre total d'individus possédant la caractéristique $Y = y_i$, respectivement $X = x_j$.

Le χ^2 d'écart à l'indépendance

Généralement, la mesure de liaison suivante est adoptée :

$$d^2(X, Y) = \sum_{j=1}^{t} \sum_{i=1}^{l} \frac{\left(n_{ij} - \frac{n_{i.}n_{.j}}{n}\right)^2}{\frac{n_{i.}n_{.j}}{n}}$$

d^2 s'annule bien dans le cas où les variables X et Y sont indépendantes. De plus, la mesure de liaison est bornée et nous avons le résultat suivant :

$$\frac{d^2}{n} \leq \inf\left(t-1; l-1\right)$$

Cette borne est atteinte dans le cas de la dépendance fonctionnelle entre X et Y.

Les mesures associées au χ^2

Le coefficient de liaison d^2 n'étant pas normé, divers coefficients compris entre 0 (indépendance) et 1 (liaison fonctionnelle) ont été proposés. Le coefficient de contingence de Pearson (1904), C, est défini par :

$$C = \left(\frac{d^2}{n+d^2}\right)^{1/2}$$

Le coefficient de Tschuprow, T vaut :

$$T = \left(\frac{d^2}{n\sqrt{(t-1)(l-1)}}\right)^{1/2}$$

Le coefficient de Cramer, se note V :

$$V = \left(\frac{d^2}{n\inf\left(t-1; l-1\right)}\right)^{1/2}$$

Un autre calcul intéressant associé à d^2 est la *contribution au* χ^2. Elle compare le tableau de contingence avec le tableau de contingence de l'indépendance. Ainsi, pour chaque case du tableau de contingence, il est possible de calculer le coefficient $\frac{(n_{ij}-n_{i.}n_{.j})^2}{\frac{n_{i.}n_{.j}}{n}} \cdot \frac{1}{d^2}$. Ce coefficient met en évidence les associations significatives entre la catégorie i de la variable aléatoire Y et la catégorie j de la variable aléatoire X.

En fait, $d^2(X,Y)$ est une réalisation d'une variable aléatoire $D^2(X,Y)$ qui suit approximativement une loi du $\chi^2_{(t-1)(l-1)}$ sous l'hypothèse d'indépendance entre X et Y.

Autres mesures de dépendance

Les indices dérivés du χ^2 ne sont pas les seules mesures de dépendance utilisables. De nombreux autres indices existent. Nous citerons parmi ceux-là le G^2 ou χ^2 de vraisemblance :

$$G^2 = 2\sum_{j=1}^{t}\sum_{i=1}^{l} n_{ij} \ln\left(\frac{\frac{n_{ij}}{n_{i.}n_{.j}}}{n}\right)$$

qui sous l'hypothèse d'indépendance suit une loi du $\chi^2_{(t-1)(l-1)}$.

Nous avons aussi le τ_b de Goodman et Kruskal qui est un indice non symétrique de dépendance :

$$\tau_{bY|X} = \frac{\sum_{j=1}^{t}\sum_{i=1}^{l}\frac{n_{ij}^2}{nn_{.j}} - \sum_{i=1}^{l}\left(\frac{n_{i.}}{n}\right)^2}{1 - \sum_{i=1}^{l}(\frac{n_{i.}}{n})^2}$$

Normalisation de l'information mutuelle

Enfin, pour faire le lien avec les mesures de dépendance, des versions normalisées de l'information mutuelle ont été proposées. Soient X et Y, deux variables aléatoires, si l'on interprète l'information mutuelle comme la H-information obtenue à partir de l'entropie de Shannon (Définition 2.2.7), une première normalisation est le *coefficient d'incertitude asymétrique*, U_H, (Särndal (1974)) :

$$U_H(X;Y) = \frac{I(X;Y)}{H(X)} = \frac{H(X) - H(X|Y)}{H(X)}$$

Le coefficient U_H d'interprète donc comme la réduction relative d'incertitude sur X due à la connaissance de Y.

La version symétrique de ce coefficient, S, est :

$$S(X;Y) = \frac{I(X;Y)}{\frac{1}{2}\left[H(X) + H(Y)\right]}$$

Ce coefficient S prend bien ses valeurs dans $[0,1]$ mais ne caractérise pas la dépendance fonctionnelle. En effet, X et Y fonctionnellement dépendants n'impliquent pas que S soit égal à 1. C'est pourquoi Joe (1989b) a proposé le coefficient suivant, $\tilde{I}_d$:

$$\tilde{I}_d(X;Y) = \frac{I(X;Y)}{\min[H(X), H(Y)]}$$

Joe (1989b) a démontré que son coefficient était bien normalisé et que $\tilde{I}_d(X;Y) = 1$ si et seulement si X et Y sont deux variables fonctionnellement dépendantes.

Conclusion

Nous avons détaillé dans ce chapitre la notion d'information mutuelle étroitement liée à la notion d'entropie. Nous avons vu que des travaux sur l'estimation de cette mesure de dépendance existent ainsi que sur sa normalisation.

La mesure de pertinence liée à l'algorithme de sélection de variables que nous proposons, ClassAdd, est l'information mutuelle. Lors de nos expérimentations, nous considérons son estimation classique et son estimation bayésienne. Nous utilisons aussi cette

mesure, en version normalisée, comme distance entre variables pour la construction de la Classification Ascendante Hiérarchique sur l'ensemble des variables du jeu de données.

3

La classification de variables

Résumé

Nos travaux proposent de structurer l'ensemble des variables grâce à une classification de celles-ci. Ce chapitre présente donc l'algorithme de Classification Ascendante Hiérarchique et son application à la classification de variables.

Nous présentons, tout d'abord, quelques prérequis (section 3.1) pour la compréhension de la notion de classification ascendante. Nous rappelons ainsi les définitions d'une mesure de similarité, dissimilarité et d'une distance.

Dans la section 3.2, nous détaillons l'algorithme de Classification Ascendante Hiérarchique de la notion de hiérarchie à la stratégie d'agrégation. Nous terminons cette partie en présentant les deux principaux critères de qualité d'une classe que sont l'homogénéité et la séparation d'une partition.

Dans la section 3.3, nous présentons la classification de variables en détaillant notamment l'Analyse de Vraisemblance du Lien.

Sommaire

Introduction

Différentes méthodes de classification ont été proposées (Brucker and Barthélemy, 2007). Mais l'objectif le plus classiquement visé reste la construction d'une partition ou d'une suite de partitions emboîtées. Deux critères doivent être simultanément satisfaits : les classes obtenues doivent être homogènes (rassembler ce qui se ressemble) et bien séparées.

Une des approches la plus utilisée est sans doute la Classification Ascendante Hiérarchique. Elle consiste à fournir un ensemble de partitions en classes de moins en moins fines obtenues par regroupements successifs de parties. Son application première pour la classification d'individus s'est vite étendue aux variables avec notamment les travaux de Lerman (1981). C'est cette application aux variables qui nous intéresse pour nos travaux.

Après une présentation des notations et de quelques définitions basiques nécessaires pour ce chapitre, nous étudions la théorie liée à l'algorithme de Classification Ascendante Hiérarchique : son principe général, sa complexité et la notion de qualité d'une partition. En dernière section, nous présentons l'application de cet algorithme à la classification de variables.

3.1 Prérequis

Soit E un ensemble de n objets dont nous avons besoin de connaître les écarts deux à deux. Nous allons présenter les différentes notions d'éloignement entre deux objets de la moins contrainte à la plus contrainte en termes de propriétés.

Définition 3.1.1 (Mesure de similarité) *Une application, s, définie de $E \times E$ dans $\mathbb{R}^+$, est une mesure de similarité (ou ressemblance) si elle vérifie les propriétés suivantes :*

- ***Non-négativité***
$$\forall(i,j) \in E \times E, s(i,j) \geq 0$$
- ***Symétrie***
$$\forall(i,j) \in E \times E, s(i,j) = s(j,i)$$
- ***Majoration de la similarité***
$$\forall(i,j) \in E \times E, i \neq j, s(i,i) = s(j,j) \geq s(i,j)$$

Un indice de similarité normé, s^*, s'obtient facilement à partir de s en posant, par exemple :
$$s^*(i,j) = \frac{s(i,j)}{s(i,i)}, \forall(i,j) \in E \times E$$
s^* est alors une application de $E \times E$ dans $[0,1]$

Définition 3.1.2 (Mesure de dissimilarité) *Une application, d, définie de $E \times E$ dans $\mathbb{R}^+$, est une mesure de dissimilarité (ou dissemblance) si elle vérifie les propriétés suivantes :*
- ***Non-négativité***
$$\forall(i,j) \in E \times E, d(i,j) \geq 0$$
- ***Symétrie***
$$\forall(i,j) \in E \times E, d(i,j) = d(j,i)$$
- ***Nullité***
$$\forall i \in E, d(i,i) = 0$$

Les notions de similarité et de dissimilarité sont très liées. En effet, si s est un indice de similarité alors, l'application d, définie par :
$$d(i,j) = s(i,i) - s(i,j), \forall(i,j) \in E \times E$$
est un indice de dissimilarité. À partir d'une mesure de dissimilarité d, l'application d^* peut être définie de $E \times E$ dans $[0,1]$ comme indice de dissimilarité normé par :
$$d^*(i,j) = \frac{d(i,j)}{\max_{(i,j)\in E\times E} d(i,j)}$$
D'autres transformations ont été proposées dans la littérature.

Définition 3.1.3 (Distance) *Une distance, d, est une application de $E \times E$ dans $\mathbb{R}^+$ qui vérifie les propriétés suivantes :*
- ***Non-négativité***
$$\forall(i,j) \in E \times E, d(i,j) \geq 0$$
- ***Symétrie***
$$\forall(i,j) \in E \times E, d(i,j) = d(j,i)$$

– ***Identité***

$$\forall (i,j) \in E \times E, d(i,j) = 0 \Leftrightarrow i = j$$

– ***Inégalité triangulaire***

$$\forall (i,j,k) \in E^3, d(i,j) \leq d(i,k) + d(k,j)$$

3.2 La Classification Ascendante Hiérarchique

Une première version de l'algorithme de Classification Ascendante Hiérarchique est attribuée à Sokal and Sneath (1963). La base *Phenetics* est souvent citée comme une des premières taxonomies construite grâce à un algorithme numérique. Les organismes à classifier sont décrits par différentes caractéristiques numériques à partir desquelles un coefficient de similarité est calculé pour chaque paire d'organismes. Cette matrice de similarité permet de construire la taxonomie ou classification. Comme pour la taxonomie *Phenetics*, la plupart des algorithmes de classification ont pour point de départ une mesure de dissimilarité entre les objets à classer. Il existe une multitude de façons pour évaluer ces dissemblances et le choix de la dissimilarité aura une influence décisive sur les résultats. Nous renvoyons aux travaux de Brucker and Barthélemy (2007) pour une mise en perspective historique de cette question fondamentale.

3.2.1 Principe général

Pour effectuer une classification ascendante hiérarchique sur un ensemble E de n objets, il faut au préalable calculer les dissimilarités entre tous les objets, deux à deux. Nous noterons par la suite d cette mesure de dissimilarité ($d : E \times E \rightarrow \mathbb{R}^+$). L'algorithme procède alors par étapes successives de manière ascendante. À la première étape les deux objets les plus proches (le couple $(x,y) \in E \times E$ tel que $d(x,y) = \min_{(i,j) \in E \times E} d(i,j)$) sont regroupés. Il ne reste alors plus que $n-1$ objets de E (ou classe d'objets de $\mathcal{P}(E)$) à classer. Le processus est réitéré jusqu'à ce que tous les objets aient été réunis dans une seule classe. À la fin de chaque étape, les distances entre la classe nouvellement créée et les autres objets (ou classe d'objets) sont recalculées. Lorsque cela est terminé, un arbre hiérarchique de classification, le dendrogramme, est construit. Les nœuds de l'arbre représentent les regroupements successifs des objets dans les classes, et la hauteur de ces nœuds représente l'écart entre les deux objets (ou classe d'objets) fusionnés.

La notion de hiérarchie

Définition 3.2.1 (Hiérarchie de parties d'un ensemble) *Soit E un ensemble de n objets, une famille H de parties de E est une hiérarchie si :*

– *E et les singletons de E appartiennent à H*

– *$\forall A, B \in H$, $A \cap B \in \{A, B, \emptyset\}$: deux classes sont soit disjointes soit contenues l'une dans l'autre.*

– *Toute classe est la réunion des classes qui sont incluses en elle.*

Le dendrogramme est la représentation visuelle d'une hiérarchie. Par exemple, la Figure 3.1 représente la hiérarchie

$$\{\{i\},\{j\},\{k\},\{l\},\{m\},\{j,k\},\{l,m\},\{i,j,k\},\{i,j,k,l,m\}\}$$

pour l'ensemble $E = \{i, j, k, l, m\}$.

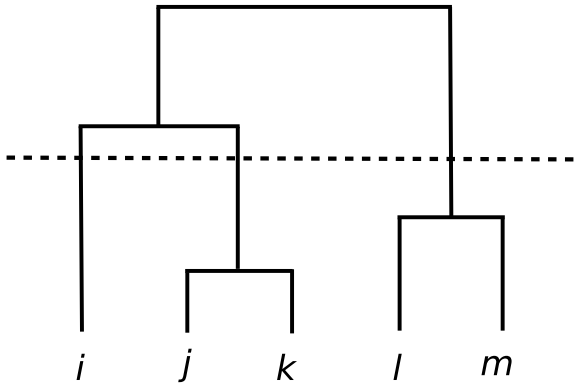


FIG. 3.1 – *Exemple de hiérarchie pour un ensemble* $E = \{i, j, k, l, m\}$.

Définition 3.2.2 (Partition compatible avec une hiérarchie) *Une partition d'un ensemble E est dite compatible avec une hiérarchie H si les classes de la partition sont des éléments de H.*

Par souci de simplicité, nous nous plaçons dans le cas où une partition compatible avec une hiérarchie est obtenue en coupant horizontalement le dendrogramme correspondant à cette hiérarchie. Par exemple, en se référant à la Figure 3.1, une partition compatible avec l'ensemble $E = \{i, j, k, l, m\}$ est la suivante : $\{\{i\},\{j,k\},\{l,m\}\}$.

Définition 3.2.3 (Hiérarchie indicée) *Une hiérarchie, H, est indicée s'il existe une application indice de H dans $\mathbb{R}^+$ croissante telle que :*
si $A \subset B$, $A, B \in H$, alors $indice(A) \leq indice(B)$.

Les indices sont aussi appelés *niveaux d'agrégation* : $indice(A)$, $A \in H$, est le niveau auquel tous les éléments de A ont été regroupés pour la première fois. L'arbre construit par une classification ascendante hiérarchique s'apparente à une hiérarchie indicée avec généralement comme niveaux d'agrégation la dissimilarité des deux parties regroupées à ce niveau. Par exemple, sur la Figure 3.2, nous avons $indice(i, j, k) = \delta(i, (j, k)) = 0,65$.

Partant de la partition discrète où chaque classe contient une seule variable, il s'agit d'établir une hiérarchie des agrégations faisant apparaître des classes et des sous-classes

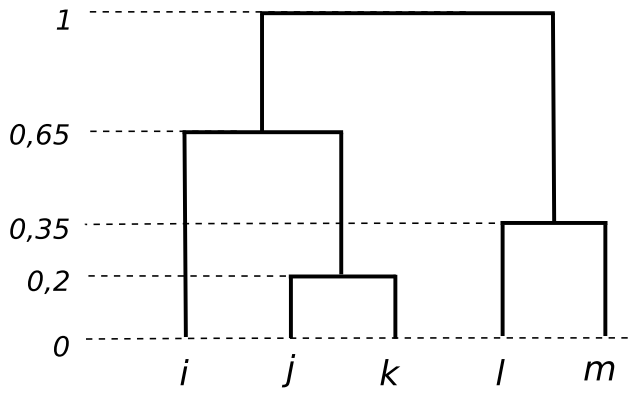


FIG. 3.2 – *Exemple de hiérarchie indicée pour un ensemble* $E = \{i, j, k, l, m\}$.

qui sont cohérentes et qui définissent différents niveaux de généralisation. Ces regroupements successifs sont représentés par un dendrogramme qui est une hiérarchie indicée. Pour établir une « bonne » hiérarchie, un critère numérique est nécessaire afin de sélectionner à chaque étape de l'algorithme la « meilleure » agrégation ou le meilleur regroupement de classes. Ce critère est défini à partir de l'indice de dissimilarité entre les variables de l'ensemble à classifier. En effet, un regroupement de classes s'effectue sur le critère d'une similarité forte entre les deux classes à regrouper. C'est là, la principale difficulté de ce type d'algorithme : le choix d'une mesure de dissimilarité entre les classes créées. On parle alors de *stratégie d'agrégation.*

Les stratégies d'agrégation

Nous rappelons quelques stratégies d'agrégation utilisées dans la littérature. Pour cela, nous supposons que nous avons bien défini au préalable un indice de dissimilarité d sur les objets de E. Le problème est maintenant de définir une dissimilarité entre la réunion de deux classes et d'une troisième. On parle alors de critère d'agrégation : comment définir $\delta((i, j), k), (i, j, k) \in E^3$?

Le lien minimum ou lien simple La stratégie du lien minimum (« single linkage ») consiste à définir comme dissimilarité entre la réunion de deux éléments et un troisième la fonction δ, telle que :

$$\delta((i, j), k) = \min(d(i, k), d(j, k))$$

La distance entre deux parties A et B de E est donc la plus petite distance entre deux éléments de A et B : $\delta(A, B) = \min_{a \in A, b \in B}(d(a, b))$ (Figure 3.3).

Le lien maximum ou diamètre La stratégie du diamètre (« complete linkage ») consiste à définir comme dissimilarité entre la réunion de deux éléments et un troisième

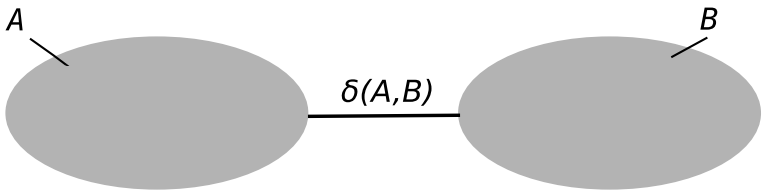


FIG. 3.3 – *Stratégie du lien minimum entre A et B.*

la fonction δ, telle que :

$$\delta((i,j),k) = \max(d(i,k), d(j,k))$$

La distance entre deux parties A et B de E est donc la plus grande distance entre deux éléments de A et B : $\delta(A,B) = \max_{a\in A, b\in B}(d(a,b))$ (Figure 3.4).

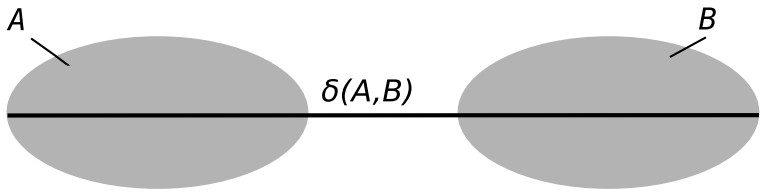


FIG. 3.4 – *Stratégie du diamètre entre A et B.*

Le lien moyen La stratégie du lien moyen (« average linkage ») consiste à définir comme dissimilarité entre la réunion de deux éléments et un troisième la fonction δ, telle que :

$$\delta((i,j),k) = \frac{d(i,k) + d(j,k)}{2}$$

La distance entre deux parties A et B de E est donc la moyenne des distances entre deux éléments de A et B : $\delta(A,B) = moyenne_{a\in A, b\in B}(d(a,b))$ (Figure 3.5).

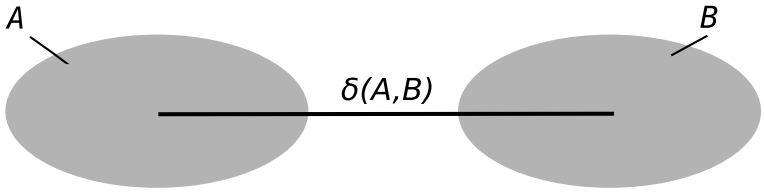


FIG. 3.5 – *Stratégie du lien moyen entre A et B.*

La Figure 3.7 nous permet de réaliser l'importance du choix de la stratégie d'agrégation. En effet, le saut minimum (Figure 3.7(a)) a tendance à « écraser » les niveaux de liaison, tandis que la méthode du diamètre (Figure 3.7(b)) les distend. Avec le saut minimum, il faut accepter que 1'on puisse rapprocher des points extrêmement différents ; c'est ce que l'on appelle « l'effet de chaîne ».

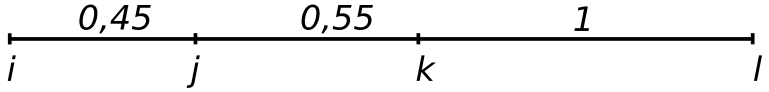
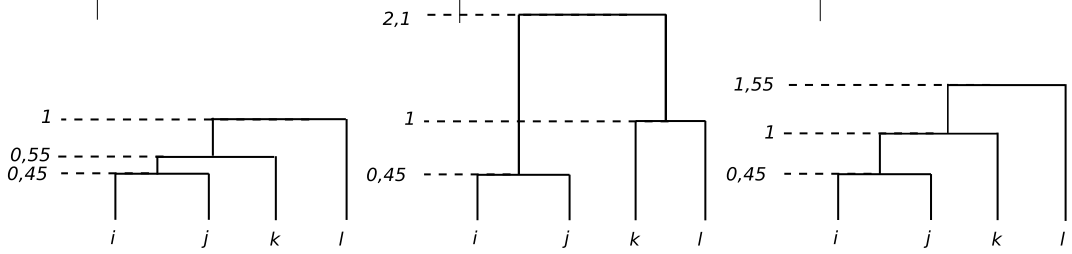


FIG. 3.6 – *Exemple de jeu de données à quatre éléments.*



(a) Agrégation par le lien minimum (b) Agrégation par le lien du diamètre (c) Agrégation par le lien moyen

FIG. 3.7 – *Construction d'un dendrogramme avec plusieurs stratégies d'agrégation sur le jeu de données présenté en Figure 3.6.*

Autres stratégies De nombreuses autres stratégies d'agrégation ont été proposées dans la littérature (e.g. Saporta, 2006). Certaines sont des généralisations de la formule de récurrence proposée par Lance and Williams (1967). Soient $A, B, C \in \mathcal{P}(E)$:

$$\delta(A \cup B, C) = \alpha_1 d(A,C) + \alpha_2 d(B,C) + \alpha_3 d(A,B) + \alpha_4 \mid d(A,C) - d(B,C) \mid, \alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}$$

La stratégie du lien minimum consiste à prendre $\alpha_1 = \alpha_2 = \frac{1}{2}$, $\alpha_3 = 0$ et $\alpha_4 = -\frac{1}{2}$. Pour le lien maximum, $\alpha_1 = \alpha_2 = \frac{1}{2}$, $\alpha_3 = 0$ et $\alpha_4 = \frac{1}{2}$. Ajoutons enfin le critère de Ward qui consiste à choisir le regroupement de classes qui minimise l'inertie intraclasse (définition 3.2.5). Le niveau d'agrégation est alors égal à la perte d'inertie intraclasse résultant du regroupement choisi.

Le phénomène d'inversion Dans tous les cas, nous devons nous assurer que la stratégie d'agrégation ne risque pas de créer un phénomène d'inversion dans la hiérarchie. L'inversion arrive si l'on ne s'assure pas que les niveaux ultérieurs de la hiérarchie seront supérieurs à celui que l'on est en train de créer. Un exemple est présenté dans la Figure 3.8 où la distance entre l'élément k et le groupe (i, j) est inférieure à la distance entre i et j. Une mesure de dissimilarité ou stratégie d'agrégation doit donc vérifier :

$$\delta((i,j),k) \geq d(i,j) \forall (i,j,k) \in E^3$$

C'est le cas des stratégies d'agrégation classiques que nous venons de présenter.

3.2.2 Complexité de l'algorithme

À la première étape de l'algorithme sur un ensemble E à n éléments, il y a $\frac{n(n-1)}{2}$ dissimilarités à calculer et à comparer pour trouver la dissimilarité minimale et réunir

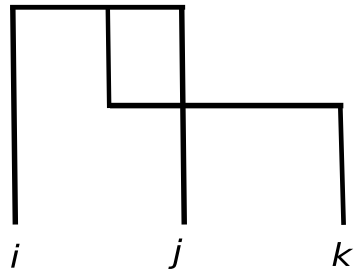


FIG. 3.8 – *Illustration d'un phénomène d'inversion : $d(i,j) \geq \delta((i,j),k)$.*

les deux objets correspondant à cette valeur minimale. Puis, il faut mettre à jour les dissimilarités en considérant la réunion précédente comme un nouvel objet et réitérer la première étape sur $n-1$ objets. La complexité d'un tel algorithme est donc en n^3 (e.g. Saporta, 2006) : une classification est vite coûteuse en terme de temps de calcul. Cependant, des algorithmes accélérés ont été proposés pour une grande variété de mesures (e.g. Brucker and Barthélemy, 2007). Ils offrent une complexité en $O(n^2)$.

Algorithme 10 L'algorithme de base de Classification Ascendante Hiérarchique.

Entrées:

E, un ensemble de n éléments à classer

Sorties:

H, une hiérarchie indicée de E

Algorithme

Initialisation. Chaque singleton de E forme une classe. Le nombre de classes est n. Une matrice $n \times n$ des dissimilarités deux à deux est calculée.

Tant que Tous les éléments de E n'ont pas été agrégés en une seule classe **faire**

1. Les deux classes les plus proches, au sens de la dissimilarité choisie, sont agrégées.
2. La matrice des dissimilarités est mise à jour en remplaçant les deux classes par la nouvelle et en recalculant les dissimilarités entre la nouvelle classe et les autres classes.

fin tant que

Fin. Une hiérarchie indicée de E a été construite par regroupements successifs de classes.

3.2.3 Qualité d'une partition

L'algorithme de Classification Ascendante Hiérarchique construit une hiérarchie de classes. Lorsque celle-ci n'est pas trop grande, le choix d'une « bonne » partition peut se faire graphiquement grâce à la recherche d'un « saut » dans les hauteurs successives du dendrogramme. Mais ce choix reste très subjectif et nécessite l'intervention d'une personne dans le processus de choix. C'est pourquoi des mesures d'évaluation de la qualité d'une partition ont été définies afin de guider le choix de la « meilleure » partition compatible avec la hiérarchie obtenue.

Une partition pour être « bonne » doit satisfaire les deux critères suivants :
- 1. Les éléments proches, au sens de la mesure de dissimilarité choisie, doivent être regroupés dans une même classe : la partition est dite *homogène*.
- 2. Les éléments éloignés, au sens de la mesure de dissimilarité choisie, ne doivent pas appartenir à la même classe : la partition est dite *séparée*.

Ces deux notions d'*homogénéité* et de *séparation* d'une partition sont étroitement liées à la mesure de dissimilarité choisie.

Dans la suite, nous noterons $\mathcal{C} = \{C_1, ...C_k\}$ une partition en k classes compatible avec la hiérarchie H de $E = \{e_1, \ldots, e_n\}$. Chaque classe $C_i = \{c_{i_1}, \ldots, c_{i_{n_i}}\}$ possède n_i éléments. Si de plus les éléments appartiennent à un espace euclidien, nous noterons g_i, $i \in \{1, \ldots, k\}$, le centre de gravité de la classe C_i. On a donc $g_i = \frac{1}{|C_i|} \sum_{a=1}^{n_i} c_{i_a}$. Le centre de gravité de l'ensemble E sera alors noté g.

Le critère d'homogénéité d'une classe

L'*homogénéité* d'une classe C_i peut être mesurée par (e.g. Hansen and Jaumard, 1997) :
- Le *diamètre*, *diam*, ou la plus grande dissimilarité entre les éléments de la classe :

$$diam(C_i) = \max_{(a,b) \in C_i^2} d(a, b)$$

- Le *rayon*, r, ou la plus petite des plus grandes dissimilarités de la classe :

$$r(C_i) = \min_{a \in C_i} \max_{b \in C_i} d(a, b)$$

- L'*étoile*, *st*, ou la plus petite somme des dissimilarités de la classe :

$$st(C_i) = \min_{a \in C_i} \sum_{b \in C_i} d(a, b)$$

- La *clique*, *cl*, ou la somme des dissimilarités entre les éléments de la classe :

$$cl(C_i) = \sum_{(a,b) \in C_i^2} d(a, b)$$

Nous remarquons que les deux dernières mesures ne sont pas normalisées mais leur normalisation est simple (voir Hansen and Jaumard (1997)).

Si, de plus, nos données sont dans un espace euclidien alors une mesure d'homogénéité classique d'une partition est l'inertie qui est la moyenne des distances entre les points de la classe et son centre de gravité.

Définition 3.2.4 (Inertie totale d'une classe) *Soit C_i, $i \in \{1, \ldots, k\}$, une classe de la partition $\mathcal{C}$, l'inertie totale de cette classe est définie par :*

$$\mathcal{I}(C_i) = \sum_{a=1}^{n_i} p_a d(c_a, g_i)^2$$

avec p_a la pondération associée à l'élément a.

Une classe sera d'autant plus homogène que son inertie totale sera faible : plus les points d'une classe sont près du centre de la classe, plus la classe est homogène.

Le critère de séparation d'une classe

La *séparation* d'une classe C_i peut être mesurée par (e.g. Hansen and Jaumard, 1997) :

- L'*écart*, *ec*, ou la plus petite dissimilarité entre un élément d'une classe et le complémentaire de la classe :

$$ec(C_i) = \min_{a \in C_i; b \in C_j; i \neq j} d(a, b)$$

- Le *cut*, *c*, ou la somme des dissimilarités entre les éléments de la classe et ceux d'une autre :

$$c(C_i) = \sum_{a \in C_i} \sum_{b \in C_j; i \neq j} d(a, b)$$

L'homogénéité et la séparation d'une partition

Nous avons présenté des critères d'homogénéité et de séparation pour une classe d'une partition. Il est maintenant intéressant de pouvoir évaluer la partition complète en terme de séparation et d'homogénéité. Nous cherchons donc, pour avoir une bonne partition, à minimiser les critères d'homogénéité et à maximiser ceux de séparation. Pour cela, on peut agréger les coefficients d'homogénéité/séparation calculés pour chaque classe. Le minimum, le maximum, la moyenne peuvent être utilisés selon le sens et l'importance que l'on souhaite donner à l'homogénéité et à la séparation dans le critère de qualité de notre partition (voir Hansen and Jaumard (1997))).

En ce qui concerne les données euclidiennes, la qualité d'une partition est usuellement traitée en terme d'inertie.

Définition 3.2.5 (Inertie intraclasse) *L'inertie intraclasse de la partition $\mathcal{C}$ est définie par :*

$$\mathcal{I}_{intra}(\mathcal{C}) = \sum_{i=1}^{k} (I(C_i))$$

L'inertie intraclasse d'une partition est donc la somme des inerties totales de chaque classe de la partition. Elle mesure donc l'homogénéité d'une partition en considérant que plus une partition est composée de classes homogènes, plus son inertie intraclasse est faible et donc plus elle est homogène. Il est intéressant de noter le comportement de l'inertie intraclasse pour les deux cas extrêmes de partitions. Si chaque classe de la partition est en fait un singleton alors l'inertie intraclasse est nulle :

$$\mathcal{I}_{intra}(\mathcal{C} = \{\{e_1\}, \ldots, \{e_n\}\}) = 0$$

Et à l'inverse si la partition n'est composée que d'une seule classe alors l'inertie intraclasse est égale à l'inertie totale de l'ensemble E :

$$\mathcal{I}_{intra}(\mathcal{C} = \{\{E\}\}) = \mathcal{I}(E)$$

Définition 3.2.6 (Inertie interclasse) *L'inertie interclasse de la partition $\mathcal{C} = \{C_1, ...C_k\}$ est définie par :*

$$\mathcal{I}_{intra}(\mathcal{C}) = \sum_{i=1}^{k} p_i \delta(g, g_i)^2$$

avec p_i une pondération associée à la classe C_i.

L'inertie interclasse suppose qu'une bonne mesure du niveau de séparation des classes est la somme pondérée des distances entre le centre de gravité de chaque classe et le centre de gravité de l'ensemble de départ. Plus l'inertie interclasse est grande plus les classes sont distinctement séparées. Si chaque classe de la partition est composée d'un singleton de E, l'inertie interclasse est nulle :

$$\mathcal{I}_{inter}(\mathcal{C} = \{\{e_1\}, \ldots, \{e_n\}\}) = 0$$

Si la partition n'a qu'une seule classe, l'inertie interclasse est encore nulle :

$$\mathcal{I}_{intra}(\mathcal{C} = \{\{E\}\}) = 0$$

Un critère de qualité pour s'assurer qu'une partition est homogène et séparée est alors de minimiser l'inertie intraclasse et de maximiser l'inertie interclasse. Ces deux critères sont liés par le théorème de Huygens :

Théorème 3.2.1 (Théorème de Huygens) *Pour chaque partition, l'inertie totale de l'ensemble E est la somme de l'inertie intraclasse et de l'inertie interclasse de la partition :*

$$I(E) = \mathcal{I}_{intra}(\mathcal{C}) + \mathcal{I}_{inter}(\mathcal{C})$$

Ainsi, minimiser l'inertie intraclasse, c'est-à-dire favoriser l'homogénéité des classes est équivalent à maximiser l'inertie interclasse, c'est-à-dire favoriser la séparation des classes. Ce lien entre le critère d'homogénéité et de séparation est intéressant à noter mais il ne se généralise pas à tous les liens présentés précédemment.

Mais il est important de ne pas oublier que le but d'une classification est de créer un nombre de classes inférieur au nombre d'éléments de l'ensemble de départ. La classification en n classes de singletons (inertie intraclasse minimale et inertie interclasse maximale) présente donc généralement assez peu d'intérêt. La qualité d'une partition en terme d'homogénéité (ou de manière équivalente de séparation) est donc un compromis entre l'inertie intraclasse (ou de manière équivalente l'inertie interclasse) et le nombre de classes de la partition.

Dans le cadre de la méthode AVL, il existe un critère combinatoire et de statistique non paramétrique pour évaluer le degré de « significativité » d'une partition (Lerman and Peter, 2003). Un maximum local du critère indique une partition qui définit un état d'équilibre dans la synthèse.

3.3 Application à la classification de variables

La Classification Ascendante Hiérarchique a surtout été utilisée pour classer des individus. Pourtant, la classification de variables en groupes homogènes présente un grand intérêt : elle permet de donner une structure à l'ensemble des variables d'un jeu de données. L'algorithme de Classification Ascendante Hiérarchique fonctionne d'ailleurs aussi très bien sur des variables. La principale difficulté est alors de choisir une distance entre les variables, ce qui est plus ou moins compliqué selon leur nature.

Nous nous plaçons ici dans le cadre de variables aléatoires discrètes comme le sont nos données. Pour les variables qualitatives, le principal problème que l'on rencontre dès lors que l'on souhaite mesurer leur similarité est que les mesures ne sont comparables entre elles que pour des nombres égaux de modalités. L'approche AVL introduite par Lerman que nous allons présenter permet de résoudre ce problème.

3.3.1 La méthode AVL (Analyse de la Vraisemblance des Liens) (Lerman, 1981)

La méthode AVL est une approche très générale de l'évaluation numérique des ressemblances mutuelles entre variables descriptives et entre objets. Elle se distingue des autres approches par une caractéristique fondamentale : les indices de similarité et les critères d'association entre variables ou objets se réfèrent à une échelle de probabilité. Ils expriment le caractère d'invraisemblance de la grandeur des indices de similarité formellement « bien » conçus. La méthode AVL a été particulièrement éprouvée dans le cadre de l'algorithme de Classification Ascendante Hiérarchique pour la classification de variables descriptives. C'est donc dans ce cadre que nous la présentons.

Soient C_1, C_2, C_3 et C_4 quatre classes d'une partition, $\mathcal{C}$, telles que deux regroupements sont alors possibles $R_1 \leftarrow C_1 \cup C_2$ ou $R_2 \leftarrow C_3 \cup C_4$ et soient d_{var}, une mesure de dissimilarité entre variables et δ_{var} une distance entre groupes de variables.

Une approche classique, de type « lien simple », va sélectionner la fusion des deux classes les plus proches en définissant la distance,δ_{var}, entre deux classes C_1 et C_2 par

$$\delta_{var}(C_1, C_2) = min\{d_{var}(x, y)|(x, y) \in C_1 \times C_2\}$$

.

L'originalité de la méthode AVL est de choisir la fusion en comparant le degré d'invraisemblance ou d'exceptionnalité de petitesse de $\delta_{var}(C_1, C_2)$ relativement à celle de $\delta_{var}(C_3, C_4)$. Cela nécessite de tenir compte de la densité des points dans (C_1, C_2) d'une part et (C_3, C_4) d'autre part. De manière formelle, la méthode AVL associe, dans le cadre d'un modèle aléatoire d'indépendance respectant les densités des points dans les classes, au couple (C_1, C_2), respectivement (C_3, C_4), un couple (C_1*, C_2*), respectivement (C_3*, C_4*). La fusion de C_1 et C_2 précédera celle de C_3 et C_4 si $\delta_{var}(C_1, C_2)$ est plus invraisemblablement petit que $\delta_{var}(C_3, C_4)$:

$$p\{\delta_{var}(C_1*, C_2*) \leq \delta_{var}(C_1, C_2)\} \leq p\{\delta_{var}(C_3*, C_4*) \leq \delta_{var}(C_3, C_4))\}$$

Les principales étapes de l'algorithme d'Analyse de Vraisemblance du Lien sont donc :

1. Définition d'un indice de similarité entre variables.
2. Définition d'un indice de dissimilarité entre classes correspondant au *lien simple*.
3. Introduction d'un modèle probabiliste d'absence de liaison, tenant compte de la taille des classes.
4. Comparaison des degrés d'invraisemblance des petitesses de $\delta_{var}(C_1, C_2)$ et $\delta_{var}(C_3, C_4)$.
5. Adoption de la règle de fusion de la paire de classes pour laquelle la probabilité est la plus petite.

Dans la méthode AVL générale, l'approche *vraisemblance du lien* intervient dans l'étape de l'agrégation des classes. Mais, il est tout à fait possible de l'utiliser dès la construction de la mesure de similarité entre variables en définissant un indice probabiliste de la vraisemblance du lien entre les variables. Dans le contexte de la classification de variables, Lerman (1981) propose de remplacer la valeur de l'indice de similarité entre variables de même nature (corrélation, χ^2, information mutuelle, ...) par la probabilité de trouver une valeur inférieure sous hypothèse d'indépendance (« absence de lien »). Ainsi, au lieu de prendre s, un indice de similarité, on prendra $p(S < s)$ sous hypothèse d'indépendance. Pour les variables qualitatives, le principal intérêt est qu'elles deviennent dès lors comparables entre elles indépendamment de leur nombre de modalités.

L'approche probabiliste de Nicolau and Bacelar-Nicolau (1998) est basée sur le même raisonnement. Elle porte sur une méthode de classification hiérarchique basée sur un coefficient de similarité, γ_{xy} avec x et y des éléments de E, et un critère d'agrégation, Γ_{AB} avec A et B des parties de E, probabilistes. Les auteurs font les hypothèses suivantes :

- γ_{xy} suit une loi uniforme,
- la similarité entre chaque paire de sous-parties A et B appartient à la famille des γ-similarités, c'est à dire à l'ensemble $\{\gamma_{ab}|(a,b) \in A \times B\}$,
- la famille des γ-similarités est uniformément répartie sur $[0,1]$.

Le coefficient γ est construit à partir d'une mesure de similarité $S : E \times E \to \mathbb{R}$; s_{xy} est la valeur de S pour la paire (x,y). La fonction S est transformée grâce à sa fonction de distribution cumulée et c'est ainsi que γ_{xy} est définie :

$$\gamma_{xy} = prob(S \leq s_{xy})$$

Conclusion

Dans ce chapitre, nous avons présenté l'algorithme de construction d'une Classification Ascendante Hiérarchique. Après une présentation générale de la classification, nous avons exposé les travaux de Lerman sur la classification de variables avec une approche probabiliste.

La classification ascendante hiérarchique que nous utilisons en pré-traitement de notre algorithme de sélection de variables, ClassAdd, utilise l'information mutuelle comme distance entre variables et le lien moyen comme critère d'agrégation entre classes.

4

ClassAdd : un algorithme de sélection de variables

Résumé

Ce chapitre présente l'algorithme de sélection de variables que nous proposons : l'algorithme ClassAdd basé sur une Classification Ascendante Hiérarchique en traitement préalable et utilisant une approximation additive de l'information mutuelle comme mesure de pertinence.

Nous présentons, tout d'abord, quelques prérequis (section 4.1) nécessaires pour construire l'approximation additive de l'information mutuelle que nous proposons. Ainsi, nous définissons la fonction de Möbius et la formule d'inversion de Rota. Nous rappelons aussi les bases des modèles log-linéaires. En effet, l'intuition utilisée pour les bâtir est la même que celle utilisée pour décomposer additivement l'information mutuelle.

Dans la section 4.2, nous présentons notre mesure de pertinence : l'information mutuelle décomposée additivement afin de l'approximer facilement.

L'heuristique de recherche que nous proposons est détaillée dans la section 4.3. Nous proposons de structurer l'espace des variables grâce à une Classification Ascendante Hiérarchique. Nous choisissons l'heuristique de ne garder au plus qu'une seule variable par classe. Dans cette partie, nous présentons les critères de qualité d'une partition que nous avons retenus. Nous concluons en présentant l'algorithme ClassAdd de manière détaillée.

Sommaire

Introduction

Dans le cadre de notre travail, nous nous intéressons au cas où les variables potentiellement discriminantes sont toutes discrètes ou nominales et nous proposons une procédure *filtre*, ClassAdd, utilisant une *troncature k-additive de l'information mutuelle* (Kojadinovic, 2005) comme mesure de pertinence. L'approximation que nous utilisons permet d'approcher la pertinence d'un ensemble de variables à partir des pertinences de ses sous-ensembles de faible cardinal. Afin d'éviter d'avoir à parcourir la totalité des sous-ensembles non vides de l'ensemble des variables potentiellement discriminantes ou d'avoir à recourir à des heuristiques souvent trop sous-optimales du type *sélection pas à pas*, nous proposons d'effectuer, en pré-traitement de la sélection de variables, une classification ascendante hiérarchique de l'ensemble des variables potentiellement discriminantes afin d'en identifier la *structure*.

Dans ce chapitre, nous présentons en pré-requis la fonction de Möbius, la formule d'inversion de Rota et les modèles log-linéaires. Puis nous nous servons de ces outils pour définir une troncature k-additive de l'information mutuelle. Enfin, nous exposons notre classification ascendante hiérarchique de l'ensemble des variables et la définition de critères de qualité d'une partition.

4.1 Pré-requis

4.1.1 Fonctions de Möbius, formule de Rota (1964)

Chaînes dans les ensembles finis ordonnés

Définition 4.1.1 (Chaîne d'un ensemble fini ordonné) *Soit $\mathbb{L}$ un ensemble fini ordonné par une relation notée $\leq$, pour tout entier $v \geq 0$ et pour tout couple (l, l') d'éléments*

de $\mathbb{L}^2$ tels que $l \leq l'$, on appelle chaîne de longueur v joignant l à l' toute suite finie $l_0, \ldots, l_v$ d'éléments de $\mathbb{L}$ tels que :

$$l = l_0 < l_1 < \cdots < l_v = l'$$

On note $c_v(l, l')$ le nombre de ces chaînes.

Nous avons
- $c_0(l, l) = 1$
- $c_0(l, l') = 0$ pour $l < l'$
- $c_v(l, l') = 0$ pour $v > 0$
- $c_1(l, l') = 1$ pour $l < l'$

Proposition 4.1.1 *Deux relations de récurrences existent entre les nombres $c_v(l, l')$:*

$$c_{v+1}(l, l') = \sum_{l \leq l'' < l'} c_v(l, l'')$$

$$c_{v+1}(l, l') = \sum_{l < l'' \leq l'} c_v(l'', l')$$

Preuve. Toute chaîne de longueur $v+1$ joignant l à l' peut se décomposer en une chaîne de longueur v joignant l à un certain $l'' < l'$ et d'une chaîne de longueur 1 entre l'' et l'. La deuxième relation se démontre de la même manière. □

Fonction de Möbius

Définition 4.1.2 (Fonction de Möbius) *La fonction de Möbius $\mu_{\mathbb{L}}$ de l'ensemble ordonné $\mathbb{L}$ est la fonction définie sur $\mathbb{L} \times \mathbb{L}$ à valeurs dans $\mathbb{Z}$ par*

$$\mu_{\mathbb{L}}(l, l') = \sum_{v \geq 0} (-1)^v c_v(l, l')$$

si $l \leq l'$ et par

$$\mu_{\mathbb{L}}(l, l') = 0$$

sinon.

Proposition 4.1.2 *La fonction $\mu_{\mathbb{L}}$ vérifie*

$$\mu_{\mathbb{L}}(l, l) = 1$$

et si $l < l'$

$$\sum_{l \leq l'' \leq l'} \mu_{\mathbb{L}}(l, l'') = 0$$

$$\sum_{l \leq l'' \leq l'} \mu_{\mathbb{L}}(l'', l') = 0$$

Preuve. La première relation se déduit directement de la définition 4.1.1 et du fait que $\mu_{\mathbb{L}}(l,l) = c_0(l,l)$. Les deux autres relations s'obtiennent en récrivant la définition 4.1.2 de $\mu_{\mathbb{L}}$ avec la proposition 4.1.1. □

Nous présentons maintenant l'utilisation de la fonction de Möbius dans deux cas : les nombres entiers, cas usuel d'application, et les parties d'un ensemble, notre cas d'application.

Application sur l'ensemble des nombres entiers $\mathbb{N}$ Soit $n \in \mathbb{N}$ et $\mathbb{L}$ l'ensemble des diviseurs de n ordonné par la relation de divisibilité. La fonction de Möbius dans ce cas est

$$\mu_{\mathbb{L}}(l, l') = \mu(l'/l)$$

où μ est la fonction de Möbius classique donnée par

$$\mu(1) = 1$$

et si $l_1, l_2, \ldots, l_v$ sont des nombres premiers distincts

$$\mu(l_1, l_2, \ldots, l_v) = (-1)^v$$

et dans tous les autres cas

$$\mu(l) = 0$$

Application sur les parties d'un ensemble fini Soit S un ensemble fini et $\mathbb{L} = \mathcal{P}(S)$ l'ensemble des parties de S ordonné par la relation d'inclusion. La fonction de Möbius dans ce cas est

$$\mu_{\mathbb{L}}(A, B) = (-1)^{|B|-|A|}$$

si $A \subset B$ et

$$\mu_{\mathbb{L}}(A, B) = 0$$

sinon.

Formule d'inversion de Rota (1964)

Théorème 4.1.1 (Inversion de Rota) *Soit f une fonction définie sur $\mathbb{L}$ à valeurs dans un groupe abélien G. Posons*

$$g(l) = \sum_{l' \leq l} f(l'),$$

il est possible de retrouver la fonction f connaissant la fonction g grâce à la formule

$$f(l) = \sum_{l' \leq l} \mu_{\mathbb{L}}(l', l) g(l')$$

Preuve.

$$\sum_{l' \leq l} \mu_{\mathbb{L}}(l', l) g(l') = \sum_{l' \leq l} \mu_{\mathbb{L}}(l', l) \sum_{l'' \leq l'} f(l'')$$

et par un regroupement des termes

$$\sum_{l' \leq l} \mu_{\mathbb{L}}(l', l) g(l') = \sum_{l'' \leq l} f(l'') \sum_{l'' \leq l' \leq l} \mu_{\mathbb{L}}(l', l)$$

Or la somme$\sum_{l'' \leq l' \leq l} \mu_{\mathbb{L}}(l', l)$ est non nulle si et seulement si $l'' = l$ et dans ce cas là $\sum_{l'' \leq l' \leq l} \mu_{\mathbb{L}}(l', l) = 1$, d'où

$$\sum_{l' \leq l} \mu_{\mathbb{L}}(l', l) g(l') = f(l)$$

□

4.1.2 Les modèles log-linéaires

Une modélisation statistique poursuit deux buts généralement en contradiction :
- la modélisation du phénomène avec le moins de paramètres possibles,
- la minimisation de l'écart entre les observations réelles et les estimations issues du modèle.

Pour effectuer cette modélisation, nous disposons du tableau de contingence (e.g. le tableau 2.1) issu du jeu de données à étudier. Quand l'effectif estimé d'une cellule du tableau de contingence est vu comme le produit de paramètres, le modèle est dit *multiplicatif*. Ceci est directement lié au fait que l'effectif espéré sous hypothèse d'indépendance entre deux variables qualitatives X, à m modalités, et Y, à l modalités, s'exprime sous la forme d'un produit $n_{ij} = \frac{n_{i.} n_{.j}}{n_{..}}$ où $n_{i.}$ est l'effectif total de la ligne i ($X = x_i$), $n_{.j}$ est l'effectif total de la colonne j ($Y = y_j$), $n_{..}$ est l'effectif total pour $i \in \{1, ..., m\}$ et $j \in \{1, ..., l\}$. Lorsqu'on est dans ce type de modèle, le passage aux logarithmes rend le modèle additif et l'on parle de modèle *log-linéaire*.

Un tableau de contingence croisant deux variables X et Y ayant, respectivement, m et l modalités, pourra être parfaitement décrit par un modèle saturé comportant $m * l$ paramètres :
- 1 paramètre pour ajuster l'effectif total du tableau
- $l - 1$ paramètres pour ajuster la marge sur X (ordre 1)
- $m - 1$ paramètres pour ajuster la marge sur Y (ordre 1)
- $(m - 1)(l - 1)$ paramètres pour ajuster l'effectif d'autant de cellules pour décrire l'association statistique (l'écart à l'indépendance) entre X et Y (ordre 2). Dans le cas de variables indépendantes, tous ces paramètres sont nuls.

Si l'on généralise à p variables, $X_1, \ldots, X_p$ ayant, respectivement, $m_1, \ldots, m_p$ modalités, un tableau de contingence pourra être décrit par un modèle saturé comportant $m_1 * \cdots * m_p$ paramètres indépendants :
- 1 paramètre pour ajuster l'effectif total du tableau
- des paramètres d'ordre 1 pour ajuster les marges sur chaque variable

- des paramètres d'ordre 2 pour ajuster les associations partielles entre deux variables (autres variables contrôlées), ils traduisent l'écart à l'indépendance entre deux variables
- des paramètres d'ordre 3 pour ajuster les interactions entre trois variables (pour décrire comment l'association partielle entre deux variables varient avec les modalités d'une troisième)
- ..
- des paramètres d'ordre p

Prenons l'exemple avec trois variables, X, Y et Z à, respectivement, m, l, h modalités. Le modèle log-linéaire saturé du tableau de contingence de ces trois variables s'écrit alors :

$$log(n_{ijk}) = \lambda + \lambda_i + \lambda_j + \lambda_j + \lambda_{ij} + \lambda_{ik} + \lambda_{jk} + \lambda_{ijk}$$

où n_{ijk} est le nombre d'individus tels que $X = x_i$, $Y = y_j$ et $Z = z_k$ pour $i \in \{1, ..., m\}$,$j \in \{1, ..., l\}$ et $k \in \{1, ..., h\}$ Si l'on a $log(n_{ijk}) = \lambda + \lambda_i + \lambda_j + \lambda_j$, les trois variables sont mutuellement indépendantes.

Si l'on a $log(n_{ijk}) = \lambda + \lambda_i + \lambda_j + \lambda_j + \lambda_{ik}$, Y est indépendante de X et Z.

Si l'on a $log(n_{ijk}) = \lambda + \lambda_i + \lambda_j + \lambda_j + \lambda_{ik} + \lambda_{jk}$, X et Y sont indépendantes si l'on connaît Z.

Estimer tous les paramètres du modèle saturé est rapidement trop coûteux dès que le nombre de variables augmente. Á partir d'un modèle *saturé*, on définit donc un ensemble de modèles *non-saturés* en « éliminant » les paramètres à partir d'un ordre donné. Par exemple, dans le tableau de contingence de trois variables, le modèle d'ordre 2 consiste à ne pas prendre en compte les paramètres d'ordre 3. Cela revient à considérer qu'il n'y a pas d'interactions entre trois variables, c'est à dire, à supposer que l'association statistique entre deux variables quelconques ne diffère pas d'une modalité à une autre de la troisième variable. Le modèle log-linéaire est hiérarchique, c'est-à-dire qu'il inclut tous les termes d'ordre inférieur composés des variables présentes dans les termes d'ordre le plus haut.

4.2 L'information mutuelle comme mesure de pertinence

Nous définissons la pertinence d'un sous-ensemble $\mathbb{X}_r$ de $\aleph$ de taille r par

$$\omega(\mathbb{X}_r) = \begin{cases} 0, & \text{si } \mathbb{X}_r = \emptyset, (r = 0) \\ I_2(\mathbb{X}_r; Y), & \text{sinon.} \end{cases} \tag{4.1}$$

Cette mesure de pertinence est monotone par rapport à l'inclusion : ajouter une variable ne peut qu'augmenter la pertinence de l'ensemble. Sa version estimée à partir des données sera notée $\hat{\omega}$.

Nous avons vu dans la section 2.3 qu'estimer l'information mutuelle est rapidement trop coûteux dès que la taille de $\mathbb{X}_r$ augmente légèrement. Nous présentons donc dans

cette partie une approximation de l'information mutuelle basée sur une troncature k-additive de celle-ci.

4.2.1 La k-additivité

Soit $i : 2^{\aleph} \to \mathbb{R}$ la fonction d'ensemble définie par

$$i(\mathbb{X}_r) = \begin{cases} 0, & \text{si } \mathbb{X}_r = \emptyset, \\ I_{r+1}(X_{a_1}; \ldots; X_{a_r}; Y), & \text{si } \mathbb{X}_r = \{X_{a_1}, \ldots, X_{a_r}\}. \end{cases}$$

En utilisant la fonction de Möbius, définie en 4.1.2, sur les parties d'un ensemble et la formule d'inversion de Rota (théorème 4.1.1), nous pouvons exprimer la mesure de pertinence ω à partir de la fonction d'ensemble i que nous venons de définir.

En effet, la définition 2.2.10 qui généralise l'information mutuelle peut se récrire comme suit

$$\begin{array}{rcl} I_r(X_{a_1}; \ldots; X_{a_r}) & = & \sum_{k=1}^{r} \sum_{\{a_1,\ldots,a_k\} \subseteq \{1,\ldots,r\}} (-1)^{k+1} H(X_{a_1}, \ldots, X_{a_k},) \\ & = & \sum_{\mathbb{T} \subseteq \mathbb{X}_r} (-1)^{|\mathbb{T}|+1} H(\mathbb{T}) \end{array}$$

Ainsi, nous avons

$$\begin{array}{rcl} i(\mathbb{X}_r) & = & I_{r+1}(X_{a_1}; \ldots; X_{a_r}; Y) \\ & = & \sum_{\mathbb{T} \subseteq \{\mathbb{X}_r \cup Y\}} (-1)^{|\mathbb{T}|+1} H(\mathbb{T}) \\ & = & \sum_{\mathbb{T} \subseteq \mathbb{X}_r} (-1)^{|\mathbb{T}|+1} H(\mathbb{T}) + (-1)^{|\mathbb{T}|} H(\mathbb{T}, Y) + (-1)^{|\mathbb{T}|+1} H(Y) \\ & = & \sum_{\mathbb{T} \subseteq \mathbb{X}_r} (-1)^{|\mathbb{T}|+1} \left[H(\mathbb{T}) - H(\mathbb{T}, Y) + H(Y) \right] \\ & = & \sum_{\mathbb{T} \subseteq \mathbb{X}_r} (-1)^{|\mathbb{T}|+1} \omega(\mathbb{T}) \end{array}$$

Soit

$$f(\mathbb{T}) = (-1)^{|\mathbb{T}|+1} \omega(\mathbb{T})$$

alors

$$i(\mathbb{X}_r) = \sum_{\mathbb{T} \subseteq \mathbb{X}_r} f(\mathbb{T})$$

Par la formule d'inversion de Rota (théorème 4.1.1), nous avons donc

$$\begin{array}{rcl} f(\mathbb{X}_r) & = & \sum_{\mathbb{T} \subseteq \mathbb{X}_r} \mu_{\mathbb{L}}(\mathbb{T}, \mathbb{X}_r) i(\mathbb{T}) \\ & = & \sum_{\mathbb{T} \subseteq \mathbb{X}_r} (-1)^{|\mathbb{X}_r| - |\mathbb{T}|} i(\mathbb{T}) \\ (-1)^{|\mathbb{X}_r|+1} \omega(\mathbb{X}_r) & = & \sum_{\mathbb{T} \subseteq \mathbb{X}_r} (-1)^{|\mathbb{X}_r| - |\mathbb{T}|} i(\mathbb{T}) \\ \omega(\mathbb{X}_r) & = & \sum_{\mathbb{T} \subseteq \mathbb{X}_r} \frac{(-1)^{|\mathbb{X}_r| - |\mathbb{T}|}}{(-1)^{|\mathbb{X}_r|+1}} i(\mathbb{T}) \\ \omega(\mathbb{X}_r) & = & \sum_{\mathbb{T} \subseteq \mathbb{X}_r} (-1)^{|\mathbb{X}_r| - |\mathbb{T}| - |\mathbb{X}_r| - 1} i(\mathbb{T}) \\ \omega(\mathbb{X}_r) & = & \sum_{\mathbb{T} \subseteq \mathbb{X}_r} (-1)^{|\mathbb{T}|+1} i(\mathbb{T}) \end{array}$$

Nous avons donc la relation

$$\omega(\mathbb{X}_r) = \sum_{\mathbb{T}\subseteq\mathbb{X}_r} (-1)^{|\mathbb{T}|+1} i(\mathbb{T})$$

et de la même manière, nous obtenons aussi

$$i(\mathbb{X}_r) = \sum_{\mathbb{T}\subseteq\mathbb{X}_r} (-1)^{|\mathbb{T}|+1} \omega(\mathbb{T})$$

Il s'ensuit que la pertinence d'un sous-ensemble $\mathbb{X} = \{X_{i_1}, \ldots, X_{i_r}\}$ de $\aleph$ peut être réécrite comme

$$\begin{aligned}\omega(\mathbb{X}) = &\sum_{X_j\in\mathbb{X}} I_2(X_j;Y) - \sum_{\{X_j,X_k\}\subseteq\mathbb{X}} I_3(X_j;X_k;Y) \\ &+ \sum_{\{X_j,X_k,X_l\}\subseteq\mathbb{X}} I_4(X_j;X_k;X_l;Y) - \cdots + (-1)^{r+1} I_{r+1}(X_{i_1};\ldots;X_{i_r};Y). \quad (4.2)\end{aligned}$$

La pertinence de $\mathbb{X}$ est ainsi calculée d'abord en sommant les pertinences des singletons contenus dans $\mathbb{X}_r$, puis en soustrayant les informations mutuelles entre paires de variables de $\mathbb{X}_r$ et Y, ensuite en ajoutant les informations mutuelles entre variables des sous-ensembles de 3 éléments de $\mathbb{X}_r$ et Y, etc. Les informations mutuelles qui sont rajoutées ou enlevées peuvent être vues comme des *termes correcteurs* ou des termes d'*ordre supérieur* et s'apparentent aux termes d'interaction utilisés dans le contexte de l'*analyse de variance* ou des *modèles log-linéaires* (Agresti, 2002) présentés dans la section 4.1.2. Dans la suite, nous utilisons le même principe de simplification que pour les modèles log-linéaires en considérant qu'à partir d'un k fixé, les interactions entre plus de k variables peuvent être négligées. Cette simplification est utilisée pour faciliter l'estimation de l'information mutuelle et non pour construire un modèle simplifié.

Afin donc d'obtenir une approximation de l'information mutuelle moins coûteuse en terme de temps de calcul, nous proposons de procéder à une *troncature k-additive* de ω pour un $k \in \{1, \ldots, m\}$ fixé, c'est-à-dire de négliger les *termes correcteurs* d'ordre supérieur à k dans l'Eq. (4.2). La troncature k-additive de ω est simplement définie par

$$\omega^{(k)}(\mathbb{X}_r) = \sum_{\mathbb{T}\subseteq\mathbb{X}_r, |\mathbb{T}|\leq k} (-1)^{|\mathbb{T}|+1} i(\mathbb{T}), \qquad \mathbb{X}_r \subseteq \aleph.$$

À partir de l' Eq. (4.2), nous voyons ainsi qu'approcher ω par sa troncature k-additive $\omega^{(k)}$ est équivalent à considérer que l'information mutuelle entre plus de k variables potentiellement discriminantes et Y, est négligeable.

Prendre la troncature 1-additive de ω en tant que mesure de pertinence est équivalent à considérer que la pertinence d'un sous-ensemble est égale à la somme des pertinences des singletons qu'il contient, c'est-à-dire que ω est additive. Dans la plupart des situations

réelles, une telle simplification est trop extrême car, généralement, l'ensemble des variables potentiellement discriminantes contient des variables redondantes. Par exemple, nous considérons X_1 et X_2 les deux variables du sous-ensemble les plus informatives mais qui sont redondantes. Avec une approximation additive, le meilleur sous-ensemble de deux éléments serait l'ensemble $\{X_1, X_2\}$ alors qu'il serait très probablement plus intéressant de combiner l'information apportée par une de ces deux variables avec d'autre information apportée par une autre variable.

La troncature 2-additive apparaît donc plus appropriée car elle prend partiellement en compte les interactions entre variables potentiellement discriminantes sans être trop complexe en terme de nombre de coefficients. En effet, $\omega^{(2)}$ est complètement définie à partir de ses valeurs sur les singletons et les paires de variables potentiellement discriminantes, c'est-à-dire, pour tout $\mathbb{X} \subseteq \aleph$ non vide, il peut être montré (voir p. ex. Kojadinovic, 2005) que

$$\omega^{(2)}(\mathbb{X}) = \sum_{\{X_i, X_j\} \subseteq \mathbb{X}} \omega(\{X_i, X_j\}) - (|\mathbb{X}| - 2) \sum_{X_i \in \mathbb{X}} \omega(\{X_i\}).$$

Utiliser $\omega^{(2)}$ est très avantageux du point de vue du temps de calcul : une fois les pertinences des singletons et des paires de $\aleph$ estimées, la pertinence approchée de n'importe quel sous-ensemble de $\aleph$ peut être immédiatement calculée à l'aide de l'équation précédente. Du point de vue de la qualité de l'approximation, nous pouvons voir, en considérant l'Eq. (4.2), que plus la dépendance entre variables de $\aleph$ est faible, meilleure sera l'approximation de ω par sa troncature 2-additive.

4.3 Une classification hiérarchique des variables comme heuristique de recherche

Le deuxième élément fondamental d'une procédure de sélection de variables est un algorithme de recherche. Afin d'éviter d'avoir à parcourir la totalité des sous-ensembles non vides de $\aleph$ ou d'avoir à recourir à des heuristiques souvent trop sous-optimales du type *sélection pas à pas*, nous proposons d'effectuer une classification ascendante hiérarchique de $\aleph$ afin d'en identifier la *structure*.

Un algorithme de classification ascendante hiérarchique est classiquement défini par deux éléments : une mesure de similarité (ou dissimilarité) et un critère d'agrégation entre classes.

Pour la mesure de similarité, nous avons opté une fois de plus pour l'information mutuelle, cette fois-ci normalisée. La similarité entre deux variables X_i et X_j de $\aleph$ est ainsi définie par

$$I^*(X_i; X_j) = \frac{I_2(X_i; X_j)}{\min[H(p_{X_i}), H(p_{X_j})]}.$$

Il peut être vérifié que la quantité $I^*(X_i; X_j)$ est comprise entre 0 et 1 (Joe, 1989a). De plus, $I^*(X_i; X_j) = 1$ si et seulement si X_i et X_j sont fonctionnellement dépendantes.

Comme critère d'agrégation, nous avons choisi le lien moyen (section 3.2.1), souvent considéré comme une alternative « robuste » au lien simple ou au lien complet.

Une fois la classification construite, les partitions compatibles avec la hiérarchie de classes obtenue sont généralement évaluées en fonction de leur *homogénéité* et de leur *séparation* comme nous l'avons présenté en section 3.2.3. Nous présentons donc maintenant les cinq critères de qualité d'une partition que nous avons retenus de la littérature (e.g. Hardy, 1996). Nous les comparons entre eux sur nos jeux de données dans la section 6.3.

4.3.1 Les critères de qualité d'une partition

La hauteur

Nous utilisons dans ce cas-là la hauteur classique d'une classification ascendante hiérarchique et nous recherchons un saut dans le diagramme des hauteurs pour trouver la partition optimale. Ce critère est très utilisé pour sa facilité de mise en œuvre puisqu'il consiste à identifier un « coude » graphiquement. Cependant, il est difficilement automatisable, ce qui pose souvent problème.

Le critère de Calinski

Dans l'étude de Miligan and Cooper (1985), parmi 30 critères pour déterminer le nombre optimal de classes d'une partition, le critère de Calinski and Harabasz (1974) produit, sur les jeux de tests qu'ils ont utilisés, les meilleurs résultats. L'idée de l'indice de Calinski et Harabasz est de calculer la somme des distances entre la $k - eme$ classe et les $(k-1) - emes$ classes et de la comparer avec la somme des distances à l'intérieur de chacune des k classes. Cet indice est donc un rapport entre la variance interclasse et la variance intraclasse. L'indice de Calinski et Harabasz est calculé comme

$$CH(n,k) = \frac{\frac{B(k)}{(k-1)}}{\frac{W(k)}{(n-k)}}$$

où n et k sont, respectivement, le nombre d'individus et le nombre de classes dans la partition étudiée. $B(k)$ et $W(k)$ sont, respectivement, la somme des distances interclasses et la somme des distances intraclasses. On remarque que si $B(k)$ et $W(k)$ mesurent des différences ou dissimilarités, alors l'indice CH est d'autant plus élevé que la partition est bonne puisqu'une bonne partition se traduit par des distances interclasses élevées et des distances intraclasses faibles. Par conséquent, si l'on utilise des mesures de similarité alors on s'attendra à un indice CH faible pour une bonne partition. Comme la plupart des critères de choix de partitions, fixer un seuil d'acceptation pour le critère est délicat. On s'intéresse donc plutôt à la variation du critère entre la partition en k classes et la partition en $k-1$ classes. Une variation brusque du critère laisse à penser que l'on a atteint « la partition naturelle ».

Le critère de Hubert

Dans l'étude de Miligan and Cooper (1985), le critère de Hubert and Levin (1976), le *C-Index*,produit, sur les jeux de tests qu'ils ont utilisés, de très bons résultats. Il est calculé comme

$$CI = [d_w - min(d_w)]/[max(d_w) - min(d_w)]$$

où d_w est la somme des distances intraclasses. La valeur minimum dans la hiérarchie indique le nombre de classes optimal.

Le diamètre moyen

Pour chaque classe, la plus grande distance séparant deux variables de la classe est calculée. On parle alors de diamètre d'une classe. Ensuite, une moyenne des diamètres sur toutes les classes permet d'obtenir un critère de qualité de la partition étudiée. L'homogénéité des classes de la partition est donc favorisée en contrepartie de la séparation.

La distance moyenne au centre de la classe

Nous appelons le centre d'une classe la variable la plus proche de chacune des autres en moyenne. Pour chaque classe, on recherche la « variable-centre » et on évalue la distance moyenne des autres variables de la classe avec la variable-centre. Plus une classe est hétérogène, plus la distance moyenne au centre est élevée. Ce critère encore prend plus en compte l'homogénéité des classes de la partition que la séparation.

4.3.2 Algorithme général

La première étape de notre algorithme consiste donc à choisir dans la hiérarchie des partitions construite, la « meilleure »partition compatible. Idéalement, l'objectif serait de retenir, parmi les partitions les plus homogènes compatibles avec la hiérarchie obtenue, la moins fine. D'un point de vue pratique, il faut tempérer l'objectif précédent en trouvant un compromis entre une forte homogénéité et un faible nombre de classes. Nous choisissons donc un critère de qualité parmi les cinq cités précédemment. Nos expérimentations permettent de les comparer.

Une fois la « meilleure » partition, au sens du critère de qualité retenu, choisie, nous proposons une heuristique afin de limiter la taille de l'espace de recherche : nous supposons que les variables d'une même classe peuvent être considérées comme « suffisamment dépendantes ». Ainsi, il nous suffit d'estimer la pertinence des sous-ensembles composés d'au plus une variable de chaque classe. Notre heuristique sera d'autant plus efficace que les classes regrouperont des variables très proches. Ainsi, n'en choisir qu'au plus une par classe ne devrait pas empêcher l'algorithme d'évaluer des sous-ensembles de variables quasi-optimaux. Cette approche nous pousse ainsi à privilégier l'homogénéité

de la partition retenue (comme la plupart des critères que nous avons vus). Cette restriction n'est pas très gênante. En effet, la pertinence des sous-ensembles est mesurée par le biais de la troncature 2-additive de l'information mutuelle. Cette troncature pénalise les sous-ensembles contenant des variables liées, un certain degré de dépendance interclasse est donc envisageable en pratique. Cette heuristique réduit donc considérablement la taille de l'espace de recherche : le nombre de sous-ensembles de variables parcourus est ainsi de l'ordre de $O(|C_1| \times \cdots \times |C_k|)$. Cependant, cela ne reste envisageable que pour des problèmes de faible taille. Il nous semble donc intéressant de regarder si cet espace ne peut pas encore être diminué.

Nous proposons donc de comparer trois méthodes de génération des sous-ensembles à évaluer en se basant sur notre heuristique. La première génération est la génération et l'évaluation de tous les sous-ensembles possibles composés au-plus d'une seule variable par classe. La seconde consiste à ne garder que quelques variables représentatives de chaque classe et à générer tous les sous-ensembles possibles à partir de ces quelques variables. Et enfin pour le troisième cas, nous proposons de ne garder que quelques variables par classe choisies au hasard. Ces générations sont détaillées dans la section de test 7.1.

Une fois une partition compatible sélectionnée, il est demandé à l'utilisateur de donner le nombre maximal p de variables discriminantes à retenir. Les grandes lignes de l'algorithme ClassAdd sont présentées dans l'algorithme 11.

Conclusion

Dans ce chapitre, nous avons détaillé l'algorithme de sélection de variables, ClassAdd, que nous proposons. ClassAdd se base sur une mesure de pertinence définie par une troncature 2-additive de l'information mutuelle. Ainsi, la pertinence de n'importe quel sous-ensemble peut être estimée à partir des seules pertinences de ses singletons et de ses paires. Les coûts en temps de calcul se trouvent ainsi limités. De plus, cette troncature permet de tenir compte des interactions entre variables dans le calcul de la pertinence d'un sous-ensemble. Nous limitons ainsi les risques de choisir des variables redondantes.

Pour ce qui est de la recherche du sous-ensemble optimal, nous proposons de structurer l'espace des variables avec une Classification Ascendante Hiérarchique. Considérant que chaque classe est homogène, nous avons choisi l'heuristique de ne garder qu'au plus une seule variable par classe dans le sous-ensemble optimal. Et pour améliorer encore le nombre de sous-ensembles à évaluer, nous décidons de ne garder que deux variables par classe comme variables génératrices des sous-ensembles à évaluer. Pour cela, nous proposons deux alternatives : garder les deux variables les plus porteuses d'information ou bien garder deux variables au hasard.

Algorithme 11 L'algorithme ClassAdd.

Entrées:

$\aleph$ un ensemble de variables potentiellement discriminantes de Y

p : le cardinal maximal des sous-ensembles de variables pertinentes à renvoyer

Sorties:

Un ensemble de sous-ensembles de variables pertinentes

Soit H une hiérarchie des partitions obtenue par classification ascendante hiérarchique des variables de $\aleph$

Soit $\mathcal{P} = \{C_1, C_2, ..., C_k\}$: la meilleure partition de $\aleph$ en k classes compatible avec H selon le critère de qualité retenu

$q \leftarrow \min(p, k)$

pour $i = 1, \ldots, q$ **faire**

  pour chaque sous-ensemble $\mathbb{X}_i \subseteq \aleph$ de cardinal i composé d'au plus une variable de chaque classe de $\mathcal{P}$ généré selon la méthode de génération choisie **faire**

    calculer sa pertinence $\omega^{(2)}(\mathbb{X}_i)$

    stocker le couple $\left(\mathbb{X}_i, \omega^{(2)}(\mathbb{X}_i)\right)$

  fin pour

  Afficher parmi les sous-ensembles de cardinal i considérés dans la boucle précédente, celui qui a la plus forte pertinence

fin pour

5

Les données de l'étude

Résumé

Ce chapitre décrit les données que nous avons utilisées pour évaluer expérimentalement la qualité de notre algorithme.

Dans la section 5.1, nous détaillons les données issues du contexte applicatif que nous avons utilisées. L'entreprise PerformanSe dispose de nombreux jeux de données où les individus sont décrits par des variables comportementales. Nous présentons les logiciels desquels sont issus ces jeux de données : Echo, Dialog, Oriente. Puis, nous décrivons les jeux de données eux-mêmes en provenance d'une population de cadres à la recherche d'un emploi.

La section 5.2 présente les jeux de données utilisés classiquement en apprentissage. Nous en avons retenu 6. Ils sont tous composés de variables discrètes puisque notre cadre applicatif se limite à cette fouille de variables.

Enfin, dans la section 5.3, nous présentons des jeux de données complètement artificiels que nous avons créés de toutes pièces pour vérifier le fonctionnement de notre algorithme.

Sommaire

Introduction

Depuis 1988, PerformanSe (Perfectionnement et Formation au Management par Systèmes Experts) associe des professionnels en gestion des Ressources Humaines et des Universitaires spécialisés d'une part en extraction des connaissances et en gestion des connaissances et d'autre part en psychologie différentielle. C'est dans le cadre de ce partenariat que se sont développés des outils informatisés et des méthodes de travail originales appliquées à la formation, la gestion des compétences et des connaissances. L'activité de PerformanSe s'articule autour de l'évaluation et la gestion des compétences comportementales :

- Évaluer (PerformanSe-DIALECHO)
- Orienter (PerformanSe-ORIENTE)
- Construire et gérer des équipes (PerformanSe-PROJEQUIP)
- Gérer une démarche compétences comportementales (PerformanSe-TALENTS)
- Développer des attitudes managériales basées sur le concept de la Reconnaissance (PerformanSe-AURA)

Le principal outil de PerformanSe est l'outil Dialecho (PerformanSe-Echo et PerformanSe-Dialog). Il est construit autour d'une base de connaissances et d'un système d'inférences permettant à un individu de répondre à des questionnaires et délivrant un bilan personnalisé sur les compétences comportementales de la personne (diriger, argumenter, ...). Ce sont les résultats obtenus par cet outil qui nous servent de jeux de données pour nos expérimentations, le but étant d'essayer de décrire des comportements remarquables pour une population donnée.

Avant d'utiliser notre algorithme pour les besoins de l'entreprise PerformanSe, nous avons testé nos travaux sur des jeux construits de toute pièces afin d'en connaître la structure et sur des jeux classiques en sélection de variables. Tous ces jeux sont détaillés dans cette section.

5.1 Les données de PerformanSe

5.1.1 Le contexte applicatif

De nombreux outils d'évaluation de la personnalité sont utilisés en gestion des ressources humaines (Noci, 2003; Carlyn, 1977; Myers, 1962). Ils ont généralement pour but de construire le profil d'une personne afin d'aider le décideur. Un outil d'évaluation de la personnalité permet d'obtenir le profil d'une personne à partir de réponses à un questionnaire. Les intérêts de ce type d'outils sont multiples : aide au recrutement, aide à la mobilité interne, partie comportementale d'un bilan de compétences ... Il est entendu que ces outils ne doivent pas être utilisés à des fins discriminantes. En particulier, lors d'un recrutement, un tel questionnaire doit être une base d'entretien pour le responsable des ressources humaines et aucunement un outil de présélection des candidats.

Les outils existants se divisent en deux grands groupes de questionnaires : les questions ouvertes et fermées. Pour les questionnaires ouverts, la personne est libre de ses réponses. Celles-ci sont ensuite analysées par un expert en psychologie qui va dresser un profil comportemental pour cette personne. Ces outils sont parfois remis en cause car l'interprétation de l'expert est très subjective et variable. C'est pourquoi ces questionnaires ne sont pas très nombreux. Nous citons, par exemple, Phrases (Stein and Bransford, 1979) qui demande à l'utilisateur de compléter 50 phrases en 30 minutes, sous les yeux d'un examinateur. Ensuite, les réponses et le comportement de la personne lors de l'évaluation sont étudiées. Ce type d'outils a été très peu étudié car il est très difficile d'effectuer des analyses statistiques sur des questions ouvertes.

La deuxième catégorie d'outils est beaucoup plus répandue : il s'agit des questionnaires fermés. Ces questionnaires peuvent donc être informatisés facilement. Généralement, ils consistent en un ensemble de questions (les *items*) avec deux ou plus de choix possibles. Un ensemble de règles, du type de celles trouvées dans un système expert, ont été établies préalablement et permettent de construire un profil en fonction des réponses choisies. Un profil est généralement composé d'un nombre prédéterminé de traits de per-

sonnalités (les *dimensions*). De nombreux outils basés sur ce fonctionnement existent. Nous présentons les plus utilisés :

- Sosie (from ECPA) : 20 traits de personnalités évalués à partir de 98 groupes de 4 assertions,
- PAPI (PA Preference Inventory from Cubiks) :
  - le test classique : un choix pour 90 couples de phrases,
  - le test normatif : 126 assertions sur lesquelles l'utilisateur choisit entre « complètement en accord » ou « complètement en désaccord »,
- MBTI (from Myers and Brigg) : 126 questions avec un choix entre deux réponses et un profil choisi parmi 16 profils prédéfinis,
- PerformanSe Echo : 70 questions avec deux réponses au choix et un profil selon 10 dimensions bipolaires déterminés grâce à un ensemble de règles,
- Assess First : 90 questions avec deux réponses au choix et un profil selon 20 dimensions comportementales et 5 familles.

Parmi tous ces outils, très peu ont eu une réelle validation statistique. Cependant, ils sont pour la plupart basés sur des théories psychologiques reconnues : la théorie de Jungian (Cowan, 1989; Myers, 1995), la théorie du « Big Five » (Goldberg, 1981; Wiggins, 1996) ou encore l'étude des motivations (George and Jones, 2002).

5.1.2 L'outil PerformanSe Echo

L'outil d'évaluation de la personnalité, Echo, a été développé par la société PerformanSe. C'est un questionnaire composé de 70 items. Chaque item a deux réponses possibles mais ce n'est pas forcément « oui » ou « non ». C'est un questionnaire *ipsatif*, c'est à dire à choix forcé. Par exemple, la figure 5.1 montre une question issue d'Echo et ses deux réponses possibles.

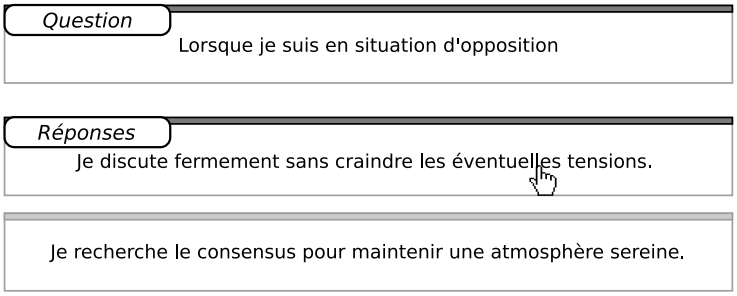


FIG. 5.1 – *Exemple de question ipsative du logiciel Echo.*

Une fois que toutes les questions ont été renseignées, l'outil propose le profil comportemental de l'évalué. Ce profil est construit à partir de 10 dimensions bipolaires détaillés dans le tableau 5.1. Chaque pôle d'une dimension est appelé un *trait* et a une valeur comprise entre 0 et 35. Chaque réponse à une question déclenche un ajout ou un retrait de points sur un ou plusieurs traits. Ensuite, à partir du score obtenu sur chaque couple

de traits, un score de dimension bipolaire est calculé, compris entre 0 et 100. Afin de faciliter l'interprétation, ces scores sont discrétisés selon 3 zones : les petites valeurs (en dessous de 40) marquées -, les valeurs moyennes (entre 40 et 60) marquées 0 et enfin les valeurs élevées (au dessus de 60) marquées +. Par exemple, pour l'EXTraversion, les individus sont classés dans les 3 zones EXT-, EXT0 et EXT+. La figure 5.2 montre un exemple de profil comportemental bipolaire.

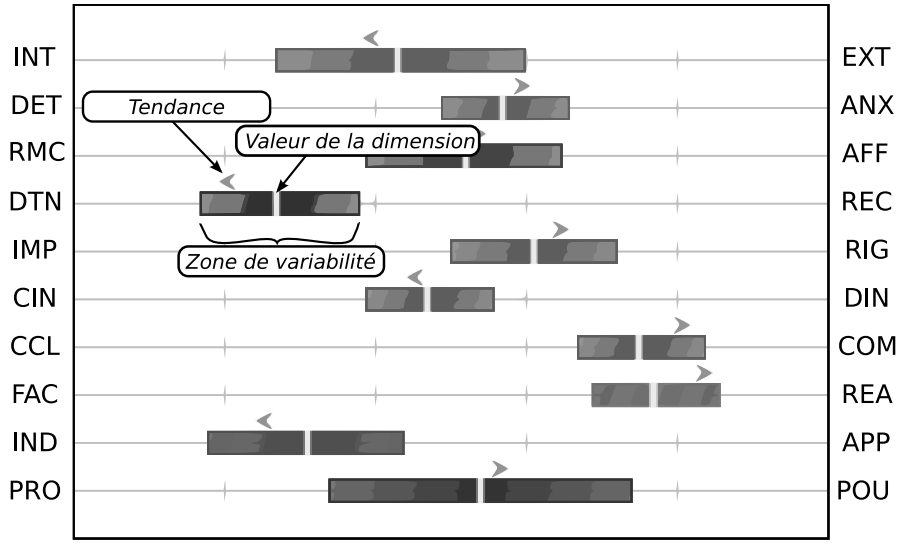


FIG. 5.2 – *Profil bipolaire Echo sur 10 dimensions. Par exemple, la dimension ConCiLiation / COMbativité est très marquée : la COMbativité est élevée, avec une tendance à la COMbativité et la zone de variabilité est faible. Cette dernière notion traduit la latitude de l'individu à évoluer sur une dimension selon les situations. Le trait motivation de PROtection / motivation de POUvoir est assez neutre : valeur moyenne, tendance au POUvoir mais zone de variabilité très grande.*

L'outil Echo est composé de 27000 règles qui permettent de construire en plus du profil un rapport textuel à partir d'une base de textes de 2500 pages. Ce rapport considère les combinaisons entre les dimensions pour essayer de dresser les comportements de l'évalué. La figure 5.3 montre le fonctionnement du système. Ce modèle de personnalité se fonde sur une théorie systémique développée dans les années 50 par l'école de Palo Alto (PerformanSe, 2003). Ce groupe de chercheurs a contribué à définir des formes de communication par des processus relationnels et interactionnels. Les rapports qui s'instaurent entre les individus priment sur les individus eux-mêmes. Il en découle que tout comportement est communication. L'ensemble des relations est considéré comme un système de communications qui interagissent. Il s'en dégage une logique de communication qui forme un ensemble de règles. Une communication perturbée entre un individu et les autres traduit un trouble de comportements. Le modèle PerformanSe a adopté cette démarche en considérant l'individu comme un système en interaction avec un contexte (environnement, entourage, normes de travail ...). C'est une extension du modèle du

TAB. 5.1 – *Les 10 dimensions comportementales.*

Introversion (INT)	**Extraversion (EXT)**
Exprime : réserve, modestie, discrétion, risque de paraître froid, difficulté à communiquer capacité à se concentrer	*Exprime :* expansion, désir d'être remarqué, facilité d'expression, risque de mobiliser l'attention, tendance à l'intrusion
Détente (DET)	**Anxiété (ANX)**
Exprime : état de relaxation, sang-froid	*Exprime :* stress, inquiétude, pouvoir émotionnel, tension, concentration
Remise en Cause (RMC)	**Affirmation (AFF)**
Exprime : volonté de progression, sang-froid	*Exprime :* confiance en soi, convictions intimes, avis tranché
Détermination (DET)	**Réceptivité (REC)**
Exprime : distance aux autres, émotion, résistance passive	*Exprime :* ouverture sur les autres, goût pour l'écoute, empathie
Improvisation (IMP)	**Rigueur (RIG)**
Exprime : goût pour l'inattendu et l'adaptation, réactions spontanées aux évènements, impulsivité	*Exprime :* environnement de travail structuré, sens de la méthode et planification, sens de la hiérarchie
Conformisme Intellectuel (CIN)	**Dynamisme Intellectuel (DIN)**
Exprime : référence à des solutions bien connues, approche analytique, précision, difficulté à avoir une vue globale de situation, savoir expert	*Exprime :* créativité, relations sociales, curiosité intellectuelle (nouvelles idées), compréhension globale des situations, risque de négliger des détails
Conciliation (CCL)	**Combativité (COM)**
Exprime : patience, recherche de relations sereines, esprit de consensus, capacité à agir en arbitre	*Exprime :* comportement réactif, recherche d'intérêts, esprit compétitif, attitude offensive, impatience
Motivation de facilitation (FAC)	**Motivation de réalisation (REA)**
Exprime : plaisir immédiat	*Exprime :* persévérance et succès
Principale crainte : avoir trop de travail	*Principale crainte :* être obligé d'abandonner
Stimulus : facilité, missions courtes	*Stimulus :* projets difficiles
Satisfaction : réussir des succès faciles	*Satisfaction :* faire des efforts
L'argent sanctionne : la saisie des opportunités	*L'argent sanctionne :* le mérite
Relation au temps : projets à court terme	*Relation au temps :* projets à long terme, sentiment de culpabilité lié à la perte de temps
Voc. : garder du temps, utiliser des raccourcis, privilégier le présent...	*Voc. :* construire, persévérer, ténacité...
Motivation d'indépendance (IND)	**Motivation d'appartenance (APP)**
Exprime : indépendance	*Exprime :* influence
Principale crainte : être avalé par le groupe	*Principale crainte :* être exclus
Stimulus : liberté personnelle	*Stimulus :* la communauté
Satisfaction : avoir chacun son territoire	*Satisfaction :* être en bons termes avec les gens
L'argent sanctionne : résultats individuels	*L'argent sanctionne :* résultats communs
Relation au temps : se garder du temps pour soi	*Relation au temps :* donner son temps au groupe
Voc. : mesurer les conséquences de ses actes...	*Voc. :* consensus, solidarité...
Motivation de protection (PRO)	**Motivation de pouvoir (POU)**
Exprime : le besoin de sécurité	*Exprime :* la prise de risque
Principale crainte : ne pas avoir de garanties	*Principale crainte :* ne pas avoir d'influence
Stimulus : maintien des acquis	*Stimulus :* challenge, décider et mener
Satisfaction : esprit en paix	*Satisfaction :* initiateur d'évènements
L'argent sanctionne : un droit acquis	*L'argent sanctionne :* le risque et la prise de responsabilité
Relation au temps : être prévoyant	*Relation au temps :* envie de laisser une marque
Voc. : rester en terrain connu, éviter les surprises, s'organiser	*Voc. :* être en position dominante, être ambitieux...

« Big Five » qui décrit une personnalité selon 5 dimensions que sont l'affirmation, la bienveillance, l'émotivité, l'ouverture et le sens des responsabilités (PerformanSe, 2003). Ce modèle est le résultat de plus de 40 années de travaux menés par une douzaine de chercheurs en psychologie tels que Catell, Fiske ou plus récemment Goldberg. Le modèle PerformanSe est un enrichissement de ce modèle avec en plus :

- l'étude des motivations qui permettent à un individu d'agir,
- l'approche systémique et comportementale qui considère un individu et son environnement comme un tout.

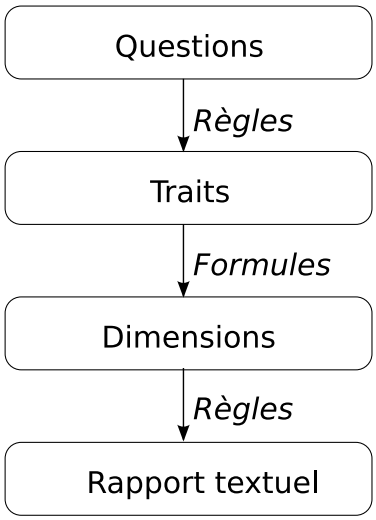


FIG. 5.3 – *Système Echo à base de règles.*

5.1.3 Les activités Oriente

L'expertise Oriente est une expertise complémentaire de l'expertise Echo. Elle se base sur les scores obtenus par un individu sur les 10 dimensions. À partir de ces scores, un diagnostic d'adaptation pour 8 grandes activités est généré :

- Administrer
- Créer
- Echanger
- Produire
- Concevoir
- Gérer
- Argumenter
- Encadrer

Pour chaque activité, un individu est classé dans une des trois catégories suivantes :

- I : atouts notables pour cette activité,
- N : facilités pour cette activité,
- A : conditions particulières pour cette activité.

Centrée sur une approche relationnelle, l'expertise Oriente permet de structurer les démarches d'accompagnement en stimulant les effets d'appropriation des éléments du bilan et de projection vers un environnement professionnel. Cet outil est conçu comme une aide à la réalisation des bilans de compétences et d'élaboration du projet professionnel. Dans les stratégies de mobilité interne, cet outil permet d'ouvrir de nouveaux horizons professionnels et constitue un support de réflexion.

5.1.4 Les données de l'APEC

Nous disposons de 2 jeux de données provenant de l'Association Pour l'Emploi des Cadres, c'est à dire une population de cadres en recherche d'emploi. Le premier jeu contient 1622 passations de l'outil Echo sur 155 variables : nous nommons ce jeu « Activités Oriente ». Une passation Echo est composée pour chaque individu de ses scores d'équilibre, d'amplitude et de tendance sur chacun des 10 traits du modèle, du classement des activités Oriente et d'informations personnelles telles que sexe, langue dans laquelle le questionnaire est passé, âge, diplôme (ce dernier est facultatif). De plus, ce jeu contient aussi les scores des dimensions bipolarisées. Une activité Oriente est discrétisée selon les trois catégories vues précédemment : I, N, A. Nous utilisons ce jeu de test pour sélectionner des sous-ensembles de variables potentiellement explicatives de chacune des activités. Le deuxième jeu contient 1557 passations Echo sur 64 variables : ce jeu s'appelle « Ages ». Cette fois, nous ne disposons pas des scores sur chaque dimension mais seulement de l'appartenance à une dimension en + ou en - ou en 0. Nous avons aussi les activités Oriente. Les individus sont divisés en trois classes d'âge : moins de 30 ans, plus de 45 ans, plus de 50 ans. Ce découpage surprenant s'explique par le fait que ces données ont été extraites afin de comparer le rapport à la recherche d'emploi d'une population de cadres débutants avec une population de cadres séniors. Les cadres en milieu de carrière ont donc été exclus de l'étude. Nous utilisons ce jeu pour trouver des sous-ensembles de variables pertinentes pour contribuer à la compréhension de l'âge des cadres en recherche d'emploi.

5.2 Des jeux de données classiques

Nous avons utilisé des jeux de données test, classiquement utilisés dans la littérature, issus de *University of California at Irvine repository* D.J. Newman and Merz (1998).

5.2.1 Données Soybean

Le jeu de données Soybean est composé de 35 variables discrètes potentiellement discriminantes et de 307 individus. Les individus sont répartis en 19 classes, les quatre dernières étant très peu représentées.

5.2.2 Données Connect

Le jeu de données Connect est composé de 42 variables discrètes correspondant aux positions possibles du jeu "Connect-4". Les 67557 séquences de jeu sont réparties en trois classes traduisant l'état du jeu : victoire, défaite, match nul.

5.2.3 Données Optdigits

Le jeu de données Optdigits est composé de 64 variables discrètes représentant les pixels d'une lettres manuscrites dans une matrice 8*8. Les 3823 matrices se divisent équitablement en neuf classes.

5.2.4 Données Splice

Le jeu de données Splice est un jeu de la même famille que Promoters. Il est composé de 3190 séquences de 60 nucléotides classées en trois catégories.

5.2.5 Données Lung Cancer

Le jeu de données Lung Cancer est décrit 3 pathologies de cancer du poumon qui sont les trois classes possibles des 32 individus décrits par 56 variables discrètes potentiellement discriminantes.

5.2.6 Audiology

Le jeu de données Audiology contient 200 individus qui appartiennent à 24 classes de problèmes audiologiques. Ils sont décrits par 67 variables.

5.3 Des données artificielles

Afin de calibrer nos algorithmes et de s'assurer de leur pertinence, nous avons construit des jeux de données avec une structure prédéfinie. Ainsi, les résultats sont prévisibles et nous pouvons donc évaluer facilement la qualité des sous-ensembles retournés.

5.3.1 Jeu à 15 variables

Soit un jeu de données D, nous considérons un ensemble de 15 variables discrètes potentiellement discriminantes d'une 16 ème.

- $X_1, \ldots, X_4$ et X_{14}, X_{15} sont mutuellement indépendantes, à valeurs dans $\{1, 2, 3, 4\}$, et distribuées selon une loi uniforme.
- $X_5, \ldots, X_8$ sont définies par $X_i := 4 - X_{i-4}$
- $X_9, \ldots, X_{12}$ sont définies par $X_i := X_{i-8}^2$
- X_{13} est définie par $X_{13} := \min(X_1, X_2)$

La variable aléatoire Y à expliquer est définie par $Y := \max(X1, X2) + \min(X_3, X_4)$.

Nous remarquons que dans ce jeu de données, les interactions entre variables sont au plus de taille 2. Ceci a l'avantage de simplifier le modèle et donc de bien comprendre le fonctionnement de l' algorithme. Cependant, ceci devient une limite si l'on s'intéresse à la qualité de l'approximation 2-additive de l'information mutuelle. En effet, cette dernière prend en compte les interactions entre tous les couples de variables donc toutes les interactions dans notre cas. C'est pourquoi nous définissons aussi un jeu de données de 15 variables avec des relations plus complexes mettant en jeu plus de deux variables.

5.3.2 Jeu à 15 variables avec des relations entre trois variables

Soit un jeu de données D, nous considérons un ensemble de 15 variables discrètes potentiellement discriminantes d'une 16 ème.

- $X_1, \ldots, X_4$ et X_{14}, X_{15} sont mutuellement indépendantes, à valeurs dans $\{1, 2, 3, 4\}$, et distribuées selon une loi uniforme.
- $X_5, \ldots, X_8$ sont définies par $X_i := 4 - X_{i-4}$
- X_9 est définie par $X_9 := X_1 * X_2 * X_3$
- X_{10} est définie par $X_{10} := X_2 * X_3$
- X_{11} est définie par $X_{11} := X_3 * X_4$
- X_{12} est définie par $X_{10} := X_4 * X_1$
- X_{13} est définie par $X_{13} := \min(X_1, X_2) + X_4$

La variable aléatoire Y à expliquer est définie par $Y := \max(X1, X2) + \min(X_3, X_4)$.

5.3.3 Jeu à 22 variables

Nous considérons un ensemble de 22 variables discrètes potentiellement discriminantes d'une 23 ème.

- $X_1, \ldots, X_5$ et X_{21}, X_{22} sont mutuellement indépendantes, à valeurs dans $\{1, 2, 3, 4\}$, et distribuées selon une loi uniforme.
- $X_6, \ldots, X_{10}$ sont définies par $X_i := 4 - X_{i-5}$
- $X_{11}, \ldots, X_{15}$ sont définies par $X_i := X_{i-10}^2$
- $X_{16}, \ldots, X_{20}$ sont définies par $X_i := \min(X_1, X_2)$

La variable aléatoire Y à expliquer est définie par $Y := \max(X1, X2, X3) + \min(X_4, X_5)$.

5.3.4 Jeu à 35 variables

Nous considérons un ensemble de 35 variables discrètes potentiellement discriminantes d'une 36 ème.

- $X_1, \ldots, X_5$ et $X_{21}, X_{22}, X_{28}, X_{29}$ sont mutuellement indépendantes, à valeurs dans $\{1, 2, 3, 4\}$, et distribuées selon une loi uniforme.
- $X_6, \ldots, X_{10}$ sont définies par $X_i := 4 - X_{i-5}$
- $X_{11}, \ldots, X_{15}$ sont définies par $X_i := X_{i-10}^2$
- $X_{16}, \ldots, X_{20}$ sont définies par $X_i := \min(X_1, X_2)$
- $X_{23} := 3X_1 + 1$ et $X_{24} := 2X_2 - 1$
- $X_{25} := X_1^3$
- $X_{26} := X_6 + X_{25}$ et $X_{27} := X_7 + X_{26}$
- $X_{30} := X_1 - 1$ si $X_1 < 3$ et $X_{30} := X_1 + 1$ sinon
- $X_{31} := X_1$, $X_{32} := 2 - X_{31}$ et $X_{33} := X_6 + X_7$
- $X_{34} := X_4 - X_5 + X_3$
- $X_{35} := X_2 + X_3$ si $X_2 < 3$ et $X_{35} := X_1$ si $X_2 < 3$ et $X_3 < 3$ et $X_{35} := X_4$ sinon

La variable aléatoire Y à expliquer est définie par $Y := \max(X1, X2, X3) + \min(X_4, X_5)$. Nous voyons ainsi que les variables $X_6, \ldots, X_{20}$, $X_{23}, \ldots, X_{27}$ et $X_{30}, \ldots, X_{35}$ sont redondantes par rapport aux variables $X_1, \ldots, X_5$. Les variables $X_{21}, X_{22}, X_{28}, X_{29}$ sont quant à elles non pertinentes.

5.3.5 Bruitage des données

Pour chacun de ces jeux de données créés de toutes pièces, nous avons construit des jeux de données bruités. Nous avons considéré qu'un jeu était bruité si les relations entre variables n'étaient plus exactes.

Prenons, par exemple, le jeu de 15 variables. Les variables X_1, X_2, X_3 et X_4 restent distribuées selon une loi uniforme et prennent leurs valeurs dans $\{1, 2, 3, 4\}$. Le bruit est introduit sur les variables X_5 à X_{13} et Y. La variable X_5 est maintenant définie par la relation $X_5 := 4 - X_1$ et aléatoirement nous rajoutons 1 à la valeur obtenue. Les autres variables sont définies de la même façon. Ici, une valeur aléatoire comprise entre 0 et 1 est tirée d'une distribution uniforme. Si cette valeur est inférieure à un seuil alors la valeur de la variable pour un individu est modifiée de 1. Plus le seuil est bas, moins le cas se présente et moins les données sont bruitées et inversement avec un seuil haut. Cela nous a permis de définir deux niveaux de bruitage : données légèrement bruitées (le seuil est égal à $0,2$) et données très bruitées (le seuil est égal à $0,5$). Les données non bruitées correspondent bien à un seuil égal à 0.

5.3.6 Taille des données

Pour chacun des ensembles de variables décrits précédemment, nous avons généré des jeux de données à 200, 800 et 1600 individus. Cela nous permet d'étudier la résistance

à la taille des données des algorithmes proposés.

Conclusion

Dans cette section, nous avons présenté les divers jeux que nous utilisons dans nos expérimentations. Tout d'abord, nous avons construit trois jeux de données à 15, 22 et 35 variables qui nous servent de base à une première validation de nos travaux. En effet, nous connaissons parfaitement leur structure, nous attendons donc un certain résultat pour une expérimentation donnée. Puis, dans un deuxième temps, nous regardons le comportement de nos algorithmes sur des jeux de données classiques en apprentissage : Soybean, Connect, Optdigit, Splice, Lung Cancer, Audiology. Enfin, nous pouvons évaluer la qualité de nos travaux dans un cadre applicatif concret que sont les données comportementales de l'entreprise PerformanSe. Cette application est intéressante car les experts en psychologie de l'entreprise connaissent parfaitement le modèle qu'ils ont construit. Ainsi, ils peuvent à la fois valider des résultats issus de la construction du modèle lui-même et l'apparition de tendances issues du seul jeu de données.

6

Analyse de la robustesse

Résumé

Ce chapitre présente les tests de robustesse que nous avons effectués : étude de la mesure de pertinence et des indices de qualité d'une partition.

Dans la section 6.1, nous étudions la qualité de l'approximation 2-additive de l'information mutuelle. Pour cela, nous utilisons nos jeux de données créés à la main et nous définissons des sous-ensembles à évaluer. Nous pouvons prévoir leur pertinence connaissant parfaitement le jeu de données. Nous vérifions donc que l'approximation que nous proposons se comporte comme nous le souhaitons.

Nous complétons cette étude dans la section 6.2. Nous regardons si cette mesure de pertinence n'est pas trop sensible à la taille des données ou au bruitage sur celles-ci. En effet, la robustesse d'une mesure est indispensable pour une procédure de sélection de variables réutilisable.

La section 6.3 porte sur les indices de qualité d'une partition que nous avons présentés dans le chapitre 4. Pour chacun d'eux, nous regardons son évolution en fonction du nombre de classes retenu pour la partition. Ces expérimentations ont encore lieu sur un jeu bien connu dont on peut attendre le nombre de classes.

Enfin, la section 6.4, suit la même démarche que celle décrite dans la section 6.2 mais sur les indices de qualité, afin de nous assurer de leur robustesse.

Sommaire

Introduction

Avant d'évaluer le fonctionnement de notre algorithme sur les jeux de données décrits dans le chapitre 5, nous analysons dans cette section la qualité des mesures que nous utilisons.

Un point crucial de notre algorithme est l'approximation 2-additive de l'information mutuelle afin d'optimiser les temps de calcul. Nous nous assurons donc, dans un premier temps, que cette approximation n'altère pas trop la qualité de l'estimation. En effet, l'algorithme doit gagner en rapidité sans pour autant trop perdre en qualité.

De plus, pour être réutilisable, un algorithme ne doit pas être seulement rapide ou performant, il doit aussi être robuste. Ainsi, s'il est très efficace sur un jeu de données il doit aussi être efficace sur ce même jeu mais plus ou moins bruité. C'est la première notion de robustesse que nous demandons. Ensuite, il doit aussi être robuste à la taille des données. En effet, si nous imaginons un jeu construit sur un modèle donné. Un bon algorithme doit être aussi performant que l'on génère 200 ou 1600 individus de ce jeu. Nous présentons donc dans cette section les résultats que nous avons obtenus sur la robustesse de la mesure de pertinence qu'est l'information mutuelle et sur la robustesse des différents indices de mesure de la qualité d'une partition présentés en section 4.3.1.

6.1 Qualité de l'approximation 2-additive

Toute la difficulté du choix de la mesure de pertinence réside dans le fait de trouver le bon compromis entre un temps de calcul raisonnable et une pertinence assez juste. À un bout de l'échelle, nous avons l'approximation 1-additive très rapide mais qui ne permet pas de reconnaître les variables redondantes par exemple. La pertinence calculée d'un ensemble est donc faussée. À l'opposé, on a l'information mutuelle par sa définition, très coûteuse à calculer dès lors que l'on a plusieurs variables mais qui permet de connaître la juste pertinence d'un ensemble. Nous avons fait le choix d'implémenter une approximation 2-additive de l'information mutuelle. La pertinence, ω, d'un sous-ensemble S pour expliquer une variable Y est donc définie par :

$$\omega^{(2)}(S) = \sum_{\{X_i,X_j\}\subseteq S} \omega(\{X_i, X_j\}) - (|S| - 2) \sum_{X_i \in S} \omega(\{X_i\})$$

Cette approximation nous semble suffisante pour estimer la pertinence d'un ensemble avec qualité. C'est ce que nous étudions avec ce premier test.

Utiliser une approximation 2-additive de l'information mutuelle pour estimer la pertinence d'un sous-ensemble est d'ores et déjà un gain de temps de calcul non négligeable par rapport à l'utilisation de l'information mutuelle. Cependant, cet intérêt n'est pas suffisant pour justifier ce choix d'approximation. En effet, se limiter au temps de calcul nous inciterait plutôt à choisir une approximation 1-additive encore plus efficace en terme de coût. Nous allons donc dans une première série de tests regarder si choisir l'approximation 2-additive par rapport à l'approximation 1-additive est un choix pertinent. En d'autres termes, nous allons vérifier que la prise en compte des interactions entre couples via la 2-additivité apporte vraiment un gain dans la qualité de l'estimation de la pertinence des sous-ensembles.

Pour cela, nous considérons les jeux de données à 15 variables (Sections 5.3.1 et 5.3.2). Connaissant, parfaitement leurs structures, nous définissons des ensembles que nous jugeons intéressants car ils contiennent des variables redondantes, des variables non pertinentes. Ces ensembles sont présentés dans le Tableau 6.1. Par exemple, l'ensemble 9 contient une variable non-pertinente, nous nous attendons donc à ce que sa pertinence estimée soit plus faible que l'ensemble 1 qui est l'ensemble le plus pertinent pour expliquer Y pour ces deux jeux de données. Ce test est effectué avec une estimation classique de l'information mutuelle sur une génération de 800 réalisations du vecteur aléatoire $(X_1, \ldots, X_{15}, Y)$.

Ce premier test s'est déroulé en trois étapes : l'étude sur des jeux non bruités, puis sur des jeux légèrement bruités et enfin sur des jeux très bruités (cf. Section 5.3.5 pour la définition du bruit). Les résultats sont présentés sous la forme de graphiques mettant en parallèle pour chaque sous-ensemble du Tableau 6.1 trois calculs de sa pertinence : l'information mutuelle par sa définition classique (cela é été possible vue la petite taille de nos sous-ensembles), une approximation 1-additive de l'information mutuelle et enfin

Indice	Sous-ensemble	Description
1	X_1, X_2, X_3, X_4	ensemble optimal
2	X_2, X_3, X_4, X_5	ensemble "presque" optimal
3	X_2, X_3, X_4, X_9	ensemble "presque" optimal
4	X_2, X_3, X_4, X_{13}	ensemble "presque" optimal
5	X_1, X_2, X_3, X_5	une variable redondante
6	X_1, X_2, X_3, X_9	une variable redondante
7	X_1, X_2, X_3, X_{13}	une variable redondante
8	X_1, X_2, X_5, X_9	deux variables redondantes
9	X_1, X_2, X_3, X_{14}	une variable non-pertinente
10	X_2, X_3, X_{14}, X_{15}	deux variables non-pertinentes

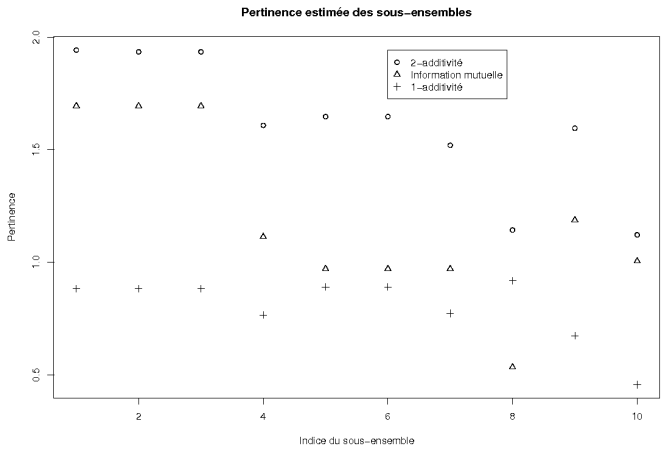
TAB. 6.1 – *Définition de sous-ensembles dont la pertinence va être étudiée.*

l'approximation que nous avons choisie, c'est à dire la troncature 2-additive de l'information mutuelle.

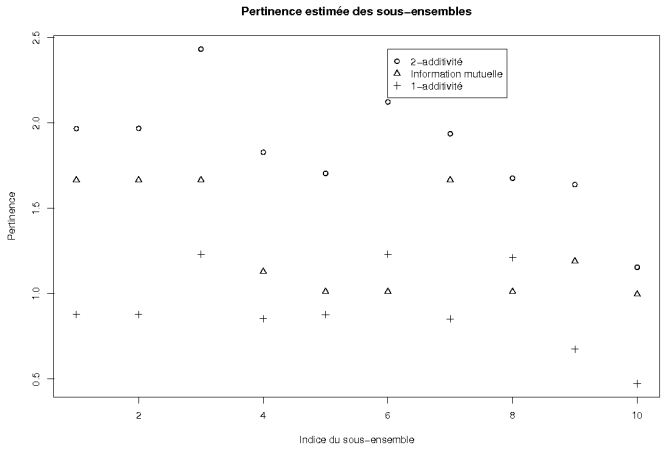
6.1.1 Données non bruitées

Nous présentons dans la Figure 6.1 la pertinence des sous-ensembles définis dans le Tableau 6.1 pour les deux jeux de données non bruitées à 15 variables et 800 individus.

Sur la Figure 6.1(a), l'approximation 2-additive se comporte comme l'information mutuelle classique, ce qui n'est pas le cas de l'approximation 1-additive. Si l'on considère l'ensemble 8, par exemple, il est composé de deux variables pertinentes et deux variables redondantes. L'approximation 2-additive permet bien de détecter cette redondance en considérant cet ensemble comme le moins pertinent. L'approximation 1-additive le classe, par contre, dans les ensembles les plus pertinents. La redondance des variables semble donc être bien détectée par l'approximation 2-additive de l'information mutuelle. Sur la Figure 6.1(b), la troncature 2-additive suit moins exactement le calcul classique mais l'allure générale. Ce résultat était attendu vu que la troncature 2-additive ne prend pas en compte les interactions entre plus de deux variables. La redondance est encore bien détectée avec l'ensemble 8 qui est peu pertinent pour la 2-additivité. Par contre, l'ensemble 3 est détecté comme le plus pertinent alors que la variable X_9 apporte de l'information puisqu'elle contient la variable X_1 mais aussi de la redondance avec les variables X_2 et X_3. Mais l'ensemble 3 restant un très bon sous-ensemble, ce résultat n'est pas aberrant. Nous notons toutefois que la troncature 2-additive ne reconnaît pas l'ensemble 7 comme plus pertinent que l'ensemble 6 alors que le premier contient l'information de la variable X_4 contrairement au deuxième. Sur ces deux graphiques, nous voyons bien que l'approximation 2-additive permet de détecter des redondances que ne détecte pas l'approximation 1-additive tout en ne faussant pas trop la pertinence des autres sous-ensembles.



(a) Variables en relations simples (Section 5.3.1).



(b) Variables en relations complexes (Section 5.3.2).

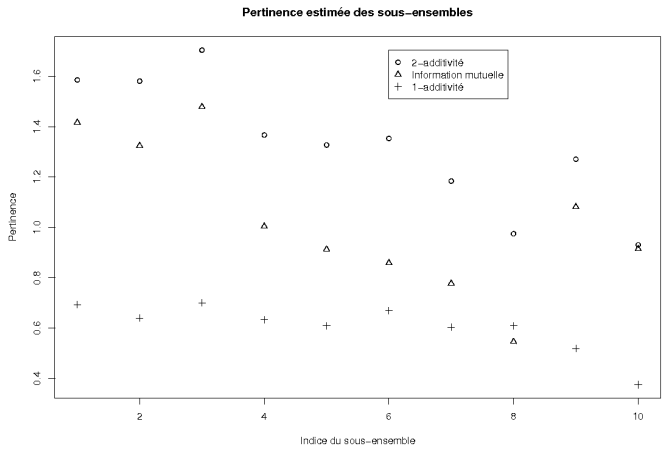
FIG. 6.1 – *Pertinence de la sélection de sous-ensembles sur des données de 800 individus non bruitées.*

6.1.2 Données légèrement bruitées

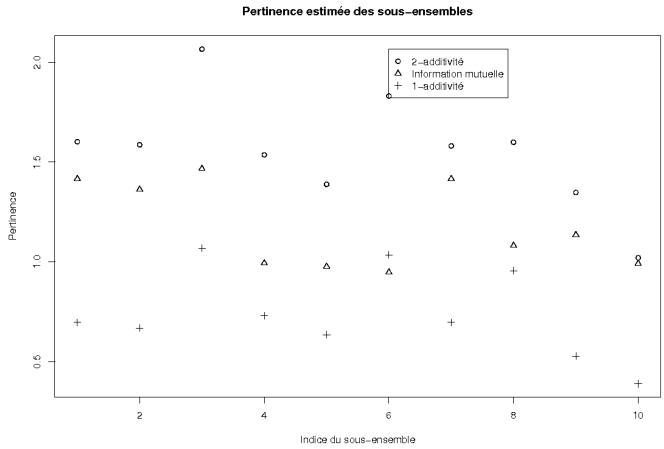
Nous présentons dans la Figure 6.2 la pertinence des sous-ensembles définis dans le Tableau 6.1 pour les deux jeux de données légèrement bruitées à 15 variables et 800 individus.

6.1.3 Données très bruitées

Nous présentons dans la Figure 6.3 la pertinence des sous-ensembles définis dans le Tableau 6.1 pour les deux jeux de données très bruitées à 15 variables et 800 individus.



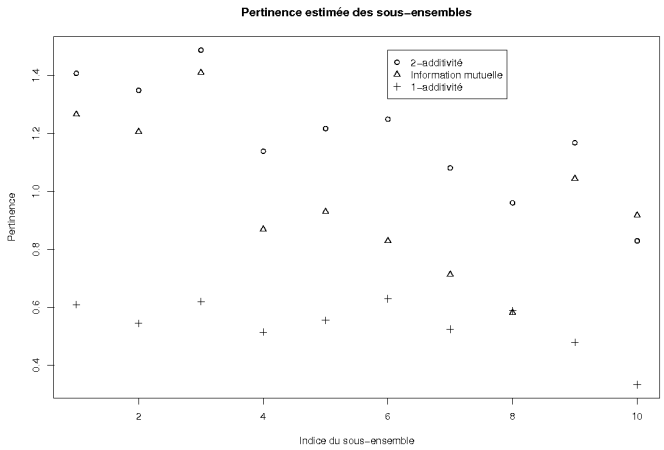
(a) Variables en relations simples (Section 5.3.1).



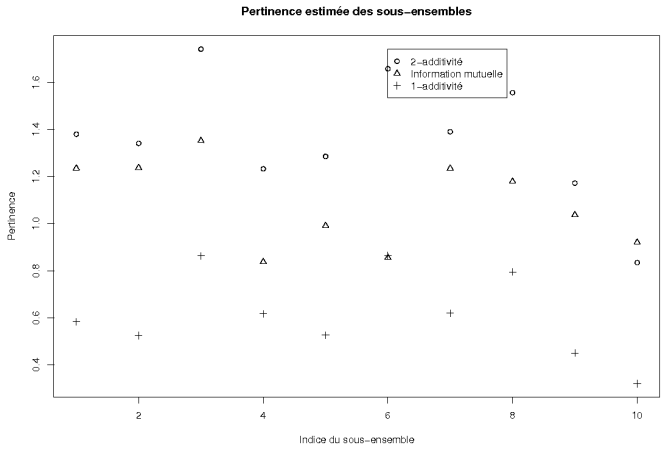
(b) Variables en relations complexes (Section 5.3.2).

FIG. 6.2 – *Pertinence de la sélection de sous-ensembles sur des données de 800 individus légèrement bruitées.*

Sur les Figures 6.2(a), 6.2(b), 6.3(a) et 6.3(b), nous remarquons que le bruit n'influe pas le comportement de l'approximation 2-additive qui reste globalement proche de celui de l'information mutuelle sur nos deux jeux de données. Nous étudions plus précisément cette intuition grâce à un test dans la sous-section suivante.



(a) Variables en relations simples (Section 5.3.1).



(b) Variables en relations complexes (Section 5.3.2).

FIG. 6.3 – *Pertinence de la sélection de sous-ensembles sur des données de 800 individus très bruitées.*

6.2 Résistance au bruit et à la taille de l'information mutuelle en tant que mesure de pertinence

Une mesure de pertinence de qualité doit être robuste : le bruitage et la taille des données ne doivent pas changer fondamentalement son comportement. C'est ce que nous allons maintenant regarder. Nous avons vu dans la Section 6.1 qu'une approximation 2-additive de l'information mutuelle était satisfaisante. C'est pourquoi ce test va être effectué avec une troncature 2-additive de l'information mutuelle estimée classiquement comme mesure de pertinence.

6.2.1 Résistance au bruit

Nous étudions la pertinence des sous-ensembles présentés dans le Tableau 6.1 sur trois jeux de données. Ces trois jeux sont basés sur le jeu à 15 variables (Section 5.3.1). Le premier est le jeu tel quel, c'est à dire non bruité. Le deuxième est le jeu légèrement bruité et enfin le troisième est le jeu très bruité (cf. Section 5.3.5 pour la définition du bruit). Pour chacun de ces trois jeux de données, nous avons généré 800 réalisations du vecteur aléatoire $(X_1, \ldots, X_{15}, Y)$.

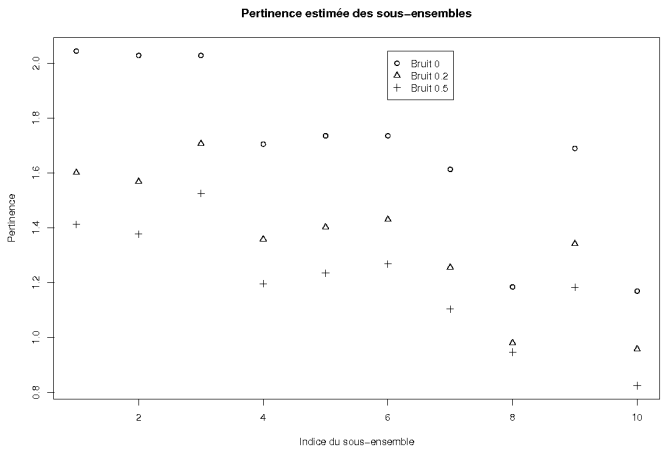


FIG. 6.4 – *Résistance au bruit de la mesure de pertinence sur un jeu de 800 individus.*

Les résultats présentés sur la Figure 6.4 sont très satisfaisants puisque les trois courbes ont la même allure. En effet, quel que soit le bruit ajouté aux données de départ, l'approximation 2-additive se comporte de la même façon : les ensembles 8 et 10 sont les moins pertinents, les ensembles 1,2 et 3 sont les plus pertinents. Cette mesure de pertinence semble donc être robuste au bruitage des données sur cet exemple.

6.2.2 Résistance à la taille

Une bonne mesure ne doit pas être influencée par la taille des données. C'est ce qui nous intéresse maintenant : la troncature 2-additive de l'information mutuelle en tant que mesure de pertinence résiste-t-elle bien à la taille des données ? Pour cela, nous utilisons le jeu à 15 variables (Section 5.3.1) légèrement bruité. Nous avons généré $200, 800$ et 1600 réalisations du vecteur aléatoire $(X_1, \ldots, X_{15}, Y)$ et nous avons regardé la pertinence estimée des sous-ensembles du Tableau 6.1 sur chacun des jeux de données.

Nous voyons que la Figure 6.2.2 que dès lors que le nombre d'individus est suffisamment grand l'approximation 2-additive ne varie plus avec la taille. Pour le jeu à 200 individus, la courbe est éloignée des autres mais elle a la même évolution. Ce graphique ne rejette donc pas l'hypothèse selon laquelle la troncature 2-additive de l'information

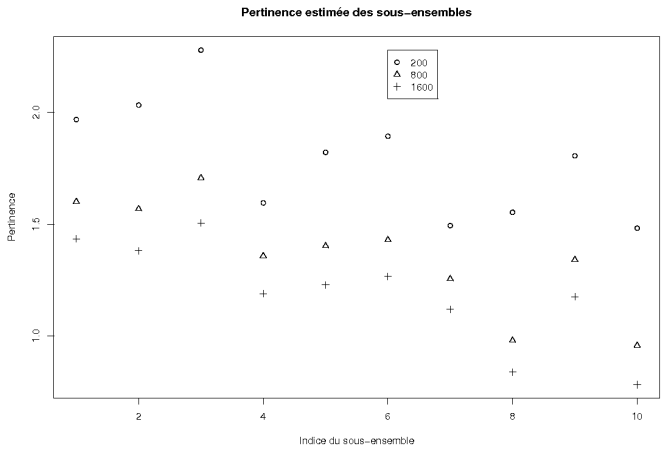


FIG. 6.5 – *Résistance à la taille de la mesure de pertinence sur un jeu de 800 individus légèrement bruité.*

mutuelle est une mesure de pertinence résistante à la taille des données. Cependant, ce résultat est à modérer si nous tenons compte des travaux de Bennani et al. (2005) qui montrent qu'en grandes dimensions, la pertinence d'une variable dépend de la taille de l'échantillon.

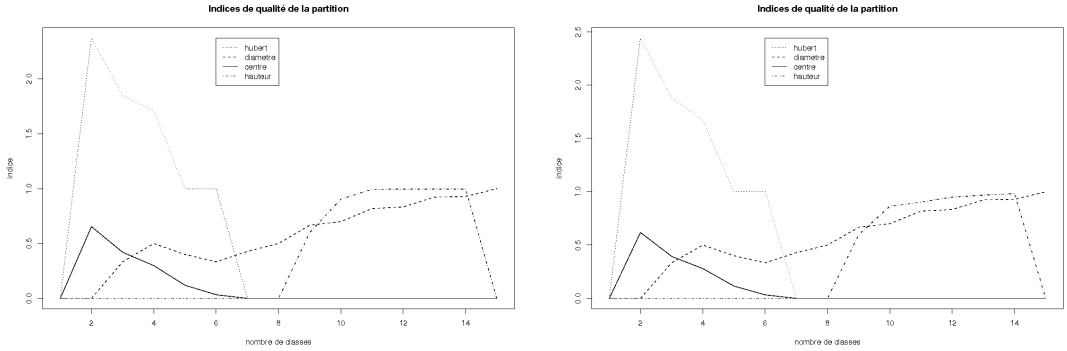
6.3 Comparaison des indices de qualité d'une partition

Ce test ne porte pas sur la mesure de pertinence mais sur l'estimation de la qualité d'une partition. Nous avons présenté en Section 4.3.1 différents indices permettant d'évaluer la qualité d'une partition et donc de choisir un nombre de classes « optimal ». Il nous a semblé intéressant de voir comment chacun d'eux se comportait sur un jeu de données ayant une structure bien connue (i.e le nombre de classes est ici donné). Nous utilisons donc le jeu de données à 15 variables (Section 5.3.1). Le principe de cette expérimentation est le suivant : nous générons toutes les partitions possibles de l'ensemble des variables et nous évaluons à l'aide des cinq critères définis dans la Section 4.3.1. Nous comparons l'évolution de ces indices en fonction du nombre de classes. Ainsi, à chaque étape de la hiérarchie des partitions, nous comparons la valeur de l'indice à l'étape courante avec la valeur de l'indice à l'étape précédente. En effet, généralement, un saut dans l'évolution de l'indice indique une bonne partition. Les résultats sur l'étude des écarts sont présentés en annexe D.1.

Connaissant bien notre jeu de données, nous nous attendons à ce qu'une bonne partition ait 4 à 7 classes. En effet, si la partition a plus de 8 classes, nous avons des variables très redondantes qui n'appartiennent pas à la même classe. À l'inverse, si le jeu possède moins de 4 classes, nous avons des variables indépendantes dans la même classe.

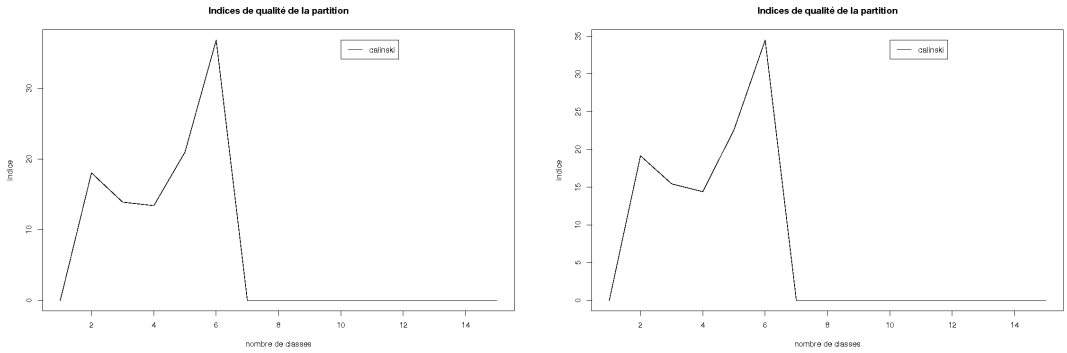
Si deux variables indépendantes sont très complémentaires sur l'information qu'elles apportent pour expliquer Y et qu'elles se retrouvent dans la même classe, alors nous allons perdre l'information apportée par l'une d'elle.

Nous avons généré 800 réalisations du vecteur aléatoire $(X_1, \ldots, X_{15}, Y)$. Nous avons effectué ce test en modifiant l'indice de similarité, base de la construction de la classification : l'information mutuelle estimée classiquement et l'information mutuelle (Section 2.3.1) avec estimation bayésienne (Section 2.3.2).



(a) Estimation classique. (b) Estimation bayésienne.

FIG. 6.6 – *Evolution des indices de qualité d'une partition sur des données de 800 individus.*



(a) Estimation classique. (b) Estimation bayésienne.

FIG. 6.7 – *Evolution de l'indice de Calinski pour évaluer la qualité d'une partition sur des données de 800 individus.*

Si nous regardons les figures 6.6 et 6.7, nous voyons déjà que quelle que soit l'estimation de l'information mutuelle choisie, les indices de qualité ont le même comportement. L'indice de Calinski propose la meilleure partition en 6 classes, tout comme l'indice de Hubert. L'indice du centre de la classe nous fait plutôt choisir une partition en 4 classes de même que l'indice du diamètre même si ce choix est moins net. Quant à l'indice

de la hauteur, il permet de choisir une partition en 8 classes. Nous voyons donc que les cinq indices se comportement différemment mais semblent tous fournir un résultat satisfaisant.

6.4 Résistance au bruit et à la taille des données de l'indice de qualité

Le but de ce dernier test est de s'assurer que les indices de qualité sélectionnés sont résistants au bruitage et à la taille des données. Pour cela, nous utilisons toujours le jeu de données dont la structure est bien connue à savoir le jeu de données à 15 variables (Section 5.3.1).

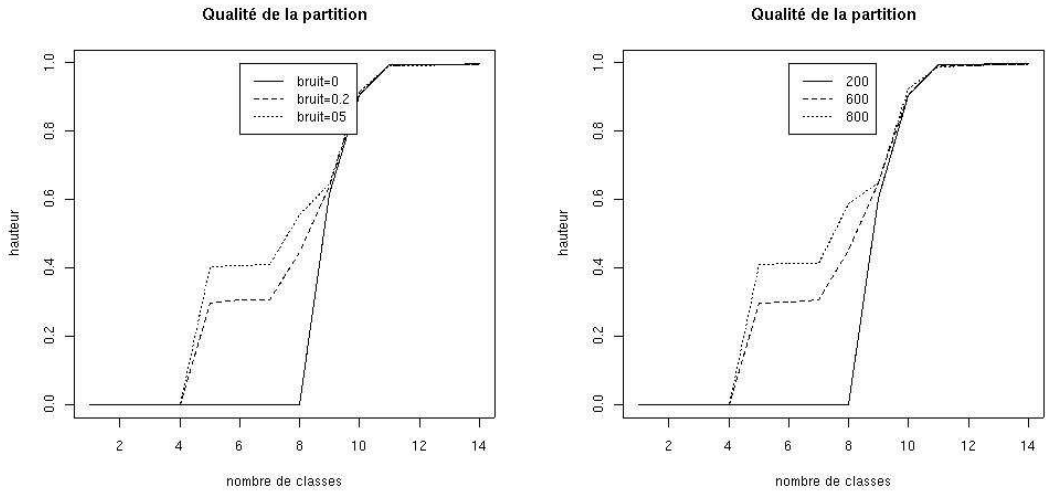
Pour chaque indice de qualité, nous regardons si l'indice a une évolution dépendante de la taille des données et du bruit ajouté aux données. Pour chacun des indices de qualité d'une partition étudié, nous effectuons deux comparaisons. Tout d'abord, nous étudions le comportement des indices sur des jeux de 15 variables pour trois niveaux de bruit comme nous l'avons décrit dans la section 5.3.5. Ensuite, nous comparons l'évolution de l'indice sur trois jeux à 15 variables de tailles différentes (de 200 à 1600 individus). Nous effectuons ce test en variant l'estimation de l'information mutuelle : estimation classique, estimation probabiliste et estimation bayésienne. La variation de l'estimation de l'information mutuelle ne change pas le comportement des indices de qualité, ce qui était très souhaitable. Nous ne commentons donc pas tous les graphiques issus de ce test ici (voir l'annexe D.2). Mais, nous présentons les résultats par indice de qualité indépendamment de l'estimation de l'information mutuelle choisie.

6.4.1 Indice de la hauteur

La figure 6.8 nous montre que l'indice de la hauteur a une évolution proche quel que soit le bruit des données et quelle que soit la taille. Par contre, nous voyons bien que pour les données bruitées ou les données avec plus d'individus, un saut supplémentaire apparaît entre les partitions à 4 et 5 classes. Du bruit sur les données peut donc nous inciter à choisir une partition en 4 classes plutôt qu'en 8 classes. Ces deux partitions semblant être très correctes, nous ne pouvons pas considérer ce comportement comme un problème de résistance au bruit.

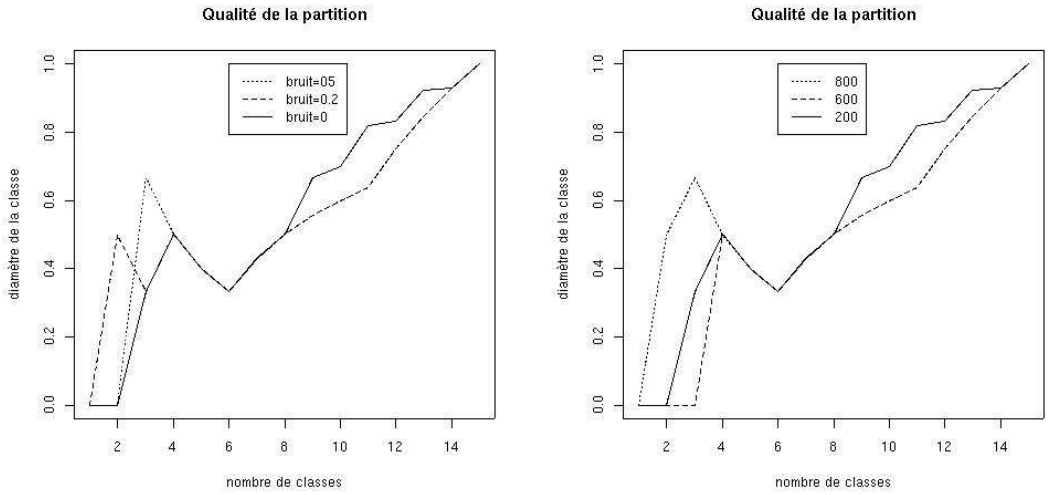
6.4.2 Indice du diamètre moyen

L'indice du diamètre moyen est bien résistant au bruit et à la taille des données, comme nous le montre la figure 6.9.



(a) Résistance au bruit. (b) Résistance à la taille.

FIG. 6.8 – *Robustesse de l'indice de la hauteur pour une classification basée sur une information mutuelle estimée classiquement.*



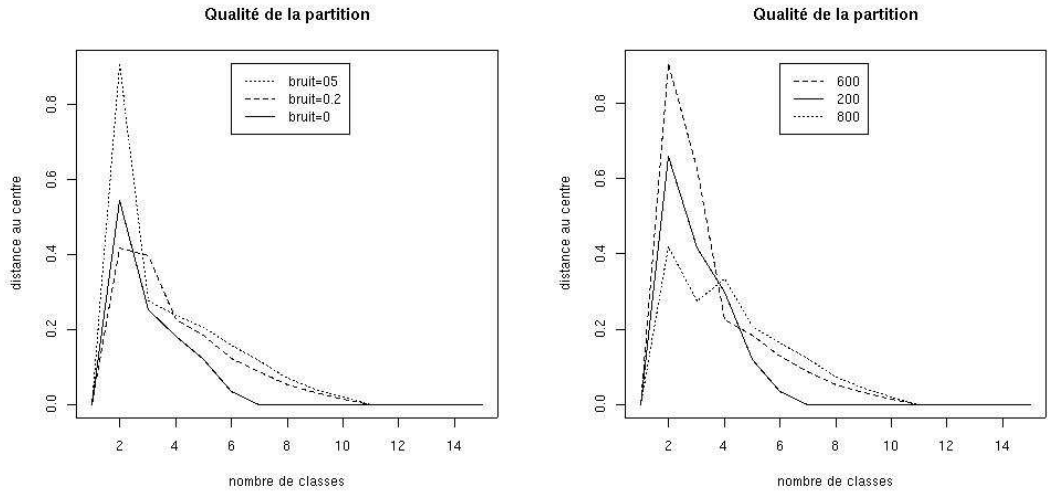
(a) Résistance au bruit. (b) Résistance à la taille.

FIG. 6.9 – *Robustesse de l'indice du diamètre pour une classification basée sur une information mutuelle estimée classiquement.*

6.4.3 Indice de la distance au centre de la classe

L'indice la distance au centre de la classe est bien résistant au bruit et à la taille des données, comme nous le montre la figure 6.9. Il est intéressant de remarquer que sur

la recherche d'un saut, les indices du diamètre et du centre semblent donner le même résultat autour de 4 classes, ces deux indices n'évoluent pas du tout de la même façon avec l'augmentation du nombre de classes de la partition.



(a) Résistance au bruit.

(b) Résistance à la taille.

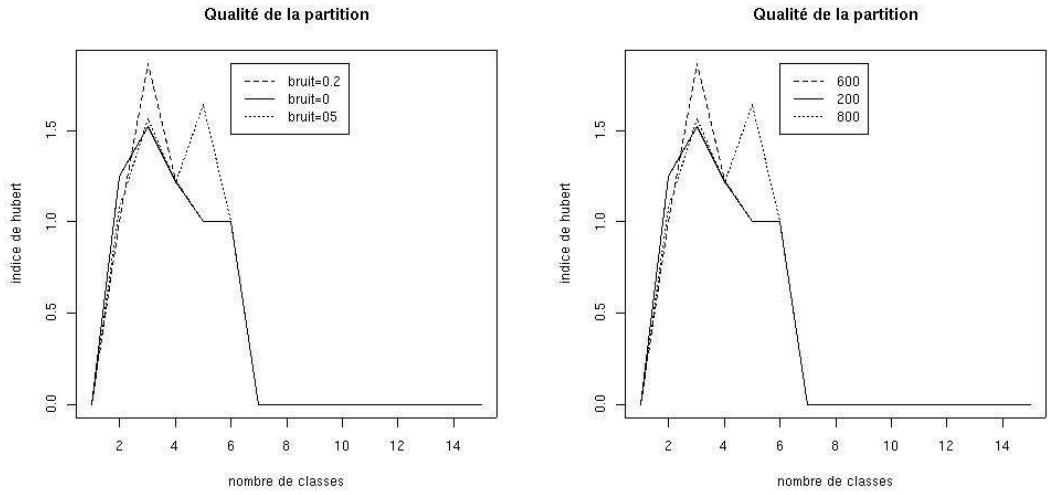
FIG. 6.10 – *Robustesse de l'indice de la distance au centre de la classe pour une classification basée sur une information mutuelle estimée classiquement.*

6.4.4 Indice de Hubert

L'indice de Hubert a un comportement très acceptable du point de vue résistance au bruit et à la taille des données comme nous le voyons sur la figure 6.11. Cependant, comme pour l'indice de la hauteur, nous remarquons que le bruit ou la taille des données peuvent faire changer d'avis le décideur sur le nombre de classes de la partition optimale. Toutefois, là encore, la variation du nombre de classes n'est pas très élevée.

6.4.5 Indice de Calinski

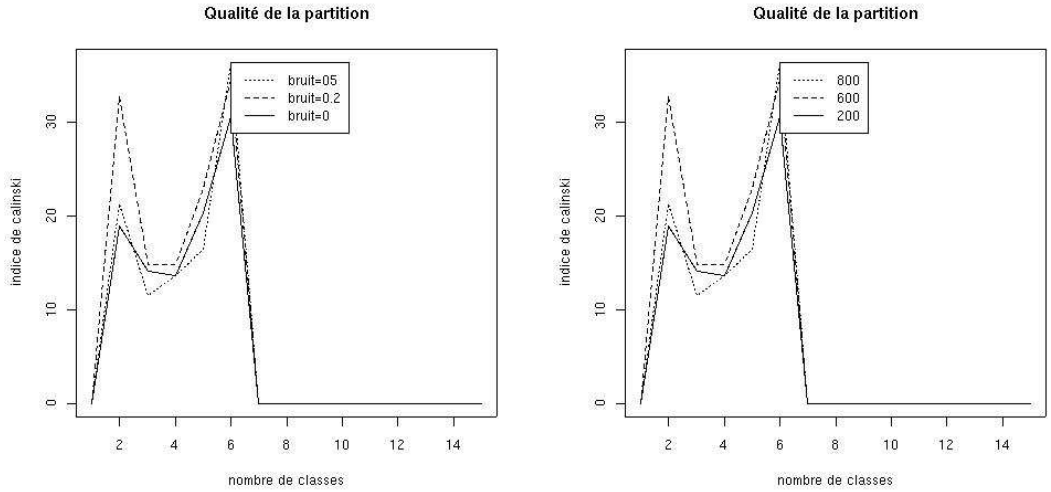
La figure 6.12 met en avant le fait que l'indice de Calinski est très résistant au bruit et à la taille des données. En effet, il se comporte exactement de la même façon dans tous les cas. La seule différence est sur l'intensité de sa variation. Mais cette différence est accentuée par le fait que l'indice n'est pas normé.



(a) Résistance au bruit.

(b) Résistance à la taille.

FIG. 6.11 – *Robustesse de l'indice de Hubert pour une classification basée sur une information mutuelle estimée avec une approche bayésienne.*



(a) Résistance au bruit.

(b) Résistance à la taille.

FIG. 6.12 – *Robustesse de l'indice de Calinski pour une classification basée sur une information mutuelle estimée avec une approche bayésienne.*

Conclusion

Dans cette première partie de tests, nous avons étudié la qualité des mesures que nous utilisons, que ce soit l'information mutuelle comme mesure de pertinence ou les

indices de Calinski, Hubert, etc, comme indices de qualité d'une partition.

Concernant l'approximation 2-additive, nous avons vu qu'elle détecte des redondances que ne détecte pas l'approximation 1-additive (utilisée assez classiquement en sélection de variables) tout en ne faussant pas trop la pertinence des autres sous-ensembles par rapport à une estimation classique de l'information mutuelle. De plus, la troncature 2-additive semble assez résistance au bruit sur les données et à la taille de celles-ci. Ces premiers tests sont des résultats encourageants sur l'utilisation de cette mesure de pertinence pour notre algorithme de sélection de variables, ClassAdd.

Concernant les indices de qualité d'une partition, les expérimentations effectuées nous permettent aussi de les utiliser assez sereinement. En effet, ils fournissent tous des résultats satisfaisants sur le nombre de classes de la partition « idéale ». Ces indices sont classiquement utilisés sur des individus. Il est donc intéressant de noter aussi leur pertinence pour la classification d'un ensemble de variables. Nous notons tout de même que leur résistance au bruit ou à la taille des données n'est pas homogène et peut varier selon les indices. Notre analyse confirme que selon les données, certains indices sont plus adaptés que d'autres. Ainsi, dans le chapitre suivant, nous conservons, pour les comparaisons, un ensemble d'indices plutôt qu'un seul.

7

Expérimentations numériques

Résumé

Ce chapitre présente les expérimentations numériques que nous avons faites sur notre algorithme : comparaison d'algorithmes de génération des sous-ensembles et étude de la qualité des résultats obtenus avec ClassAdd.

La section 7.1 décrit les expérimentations que nous avons effectuées pour comparer nos heuristiques de génération des sous-ensembles à évaluer. Une fois les variables structurées en classes, nous proposons de générer tous les sous-ensembles composés d'une seule variable par classe. Nous parlons d'une procédure de génération exhaustive. Notre deuxième proposition est de ne garder que les deux meilleures variables de chaque classe afin de diminuer leur taille et d'effectuer une génération exhaustive sur ces classes de petite taille. La dernière heuristique que nous envisageons est de construire ces classes réduites via un choix aléatoire des variables conservées. Nous présentons dans cette partie les résultats de l'algorithme ClassAdd sur les données de l'entreprise PerformanSe.

Dans la section 7.2, nous proposons une validation de notre algorithme. Nous utilisons les arbres de décisions et le taux d'erreur de classement avec ceux-ci comme outil de validation.

Sommaire

Introduction

Avec l'apparition du traitement de données décrites par un grand nombre de variables telles que les données génétiques, les algorithmes de sélection de variables se sont multipliés dans la littérature. Les évaluer n'est pas une chose aisée étant donné que généralement, chaque algorithme se comporte mieux que les autres dans certains cas et moins bien dans d'autres cas.

Ici, nous présentons tout d'abord, les méthodes d'évaluation d'algorithmes de sélection de variables rencontrées dans la littérature. Puis nous effectuons une comparaison de nos trois procédures de génération des sous-ensembles à évaluer : génération exhaustive de tous les sous-ensembles composés d'au plus une variable par classe, génération de tous les sous-ensembles possibles en ne gardant que les deux meilleures variables dans chaque classe et enfin génération de tous les sous-ensembles possibles en gardant deux variables au hasard dans chaque classe. Enfin, nous effectuons une validation classique de notre travail en utilisant un arbre de décision comme outil de validation.

Pour ce chapitre, nous nous sommes basés sur le travail de Kohavi and John (1997). Les auteurs proposent une méthode assez générique d'évaluation des algorithmes de sélection de variables. Leur approche est dépendante du modèle mais les méthodes de validation n'en restent pas moins intéressantes. Les expérimentations ont lieu sur des jeux classiques issus de *University of California at Irvine repository* D.J. Newman and Merz (1998) : DNA, Soybean, Corral, Monk, Breast Cancer, etc.

Le premier test proposé par Kohavi and John (1997) consiste à utiliser les algorithmes classiques ID3 et Naive-Bayes comme outil de validation. Pour chaque jeu de données, la précision de l'algorithme est calculée sur les données complètes puis sur les données restreintes aux variables sélectionnées par un algorithme préalable de sélection de variables.

Un deuxième point observé est le nombre de variables nécessaires à la construction d'un arbre avec ID3 avec ou sans algorithme de sélection de variables préalable. Cette étude permet de vérifier que l'algorithme ID3 ne suffit sans doute pas pour être utilisé seul comme algorithme de sélection de variables.

7.1 Comparaison des trois algorithmes de génération de sous-ensembles

Une fois la classification effectuée, nos variables potentiellement discriminantes se trouvent donc structurées en k classes. Notre heuristique consiste à ne garder qu'au plus une variable par classe. Nous supposons donc que la classification est suffisamment homogène et qu'ainsi, les variables d'une même classe sont suffisamment redondantes pour n'en prendre qu'une seule.

Dans ce test, nous comparons trois façons de générer les sous-ensembles à évaluer. La première est exhaustive, elle consiste à générer tous les sous-ensembles possibles contenant au plus une variable par classe. Cette première approche est déjà moins coûteuse que l'exhaustivité sans classification préalable mais elle reste quand même assez fastidieuse. En effet, le nombre de sous-ensembles à évaluer est de l'ordre de $|C_1| \times \cdots \times |C_k|$, où $|C_i|$ exprime la cardinalité de la classe i. Nous allons donc comparer la génération exhaustive avec des approches réduisant le nombre de sous-ensembles potentiels à évaluer. Tout d'abord, nous utilisons le hasard qui consiste à ne garder que v variables par classe, choisies au hasard. Dans ce test, nous n'initialisons qu'une fois les variables gardées mais il est envisageable de les initialiser plusieurs fois sur le principe des k-means. Puis, nous comparons enfin ces deux modes de génération avec un troisième qui consiste à ne garder que v variables par classe mais cette fois-ci les *meilleures*. Nous définissons les meilleures variables par les variables apportant le plus d'information pour expliquer Y, c'est à dire les variables les plus pertinentes au sens de la mesure de pertinence choisie.

Nous avons effectué les tests sur différents jeux de données en fixant $v = 2$ et en choisissant une estimation classique de l'information mutuelle avec approximation 2-additive.

7.1.1 Les données à 22 variables

En utilisant l'indice de la hauteur, nous obtenons une partition optimale du jeu de données à 22 variables en 8 classes. Nous recherchons maintenant les meilleurs sous-ensemble de taille 1 à 8. Cette recherche est répétée trois fois en suivant les trois procédures de génération de sous-ensembles.

Avec un parcours exhaustif, les meilleurs sous-ensembles retournés sont :

$$\{X_1, X_5\}, \{X_1, X_2, X_5\}, \{X_1, X_2, X_3, X_4\}, \{X_1, X_2, X_3, X_4, X_5\}, \{X_1, X_2, X_3, X_4, X_5, X_{22}\}$$

$$\{X_1, X_2, X_3, X_4, X_5, X_{16}, X_{22}\}, \{X_1, X_2, X_3, X_4, X_5, X_{16}, X_{21}, X_{22}\}$$

Avec un parcours au hasard, les meilleurs sous-ensembles retournés sont :

$$\{X_5, X_6\}, \{X_2, X_5, X_6\}, \{X_2, X_3, X_4, X_6\}, \{X_2, X_3, X_4, X_5, X_6\}, \{X_2, X_3, X_4, X_5, X_6, X_{22}\}$$

$$\{X_2, X_3, X_4, X_5, X_6, X_{17}, X_{22}\}, \{X_2, X_3, X_4, X_5, X_6, X_{17}, X_{21}, X_{22}\}$$

Avec un parcours des *meilleures* variables, les meilleurs sous-ensembles retournés sont :

$$\{X_1, X_5\}, \{X_1, X_2, X_5\}, \{X_1, X_2, X_3, X_4\}, \{X_1, X_2, X_3, X_4, X_5\}, \{X_1, X_2, X_3, X_4, X_5, X_{22}\}$$

$$\{X_1, X_2, X_3, X_4, X_5, X_{16}, X_{22}\}, \{X_1, X_2, X_3, X_4, X_5, X_{16}, X_{21}, X_{22}\}$$

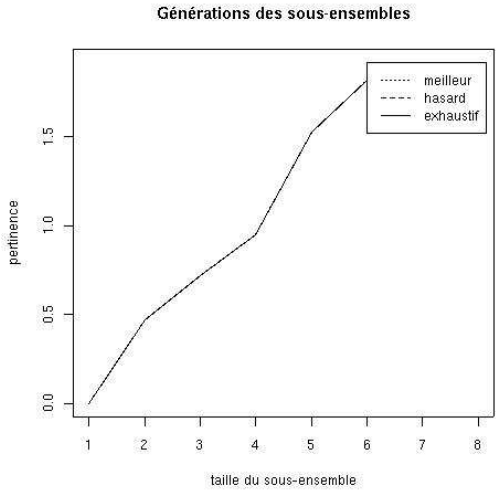


FIG. 7.1 – *Pertinence des sous-ensembles du jeu à 22 variables retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables.*

Nous mesurons maintenant la pertinence de chacun de ces sous-ensembles afin de comparer pour chaque taille de sous-ensemble les résultats obtenus par les différents axes de recherche de la solution. Les résultats sont présentés dans la figure 7.1. Nous voyons que sur un jeu de données simpliste, les trois méthodes sont équivalentes et donnent d'aussi bons résultats. En effet, les sous-ensembles retournés sont bien les sous-ensembles attendus. Nous remarquons que ce type de sélection pénalise plus les variables redondantes que les variables non pertinentes. Cet effet est dû à la structuration en classes homogènes de variables et à la contrainte fixée d'utiliser au plus une variable par classe.

7.1.2 Les données à 35 variables

Nous présentons dans cette section le nombre de classes optimal proposé par chacun des cinq indices dans le tableau 7.1. Ce premier résultat est complété par des graphiques permettant de représenter les pertinences du sous-ensemble de taille 2 au nombre de

Indice	Nombre de classes optimal
Hauteur	12
Diamètre moyen	3
Distance au centre	3
Calinski	13
Huberti	13

TAB. 7.1 – *Nombre de classes optimal pour chacun des critères de qualité d'une partition pour le jeu à 35 variables.*

classes optimal proposé par l'algorithme. Pour illustration, nous donnons aussi les sous-ensembles eux-mêmes retournés par CLASSADD dans le cas de l'indice de la hauteur.

En utilisant l'indice de la hauteur, nous obtenons une partition optimale du jeu de données à 35 variables en 12 classes (tableau 7.1). Nous recherchons maintenant les meilleurs sous-ensembles de taille 1 à 12.

Avec un parcours exhaustif, les meilleurs sous-ensembles retournés sont :

$$\{X_{27}, X_5\}, \{X_{27}, X_2, X_4\}, \{X_{27}, X_2, X_3, X_{34}\}, \{X_{27}, X_2, X_3, X_4, X_{34}\}, \{X_{27}, X_2, X_3, X_4, X_5, X_{34}\}$$

$$\{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{34}\}, \{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{21}, X_{34}\}$$

$$\{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{21}, X_{22}, X_{34}\}, \{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{21}, X_{22}, X_{28}, X_{34}\}$$

$$\{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{21}, X_{22}, X_{28}, X_{29}, X_{34}\}$$

$$\{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{21}, X_{22}, X_{28}, X_{29}, X_{34}, X_{35}\}$$

Avec un parcours au hasard, les meilleurs sous-ensembles retournés sont :

$$\{X_6, X_5\}, \{X_6, X_7, X_5\}, \{X_6, X_7, X_3, X_9\}, \{X_6, X_7, X_3, X_9, X_5\}, \{X_6, X_7, X_3, X_9, X_5, X_{34}\}$$

$$\{X_6, X_7, X_3, X_9, X_5, X_{33}, X_{34}\}, \{X_6, X_7, X_3, X_9, X_5, X_{33}, X_{21}, X_{34}\}$$

$$\{X_6, X_7, X_3, X_9, X_5, X_{33}, X_{21}, X_{22}, X_{34}\}, \{X_6, X_7, X_3, X_9, X_5, X_{33}, X_{21}, X_{22}, X_{28}, X_{34}\}$$

$$\{X_6, X_7, X_3, X_9, X_5, X_{33}, X_{21}, X_{22}, X_{28}, X_{29}, X_{34}\}$$

$$\{X_6, X_7, X_3, X_9, X_5, X_{33}, X_{21}, X_{22}, X_{28}, X_{29}, X_{34}, X_{35}\}$$

Avec un parcours des *meilleures* variables, les meilleurs sous-ensembles retournés sont :

$$\{X_{27}, X_5\}, \{X_{27}, X_2, X_4\}, \{X_{27}, X_2, X_3, X_{34}\}, \{X_{27}, X_2, X_3, X_4, X_{34}\}, \{X_{27}, X_2, X_3, X_4, X_5, X_{34}\}$$

$$\{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{34}\}, \{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{21}, X_{34}\}$$

$$\{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{21}, X_{22}, X_{34}\}, \{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{21}, X_{22}, X_{28}, X_{34}\}$$

$$\{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{21}, X_{22}, X_{28}, X_{29}, X_{34}\}$$

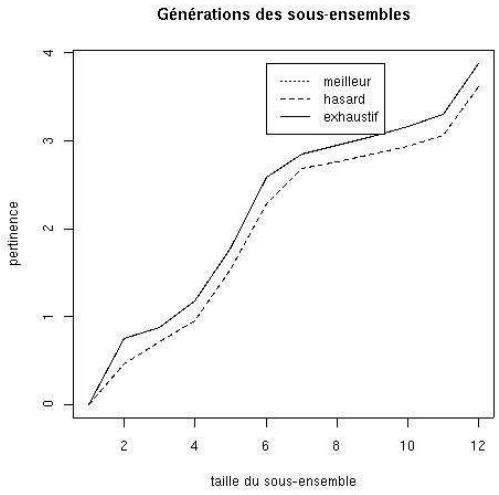


FIG. 7.2 – *Pertinence des sous-ensembles du jeu à 35 variables retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de la hauteur.*

$$\{X_{27}, X_2, X_3, X_4, X_5, X_{33}, X_{21}, X_{22}, X_{28}, X_{29}, X_{34}, X_{35}\}$$

La pertinence des sous-ensembles obtenus est présentée dans la figure 7.2.

Nous voyons sur la figure 7.2 que la qualité des sous-ensembles retournés par une approche exhaustive et une approche réduite aux meilleures variables est équivalente. La génération en utilisant le hasard est certes un peu moins bonne mais la différence n'est pas significative par rapport à la facilité à mettre en œuvre ce mode de génération. En effet, choisir deux variables au hasard est moins coûteux que choisir les deux meilleures, ce qui revient à estimer toutes les variables d'une classe et à les classer.

La figure 7.3 nous montre encore une équivalence entre la génération exhaustive des sous-ensembles et la génération réduite aux meilleurs variables. Par contre, nous avons ici une génération au hasard nettement moins performante. Ce résultat est prévisible. Il n'apparaît pas dans les tests précédents pour la simple raison que les partitions retenues découpent les jeux en classes composées en moyenne d'à peine trois variables. L'évènement consistant à choisir deux représentants de classes au hasard parmi trois a donc une forte probabilité de choisir la meilleure variable comme candidate potentielle. Là le diamètre moyen propose une partition en trois classes soit en moyenne onze variables par classes. Le hasard est donc fortement pénalisé par cette structure.

Sur les figures 7.4, 7.5, 7.6, nous retrouvons les mêmes résultats que ceux décrits précédemment.

Connaissant bien la structure de notre jeux de données à 35 variables, nous pouvons déjà, après ces premières expérimentations, effectuer quelques hypothèses. Concernant,

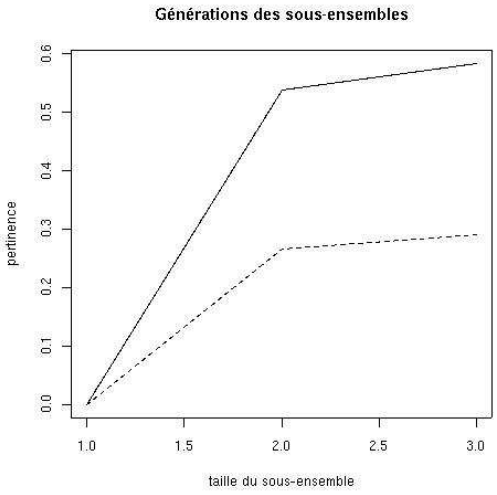


FIG. 7.3 – *Pertinence des sous-ensembles du jeu à 35 variables retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère du diamètre moyen.*

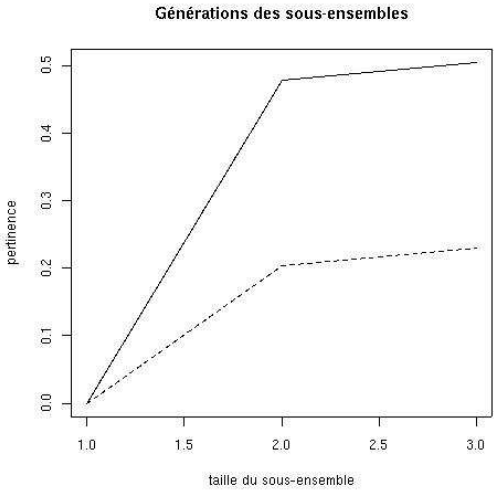


FIG. 7.4 – *Pertinence des sous-ensembles du jeu à 35 variables retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de la distance au centre de la classe.*

les indices de qualité d'une partition, nous semblons avoir deux groupes d'indices : l'indice de la hauteur, l'indice de Calinski et l'indice de Hubert dans un premier groupe et l'indice du diamètre moyen et de la distance au centre de la classe dans un autre. Ce dernier groupe était prévisible de par la définition de ces mesures qui sont finalement très

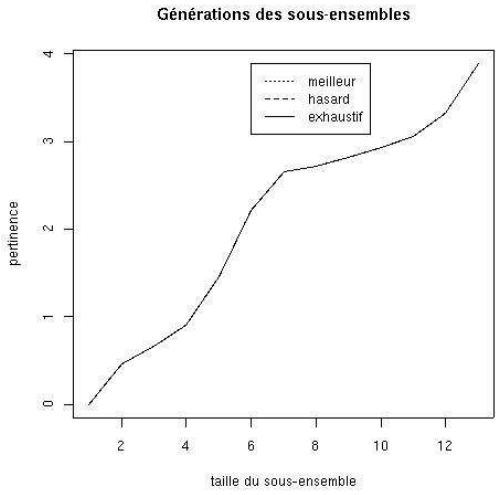


FIG. 7.5 – *Pertinence des sous-ensembles du jeu à 35 variables retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de Calinski.*

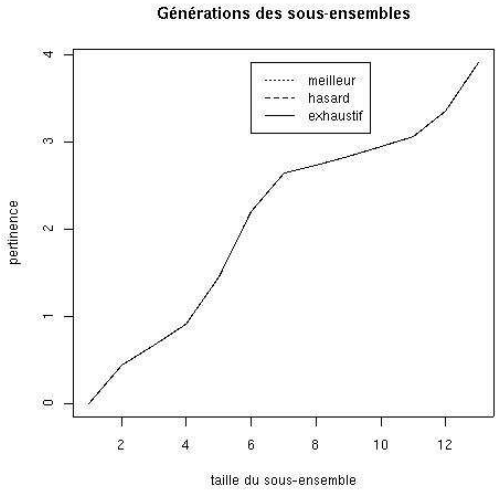


FIG. 7.6 – *Pertinence des sous-ensembles du jeu à 35 variables retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de Hubert.*

proches. En ce qui concerne le groupe hauteur, Calinski et Hubert, il serait intéressant d'étudier leurs liens de manière plus approfondie. Ces groupes d'indices sont les mêmes que ceux apparus dans le test de la section 6.3. Ensuite, nous avons vu que la qualité du parcours au hasard était très liée à la taille des classes. Mais il est dommage de se

Jeu de données	Indice	Nombre de classes optimal
Audiology	Hauteur	59
Connect	Hauteur	7
Lung Cancer	Diamètre moyen	4
OptDigit	Diamètre moyen	3
Lung Cancer	Distance au centre	3
OptDigit	Distance au centre	3
Soybean	Calinski	34
Splice	Calinski	3
Lung Cancer	Hubert	33
OptDigit	Hubert	37

TAB. 7.2 – *Nombre de classes optimal pour chacun des critères de qualité d'une partition sur des jeux de données réelles.*

priver de cette génération très simple à implémenter et peu coûteuse seulement à cause de son comportement peu performant dans certains cas. Nous pourrions donc envisager un algorithme de sélection de variables où la procédure de génération des sous-ensembles ne serait pas fixée à l'avance. Suivant la taille moyenne des classes, l'algorithme choisirait de ne garder que les meilleures variables dans le cas de grandes classes et des variables choisies au hasard dans le cas de petites classes. Evidemment, cette notion de grande et petite classe n'a de sens que par rapport à v, le nombre de variables que l'on souhaite garder pour la génération.

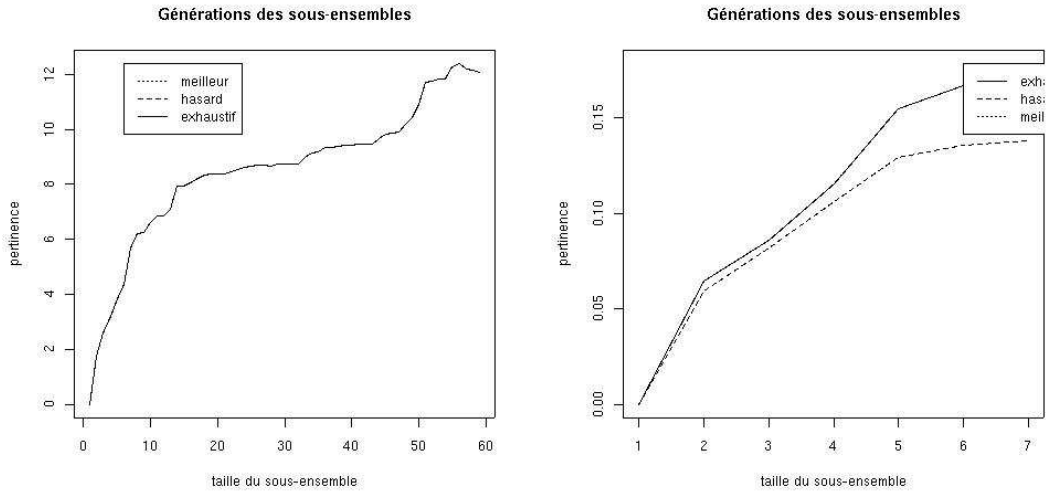
7.1.3 Les données réelles avec une estimation classique de l'information mutuelle

Nous vérifions maintenant que le comportement précédemment décrit se confirme sur les jeux de données réelles. Nous ne présentons pas les résultats sur tous les jeux de données ici. Ils sont disponibles en annexe D.3. Le tableau 7.2 récapitule les combinaisons que nous présentons ici en précisant le nombre de classes optimal.

Nous retrouvons les mêmes comportements sur ces jeux de données réelles que sur les jeux artificiels. Pour une meilleure qualité de notre algorithme, nous devons privilégier les indices construisant des partitions en un plus grand nombre de classes. En effet, si le parcours devient plus coûteux, la contrepartie en est des classes plus homogènes. Ainsi notre heuristique de ne choisir qu'une seule variable par classe est d'autant plus efficace.

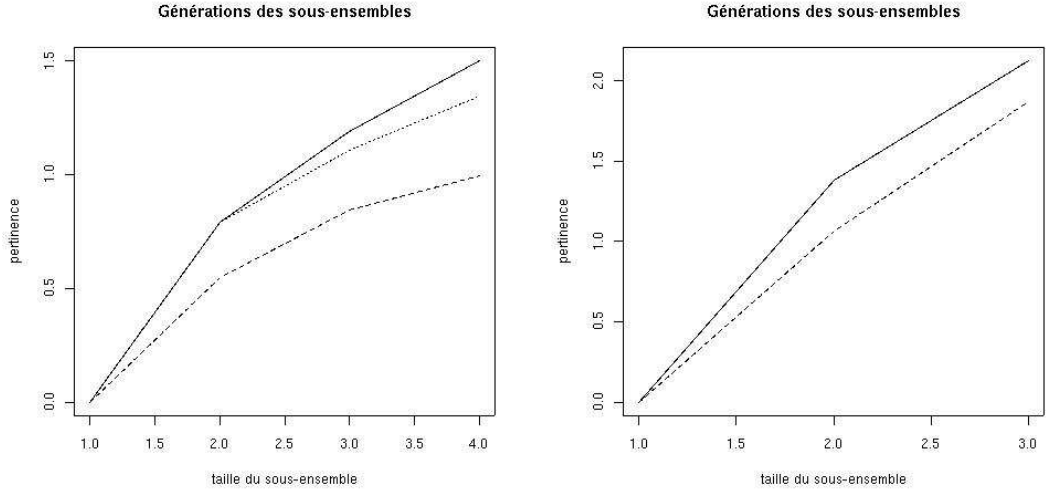
7.1.4 Les données PerformanSe

Pour des raisons de confidentialité, nous ne pouvons présenter toutes les interprétations. Nous avons cependant choisi quelques scénarios qui illustrent notre application.



(a) Données Audiology. (b) Données Connect

FIG. 7.7 – *Pertinence des sous-ensembles des données Audiology et Connect retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de la hauteur.*

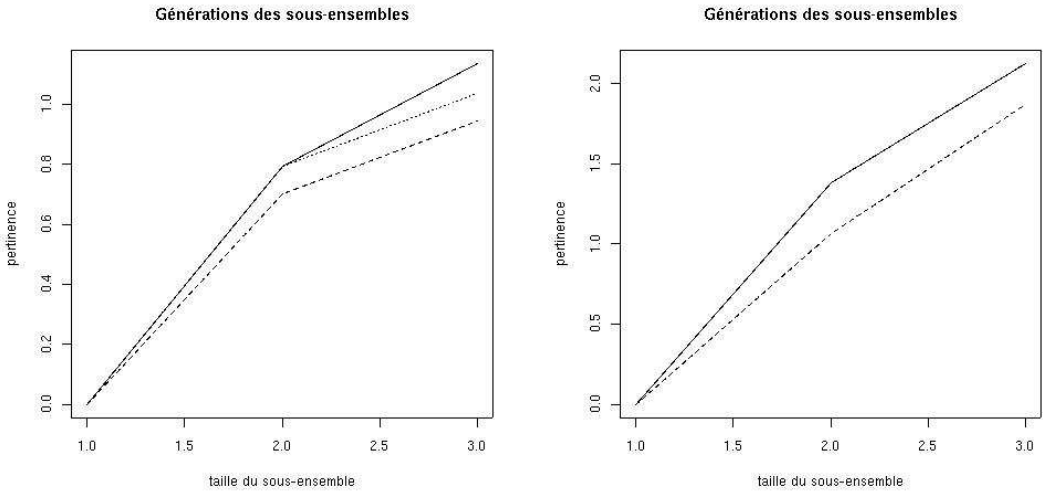


(a) Lung Cancer. (b) Optdigit.

FIG. 7.8 – *Pertinence des sous-ensembles des données Lung Cancer et Optdigit retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère du diamètre moyen.*

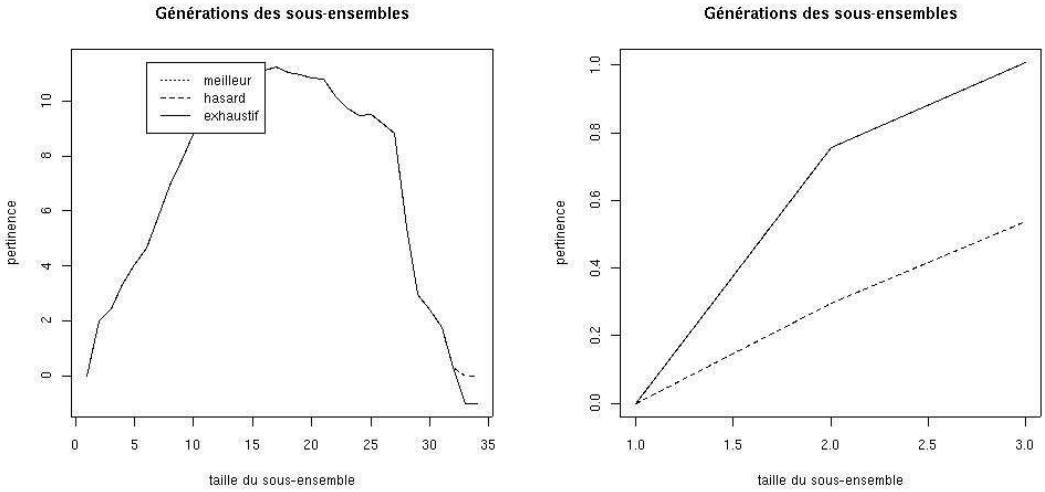
Les données Ages

Nous rappelons que dans ce jeu de données, nous cherchons à expliquer l'âge des cadres en recherche d'emploi. Nous avons effectué le test en choisissant l'indice de Hubert



(a) Lung Cancer. (b) Optdigit.

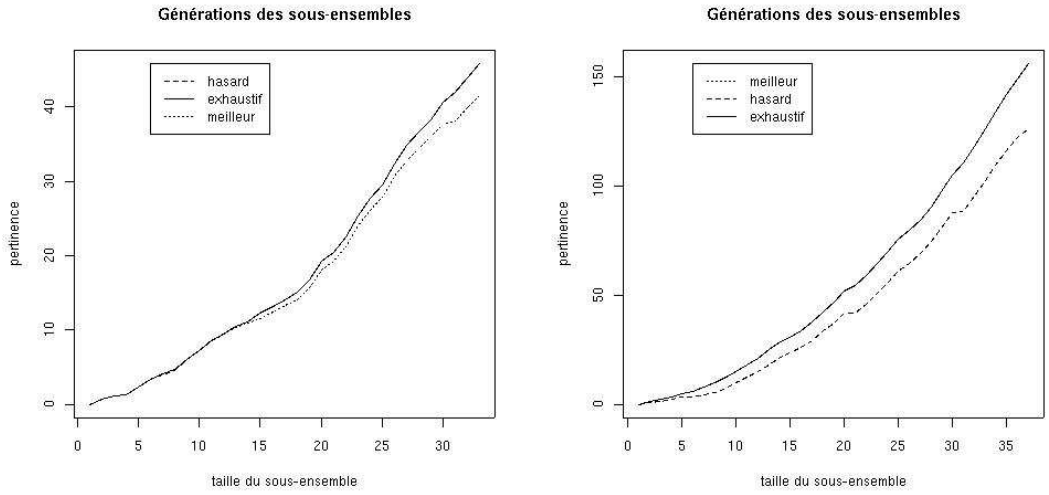
FIG. 7.9 – *Pertinence des sous-ensembles des données Lung Cancer et Optdigit retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de la distance au centre de la classe.*



(a) Soybean. (b) Splice.

FIG. 7.10 – *Pertinence des sous-ensembles des données Soybean et Splice retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de Calinski.*

comme indice de qualité d'une partition et une estimation classique de l'information



(a) Lung Cancer. (b) Optdigit.

FIG. 7.11 – *Pertinence des sous-ensembles des données Lung Cancer et Optdigit retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de Hubert.*

mutuelle comme mesure de pertinence. La partition optimale est en 27 classes.

$$\{H, F\}, \{EXT-, ANX+, AFF-, POU-, ArgumenterA, EncadrerA\}, \{EXT0\}$$

$$\{EXT+, AFF+, POU+, ArgumenterI, EncadrerI\}, \{COM-, REC+\}, \{COM0, COM+, REC-, REC0\}$$

$$\{ANX-, ANX0\}, \{REA-, REA0, REA+\}, \{DIN-, ConcevoirI, ConcevoirN, ConcevoirA\}$$

$$\{DIN0, DIN+, CreerI, CreerN, CreerA\}, \{RIG-, AdministrerA, GererN, GererA, ProduireA\}$$

$$\{RIG0, AdministrerN\}, \{RIG+, AdministrerI, GererI\}, \{AFF0, EncadrerN\}, \{POU0\}$$

$$\{APP-, APP0, APP+\}, \{AdministrerD\}, \{ConcevoirD\}, \{GererD\}, \{ArgumenterD\},$$

$$\{EncadrerD\}, \{EchangerD\}, \{CreerD\}, \{ProduireD\}, \{ArgumenterN\}$$

$$\{EchangerI, EchangerN, EchangerA\}, \{ProduireI, ProduireN\}$$

Nous pouvons déjà effectuer des commentaires sur la partition retournée. En effet, si certains regroupements sont prévisibles tels que les trois niveaux d'Echanger (un niveau exclut les deux autres donc les trois variables sont très liées), d'autres classes de variables sont porteuses d'informations très intéressantes. Prenons la classe $\{EXT+, AFF+, POU+, ArgumenterI, EncadrerI\}$, littéralement, elle signifie qu'avoir des atouts notables pour Argumenter ou Encadrer est très lié à une forte extraversion, une forte affirmation et une forte motivation de pouvoir. La classe $\{RIG+, AdministrerI, GererI\}$ fait le lien entre une forte rigueur et une très bonne capacité à Administrer ou Gérer. La classe $\{DIN0, DIN+, CreerI, CreerN, CreerA\}$ illustre le rapport entre le dynamisme intellectuel et l'activité Créer. Pour les experts en psychologie de l'entreprise PerformanSe,

cette première structuration en classes de variables est déjà très intéressante puisqu'elle permet de faire des liens entre variables du modèle psychologique et activités : liens connus de par la construction des modèles mais aussi liens qui apparaissent avec le jeu de données. Les psychologues ont validé ces regroupements, ce qui conforte l'utilisation d'une classification ascendante hiérarchique en pré-traitement de l'algorithme de sélection de variables. Nous regardons maintenant les meilleurs sous-ensembles de variables proposés pour expliquer l'âge des cadres à la recherche d'emploi. Nous nous arrêtons au sous-ensemble de taille 10.

Avec un parcours exhaustif, les meilleurs sous-ensembles retournés sont :

$$\{H, EncadrerD\}, \{H, AFF-, EncadrerD\}, \{H, AFF-, EXT0, EncadrerD\},$$
$$\{H, AFF-, EXT0, AFF+, EncadrerD\}, \{H, AFF-, EXT0, AFF+, COM-, AFF0\}$$
$$\{H, ArgumenterA, EXT0, AFF+, COM-, COM+, AFF0\}$$
$$\{H, ArgumenterA, EXT0, AFF+, COM-, COM+, ANX-, AFF0\}$$
$$\{H, ArgumenterA, EXT0, AFF+, COM-, COM0, ANX-, REA+, AFF0\}$$
$$\{H, ArgumenterA, EXT0, AFF+, COM-, COM0, ANX-, REA+, ConcevoirN, AFF0\}$$

Avec un parcours au hasard, les meilleurs sous-ensembles retournés sont :

$$\{H, EncadrerD\}, \{H, POU-, EncadrerD\}, \{H, POU-, EXT0, EncadrerD\}$$
$$\{H, POU-, EXT0, POU+, EncadrerD\}, \{H, POU-, EXT0, EXT+, COM-, AFF0\}$$
$$\{H, POU-, EXT0, EXT+, COM-, COM0, AFF0\}$$
$$\{H, POU-, EXT0, EXT+, COM-, COM0, ANX-, AFF0\}$$
$$\{H, POU-, EXT0, EXT+, COM-, COM0, ANX-, REA+, AFF0\}$$
$$\{H, POU-, EXT0, EXT+, COM-, COM0, ANX-, REA+, ConcevoirN, AFF0\}$$

Avec un parcours des *meilleures* variables, les meilleurs sous-ensembles retournés sont :

$$\{H, EncadrerD\}, \{H, AFF-, EncadrerD\}, \{H, AFF-, EXT0, EncadrerD\}$$
$$\{H, AFF-, EXT0, AFF+, EncadrerD\}, \{H, AFF-, EXT0, AFF+, COM-, AFF0\}$$
$$\{H, ArgumenterA, EXT0, AFF+, COM-, COM+, AFF0\}$$
$$\{H, ArgumenterA, EXT0, AFF+, COM-, COM+, ANX-, AFF0\}$$
$$\{H, ArgumenterA, EXT0, AFF+, COM-, COM0, ANX-, REA+, AFF0\}$$
$$\{H, ArgumenterA, EXT0, AFF+, COM-, COM0, ANX-, REA+, ConcevoirN, AFF0\}$$

L'heuristique de ne garder que les meilleures variables fournit le même résultat que le parcours exhaustif de tous les sous-ensembles possibles. L'âge des cadres à la recherche d'emploi semble pouvoir s'expliquer avec les activités Argumenter, Concevoir et Encadrer et avec les traits de personnalités Extraversion, Affirmation, Combativité, Anxiété et Réalisation. Le parcours au hasard donne à peu près les mêmes variables mais il identifie plutôt la motivation de pouvoir que l'activité Argumenter pour expliquer l'âge des cadres à la recherche d'emploi.

Les données Activités Oriente

Avec ce jeu de données, nous cherchons à expliquer une aptitude ou non pour une activité donnée. Nous présentons les résultats du test en utilisant l'indice de la hauteur comme indice de qualité d'une partition et une estimation classique de l'information mutuelle comme mesure de pertinence. La partition optimale est en 114 classes. Nous essayons dans ce test d'expliquer au mieux l'activité Administrer.

$$\{H/F\},\{Locale\},\{Activitechoisie\},\{Itemchoisis\},\{Age\}$$

$$\{Diplome, Statut\},\{Classeur\},\{Concevoir\},\{Gerer\}$$

$$\{Argumenter\},\{Encadrer\},\{Echanger\}\{Creer\},\{Produire\}$$

$$\{EXTEq, EXT-\},\{EXTAmp\},\{EXTTendEXT+\},\{EXT4060\},$$

$$\{ANXEq, ANX0\},\{ANXAmp\},\{ANXTend, ANX-\},\{ANXAmp\}$$

$$\{AFFEq, AFF0\},\{AFFAmp\},\{AFFTend, AFF-\},\{AFF+\},\{RECEq, REC-\}$$

$$\{RECAmp\},\{RECTend, REC+\},\{REC0\},\{RIGEq, RIG-\},\{RIGAmp\}$$

$$\{RIGTend, RIG+\},\{RIG0\},\{DINEq, DIN+\},\{DINAmp\},\{DINTend, DIN-\}$$

$$\{DIN0\},\{COMEq, COM-\},\{COMAmp\},\{COMTend, COM0\},\{COM+\}$$

$$\{REAEq, REA-\},\{REAAmp\},\{REATend, REA+\},\{REA0\},\{APPEq, APP0\}$$

$$\{APPAmp\},\{APPTEnd, APP-\},\{APP+\},\{POUEq, POU-\},\{POUAmp\}$$

$$\{POUTEnd, POU+\},\{POU0\},\{EXTVal, EXTBipo-\},\{EXTBipo0\}$$

$$\{EXTBipo+)\},\{INTVal, INTBipo+)\},\{INTBipo-)\},\{INTBipo0\},$$

$$\{ANXVal, ANXBipo-\},\{ANXBipo0\},\{ANXBipo+\},\{DETVal, DETBipo0\}$$

$$\{DETBipo-\},\{DETBipo+\},\{AFFVal, AFFBipo-\},\{AFFBipo0\},\{AFFBipo+\}$$

$$\{RMCVal, RMCBipo-\},\{RMCBipo0\},\{RMCBipo+\},\{RECVal, RECBipo0\}$$

$$\{RECBipo-\},\{RECBipo+\},\{DTNVal, DTNBipo0\},\{DTNBipo-\},\{DTNBipo+\}$$

$$\{RIGVal, RIGBipo+\},\{RIGBipo-\},\{RIGBipo0\},\{IMPVal, IMPBipo0\},\{ImpBipo-\}$$

$$\{IMPBipo+\},\{DINVal, DINBipo0\},\{DINBipo-\},\{DINBipo+\},\{CINVal, CINBipo0\}$$

$$\{CINBipo-\},\{CINBipo+\},\{COMVal, COMBipo-\},\{COMBipo0\},\{COMBipo+\}$$

$$\{CCLVal, CCLBipo-\},\{CCLBipo0\},\{CCLBipo+\},\{REAVal, REABipo0\},\{REABipo-\}$$

$$\{REABipo+\},\{FACVal, FACBipo+\},\{FACBipo-\},\{FACBipo0\},\{APPVal, APPBipo-\}$$

$$\{APPBipo0\},\{APPBipo+\},\{INDVal, INDBipo0\},\{INDBipo-\},\{INDBipo+\}$$

$$\{POUVal, POUBipo0\},\{POUBipo-\},\{POUBipo+\},\{PROVal, PROBipo+\}$$

$$\{PROBipo-\},\{PROBipo0\}$$

Quel que soit le parcours choisi, les meilleurs sous-ensembles retournés sont :

$$\{H/F, RIGTend\},\{H/F, Locale, RIGTend\},\{H/F, Locale, Activitechoisie, RIGTend\}$$

$\{H/F, Locale, Activitechoisie, Itemschoisis, RIGTend\}$

$\{H/F, Locale, Activitechoisie, Itemschoisis, Age, RIGTend\}$

$\{H/F, Locale, Activitechoisie, Itemschoisis, Age, Dipome, RIGTend\}$

$\{H/F, Locale, Activitechoisie, Itemschoisis, Age, Diplome, Classeur, RIGTend\}$

$\{H/F, Locale, Activitechoisie, Itemschoisis, Age, Diplome, Classeur, Concevoir, COMTend\}$

$\{H/F, Locale, Activitechoisie, Itemschoisis, Age, Diplome,$

$Classeur, Concevoir, Gerer, COMTend\}$

$\{H/F, Locale, Activitechoisie, Itemschoisis, Age, Diplome, Classeur$

$, Concevoir, Gerer, Argumenter, COMTend\}$

$\{H/F, Locale, Activitechoisie, Itemschoisis, Age, Diplome, Classeur,$

$Concevoir, Gerer, Argumenter, Encadrer, COMTend\}$

$\{H/F, Locale, Activitechoisie, Itemschoisis, Age, Diplome, Classeur,$

$Concevoir, Gerer, Argumenter, Encadrer, Echanger, COMTend\}$

Sur ce jeu de données, les résultats sont plus difficilement interprétables mais nous remarquons quand même quelques points intéressants. La tendance à la RIGueur est une variable qui ressort pour expliquer l'activité Administrer. C'est le cas aussi de la tendance à la COMbativité. De plus, nous notons que l'activité Administrer peut s'expliquer par les différentes capacités de l'individu pour les autres activités du modèle.

Nous retrouvons une remarque déjà faite. Les classes étant de très petite cardinalité, le parcours au hasard donne les mêmes résultats que les autres parcours.

7.2 Validation des résultats obtenus

Pour cette partie, nous nous basons sur les expérimentations détaillées dans les travaux de Kohavi and John (1997). Nous proposons d'utiliser un algorithme de construction d'arbre de décision comme outil de validation.

7.2.1 Protocole expérimental

Nous utilisons pour ce test les données à 35 variables (Section 5.3.4) et les données Soybean (Section 5.2.1). Pour chacun des deux problèmes, l'algorithme ClassAdd renvoie q sous-ensembles de variables potentiellement explicatives de cardinal 1 à q. La procédure de génération choisie est la génération exhaustive. Pour chaque sous-ensemble de cardinal i renvoyé, nous construisons un arbre de décision à l'aide de l'algorithme CART (Breiman et al., 1984) en utilisant un ensemble d'apprentissage contenant 70 % des individus pris aléatoirement (distribution uniforme sur l'ensemble des individus). Nous définissons ensuite l'erreur d'apprentissage par le nombre d'individus mal classés

et l'erreur de test par le nombre d'individus (parmi les 30 % restants) dont la classe a été mal prédite. Ces deux indicateurs permettent d'évaluer la qualité de l'arbre construit avec un nombre i restreint de variables explicatives et par conséquent la pertinence du sous-ensemble de variables en question. Afin d'obtenir des résultats plus « robustes », pour chaque sous-ensemble de taille i, nous générons 500 échantillons d'apprentissage comme indiqué ci-dessus et appliquons l'algorithme CART. Le critère de qualité retenu est le nombre moyen d'individus mal classés et le nombre moyen d'individus mal prédits sur ces 500 répétitions.

Afin de comparer l'algorithme ClassAdd avec des approches filtres existantes, nous effectuons ces mêmes tests sur des sous-ensembles obtenus avec l'approche additive de Lewis (1992) implantée dans le logiciel *Weka* (Witten and E.Frank, 2005), qui calcule l'information mutuelle entre chaque variable potentiellement discriminante et la classe à prédire, et ne retient que les variables les plus porteuses d'information.

Enfin, nous comparons aussi les résultats de ClassAdd avec ceux obtenus pour des sous-ensembles de variables explicatives générés aléatoirement (distribution uniforme sur l'ensemble des variables). Pour chaque i, $i \in \{1, ..., q\}$, nous générons 500 sous-ensembles de variables de cardinal i et appliquons l'algorithme CART comme indiqué précédemment.

7.2.2 Résultats expérimentaux

Problème artificiel

La partition retenue est celle en 12 classes :

$$\{X_1, X_6, X_{11}, X_{23}, X_{25}, X_{26}, X_{27}, X_{30}, X_{31}, X_{32}\},$$

$$\{X_2, X_7, X_{12}, X_{24}\}, \{X_3, X_8, X_{13}\},$$

$$\{X_4, X_9, X_{14}\}, \{X_5, X_{10}, X_{15}\},$$

$$\{X_{16}, X_{17}, X_{18}, X_{19}, X_{20}, X_{33}\},$$

$$\{X_{21}\}, \{X_{22}\}, \{X_{28}\}, \{X_{29}\}, \{X_{34}\}, \{X_{35}\}$$

Les Figures 7.12 et 7.13 présentent les résultats de l'algorithme CART appliqué aux sous-ensembles renvoyés par ClassAdd, aux sous-ensembles générés aléatoirement et à ceux obtenus avec un filtre additif pour le jeu de données artificielles. Le fait que les données contiennent un grand nombre de variables redondantes explique que l'algorithme ClassAdd retourne des ensembles ayant un meilleur pouvoir explicatif de la classe que l'algorithme additif classique. Enfin, les sous-ensembles retournés par ClassAdd sont nettement meilleurs que les sous-ensembles générés aléatoirement. Ce résultat conforte l'heuristique de la classification consistant à tenir compte de la structure de l'ensemble des variables potentiellement discriminantes.

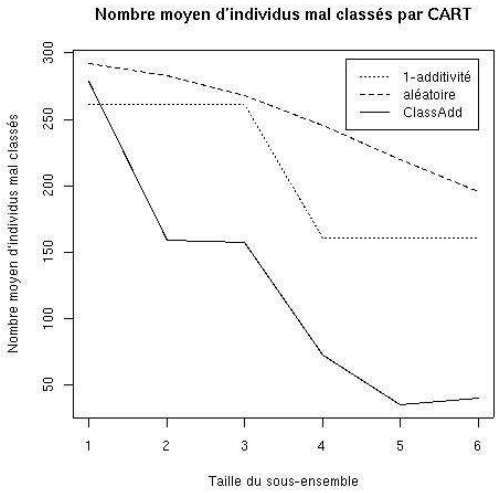


FIG. 7.12 – *Nombre moyen d'individus mal classés (ensemble d'apprentissage) avec les données artificielles.*

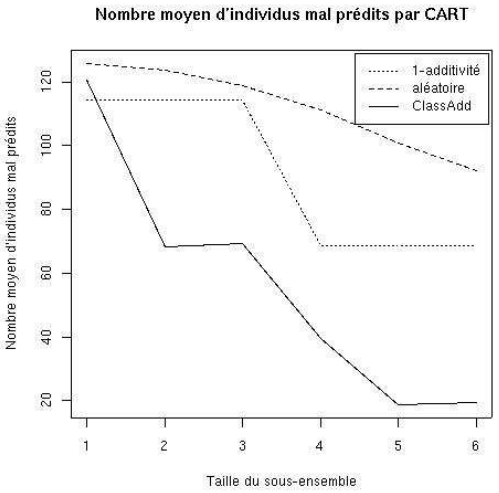


FIG. 7.13 – *Nombre moyen d'individus mal prédits (ensemble de test) avec les données artificielles.*

Les données Soybean

La partition retenue est celle en 20 classes :

$$\{X_1\}, \{X_2\}, \{X_3, X_{26}, X_{27}\}, \{X_4\}, \{X_5\},$$
$$\{X_6\}, \{X_7\}, \{X_8, X_{25}\}, \{X_9\}, \{X_{10}\},$$
$$\{X_{11}, X_{19}, X_{21}, X_{22}, X_{28}, X_{29}\},$$
$$\{X_{12}, X_{13}, X_{14}, X_{15}\}, \{X_{16}\}, \{X_{17}\}, \{X_{18}\},$$
$$\{X_{20}\}, \{X_{23}\}, \{X_{24}\}, \{X_{30}, X_{31}, X_{32}, X_{33}, X_{34}\}, \{X_{35}\}$$

Les Figures 7.14 et 7.15 présentent les résultats de l'algorithme CART appliqué sur les sous-ensembles obtenus par ClassAdd, les sous-ensembles générés aléatoirement et ceux obtenus avec un filtre additif pour le jeu de données Soybean. Nous retrouvons le même type de résultats que pour les données artificielles.

Conclusion

En première partie de cette section, nous avons comparé les trois modes de génération que nous proposons : génération exhaustive, génération à partir des deux meilleures variables par classe et génération à partir de deux variables au hasard par classe. Seule la génération à partir du hasard propose des ensembles un peu moins bons dans certains

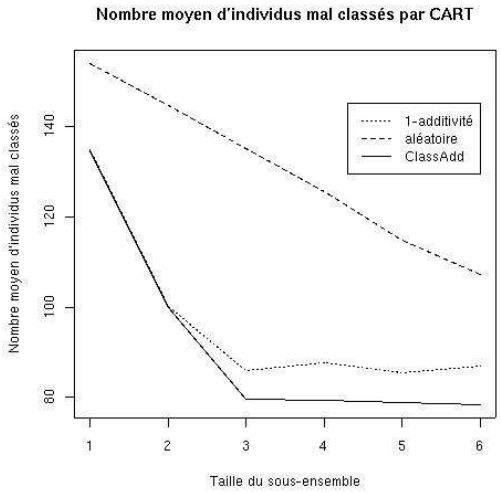


FIG. 7.14 – *Nombre moyen d'individus mal classés (ensemble d'apprentissage) avec les données Soybean.*

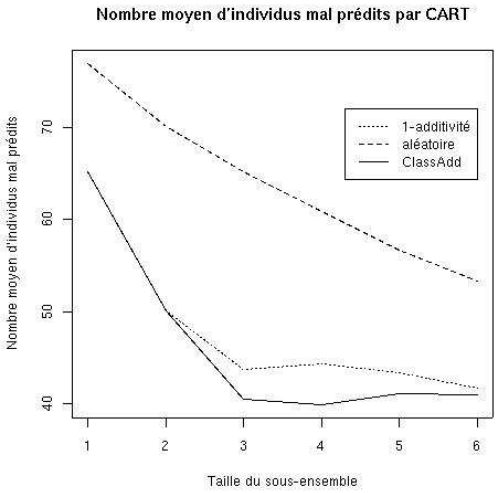


FIG. 7.15 – *Nombre moyen d'individus mal prédits (ensemble test) avec les données Soybean.*

cas, comme nous pouvions nous y attendre. La diminution d'une classe par un nombre fixé de variables représentantes semble en tout cas être une piste intéressante.

En seconde partie, nous avons testé l'algorithme complet et étudié la qualité des sous-ensembles retournés. Pour cela, nous avons utilisé l'algorithme de construction d'arbre de décisions C4.5 afin de voir si notre sous-ensemble permettait de construire un arbre de décisions avec peu d'erreurs. Notre critère de validation est d'être plus performant que l'approche additive classique, ce qui s'est avéré être le cas.

Conclusion et perspectives

Dans le cadre de cette thèse, nous souhaitions proposer un nouvel algorithme de sélection de variables dans un cadre discret. N'ayant aucune contrainte *a priori* sur le modèle de nos données, nous avons proposé une approche filtre avec une mesure de pertinence basée sur l'information mutuelle. Afin de limiter les coûts en terme de temps de calcul, nous avons proposé de l'approximer avec une troncature 2-additive. Pour réduire l'espace de recherche des sous-ensembles candidats, nous avons structuré l'ensemble des variables par une Classification Ascendante Hiérarchique. Cette classification nous a permis d'opter pour une heuristique qui consiste à ne garder qu'au plus une variable par classe dans le sous-ensemble candidat. Cette première réduction n'étant pas suffisante dès que le nombre de variables croît significativement, nous avons décidé d'élire des variables représentantes de chaque classe et de ne garder que celles-ci dans les sous-ensembles candidats. Nous avons ainsi proposé plusieurs choix des variables représentantes de la classe.

Apports principaux de notre travail

Une mesure de pertinence peu coûteuse

La mesure de pertinence estimée avec une approche 2-additive permet de prendre en compte la redondance entre variables. Ce point est très important puisque de nombreux algorithmes de sélection de variables se contentent d'une approche pas à pas en choisissant à chaque tour la meilleure variable sans tenir compte des variables déjà choisies. De plus, pouvoir estimer la pertinence d'un sous-ensemble à partir de la pertinence de ses singletons et de ses couples est très prometteur pour le coût du calcul. En effet, une fois que l'on a calculé la pertinence de toutes les variables et tous les couples de variables d'un jeu de données, la pertinence de n'importe quel sous-ensemble de variables se calcule avec une simple somme.

Un espace de recherche réduit

Face à la taille des espaces de recherche toujours plus grands, la Classification Ascendante Hiérarchique permet de structurer l'ensemble des variables et donc d'envisager des réductions d'espace significatives. L'heuristique qui consiste à ne garder qu'une seule variable est la première étape. La deuxième étape qui consiste à élire des variables représentantes de classes diminue encore la taille de l'espace. Nous nous plaçons en quelque sorte dans le cadre d'une pré-sélection de variables en entrée de notre algorithme de sélection de variables ClassAdd. Il n'en est que plus performant au point de vue temps d'exécution.

Résultats expérimentaux

L'information mutuelle comme mesure de pertinence

- La troncature 2-additive détecte bien les redondances non prises en compte par une approximation classique additive. Par construction de la troncature, ce résultat était prévisible. Nos tests sur les jeux de données le confirment expérimentalement.
- La troncature 2-additive détériore peu l'estimation de la pertinence d'un sous-ensemble. Si elle est évidemment moins précise qu'une estimation classique, le rapport entre temps de calcul et qualité de l'estimation lui est favorable.
- La troncature 2-additive semble résistante au bruit sur les données et à la taille de celles-ci.

Les indices de qualité d'une partition pour l'aide à la décision de la partition à choisir

- Les différents indices de qualité d'une partition que nous avons retenus pour ces tests (hauteur, diamètre, distance au centre, Calinski, Hubert) confirment les résultats attendus : ils permettent de déterminer automatiquement le nombre de classes, proche de la partition optimale.
- Les indices ne sont pas tous très résistants au bruit ou à la taille des données. Cette analyse sera à poursuivre pour améliorer le critère de choix.

Les procédures de génération des sous-ensemble candidats

- La procédure de génération qui consiste à ne garder que deux variables représentantes par classe fournit des résultats équivalents à une génération exhaustive des sous-ensembles candidats. Ces deux algorithmes de génération sont utilisés en appliquant l'heuristique qui consiste à ne garder qu'une seule variable par classe.
- La sélection par le hasard de deux variables représentantes a une qualité très liée à la structure de la classification. Des classes de petites tailles la rendent aussi performante que la génération exhaustive alors que des classes de grande cardinalité peuvent la rendre très mauvaise.

La qualité des sous-ensembles retournés par ClassAdd évaluée par un arbre de décision

– L'approche 2-additive est meilleure que l'approche additive classique.

Perspectives

Les perspectives qui apparaissent à l'issue de cette recherche sont multiples. Sur le plan algorithmique, le passage à l'échelle nécessite une automatisation complète de l'algorithme ClassAdd. Sur le plan théorique, nous envisageons une extension de la démarche à d'autres types de variables. Sur le plan logiciel, la réalisation d'un package dans le logiciel R permettrait sa diffusion et par là même des comparaisons plus extensives. Enfin, par l'utilisation de cet algorithme dans un cadre d'aide à la décision, une présentation des résultats adaptés à un expert en ressources humaines doit être réalisée.

Notre algorithme fonctionne sur des variables discrètes. Une extension naturelle est le passage aux variables à valeurs numériques. Les travaux sur l'information mutuelle que nous avons présentés ont été étendus au cas réel, ce qui rend cette perspective assez facilement réalisable.

La meilleure validation expérimentale d'un algorithme est son utilisation intensive sur des jeux de données très variés. Pour cela, un travail de réalisation est nécessaire afin de créer un package dans le logiciel R. Ce package serait libre d'accès pour la communauté R.

Enfin, l'algorithme actuel possède plusieurs points de paramétrage. Nous pouvons choisir d'estimer l'information mutuelle classiquement ou avec l'approche bayésienne. Nous avons le choix entre plusieurs critères de qualité pour construire notre partition de l'ensemble des variables. Ensuite, nous devons aussi décider quelle procédure de génération des sous-ensembles nous allons exécuter et le nombre de variables à garder par classe le cas échéant. Une perspective intéressante est donc de guider l'utilisateur dans ces multiples choix. Concernant la procédure de génération, nous avons remarqué que si les classes sont de petite taille, le hasard pour choisir les variables représentantes fonctionne très bien et à coût réduit. Pour les critères de qualité, nous avons regroupé des comportements communs mais sans avoir fait de lien avec la structure des données. Cette perspective d'aide à la décision pour l'utilisateur de l'algorithme doit s'effectuer en même temps que l'étude d'un algorithme qui se relancerait plusieurs fois de suite avec des initialisations différentes. Le coût est forcément plus élevé mais le résultat est certainement plus riche.

En complément, un travail de présentation des résultats est nécessaire pour faciliter leur interprétation. En effet, pour l'instant, l'algorithme ClassAdd fournit une liste de sous-ensembles, des matrices de similarité entre variables, des classes de variables, ce qui peut représenter une quantité d'information importante difficilement exploitable. Nous pouvons envisager une représentation via une interface graphique qui offrirait une

vue globale des variables, et une vue détaillée grâce à la navigation (un clic sur un ensemble pourrait afficher la pertinence de celui-ci, son classement dans l'ensemble des sous-ensembles, etc). Ce besoin est très fort pour des utilisateurs non statisticiens tels que les experts psychologues de l'entreprise PerformanSe.

Annexes

A

Propriétés de convexité et de convergence

Définition A.0.1 (Convexité d'une fonction) *Une fonction f, définie sur $[a,b]$, $a,b \in \mathbb{R}$, est dite convexe sur l'intervalle $[a,b]$ si pour tout couple $(x_1,x_2) \in [a,b] \times [a,b]$ et pour tout $0 \leq \lambda \leq 1$,*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

La fonction f est strictement convexe s'il y a égalité seulement quand $\lambda = 0$ ou $\lambda = 1$.

Définition A.0.2 (Concavité d'une fonction) *Une fonction f, définie sur $[a,b]$, $a,b \in \mathbb{R}$, est dite concave sur l'intervalle $[a,b]$ si $-f$ est convexe.*

Théorème A.0.1 *Si f, définie sur $[a,b]$, $a,b \in \mathbb{R}$, admet une dérivée seconde non-négative sur l'intervalle $[a,b]$, alors f est une fonction convexe.*

Théorème A.0.2 (Inégalité de Jensen) *Soit f une fonction convexe et X, une variable aléatoire alors*

$$E[f(X)] \geq f(E[X])$$

De plus, si f est strictement convexe alors $E[f(X)] = f(E[X])$, ce qui implique que $X = E[X]$ avec une probabilité 1, X est une constante.

Définition A.0.3 (La convergence en loi) *Soit $(F_n)_{n\in\mathbb{N}}$ la suite des fonctions de répartition associées aux variables aléatoires $(X_n)_{n\in\mathbb{N}}$, et F la fonction de répartition associée à la variable aléatoire X. La suite $(X_n)_{n\in\mathbb{N}}$ converge en loi vers X si*

$$\lim_{n\to\infty} F_n(a) = F(a)$$

pour tout réel a pour lequel F est continue.

Définition A.0.4 (La convergence en probabilité) *La suite de variables aléatoires* $(X_n)_{n\in\mathbb{N}}$ *converge en probabilité vers la variable aléatoire* X *si et seulement si pour tout* $\epsilon > 0$

$$\lim_{n\to\infty} P\left(|X_n - X| \geq \varepsilon\right) = 0$$

La convergence en probabilité implique la convergence en loi.

Théorème A.0.3 (Théorème de Slutsky de base) *Soient* $(X_n)_{n\in\mathbb{N}}$ *et* $(Y_n)_{n\in\mathbb{N}}$*, deux suites de variables aléatoires. Si* $(X_n)_{n\in\mathbb{N}}$ *converge en loi vers* X *et* $(Y_n)_{n\in\mathbb{N}}$ *converge en probabilité vers une constante* c*, alors* $(X_n + Y_n)_{n\in\mathbb{N}}$ *converge en loi vers* $X + c$*,* $(X_nY_n)_{n\in\mathbb{N}}$ *converge en loi vers* cX *et* $(X_n/Y_n)_{n\in\mathbb{N}}$ *converge en loi vers* X/c *si* c *est non nulle.*

Théorème A.0.4 (Théorème de Slutsky généralisé) *Soient* $(X_n)_{n\in\mathbb{N}}$ *et* $(Y_n)_{n\in\mathbb{N}}$*, deux suites de variables aléatoires. Si* $(X_n)_{n\in\mathbb{N}}$ *converge en loi vers* X *et* $(Y_n)_{n\in\mathbb{N}}$ *converge en probabilité vers une constante* c*, alors la suite* $g(X_n, Y_n)_{n\in\mathbb{N}}$ *converge en loi vers* $g(X, c)$ *pour toute fonction* g *continue définie de* $\mathbb{R}^2$ *dans* $\mathbb{R}$*.*

Théorème A.0.5 (Théorème central-limite) *Soit* $(X_n)_{n\in\mathbb{N}}$ *une suite de variables aléatoires de même loi d'espérance* μ *et d'écart-type* σ*. Alors la variable aléatoire* $\frac{1}{\sqrt{n}}\left(\frac{X_1+\cdots+X_n-n\mu}{\sigma}\right)$ *converge en loi vers une loi normale centrée réduite.*

B

Compléments pour l'inférence bayésienne

B.1 La distribution de Dirichlet

La loi de Dirichlet est une distribution de probabilités ayant pour fonction de densité, la fonction f définie par :

$$f(x;\alpha) \sim \prod_{i=1}^{K} x_i^{\alpha_i - 1} \delta \left(1 - \sum_{i=1}^{K} x_i\right)$$

où x est un vecteur de dimension K, $x = (x_1, \ldots, x_k)$ avec $x_i \geq 0 \forall i$, et $\alpha = (\alpha_1, \ldots, \alpha_K)$ est un vecteur de paramètre avec $\alpha_i \geq 0 for all i$. La fonction δ est le delta de Dirac. La principale relation de la distribution de Dirichlet est la suivante :

$$M|\beta \sim Dirichlet(\alpha + \beta)$$

Cette relation est utilisée en statistique bayésienne pour estimer des paramètres cachés, M, d'une distribution discrète de probabilités sur un échantillon de taille n. Intuitivement, si la probabilité *a priori* est représenté par $Dirichlet(\alpha)$ alors $Dirichlet(\alpha + \beta)$ est la probabilité *a posteriori* étant donné des observations sur un histogramme β.

B.2 La loi Beta

B.2.1 La fonction beta

La fonction gamma est une extension de la fonction factorielle aux nombres réels ou complexes. Elle est définie par :

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$$

pour tout $z \in \mathbb{C}$ avec $Re(z) > 0$.

La fontion beta, B, est une fonction définie par :

$$B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt$$

pour tout x et $y \in \mathbb{C}$ et $Re(x) > 0, Re(y) > 0$. La fonction beta est symétrique et liée à la fonction gamma, Γ, par :

$$B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

De la même façon que la fonction gamma sur les entiers correspond à la fonction factorielle, la fonction beta peut définir un coefficient binômial :

$$\binom{n}{k} = \frac{1}{(n+1)B(n-k+1, k+1)}$$

B.2.2 la loi beta

La loi beta est une distribution de probabilités ayant pour fonction de densité, la fonction f, définie sur $[0, 1]$ par :

$$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

avec α et β deux paramètres positifs et B, la fonction beta (section B.2.1). La loi beta est étroitement liée à la loi binômiale. En effet, si i et j sont deux entiers, $B(i, j)$ est alors la distribution de la j-ème plus grande valeur d'un échantillon de $i+j-1$ variables aléatoires uniformément distribuées. La probabilité cumulée entre 0 et x est donc la probabilité que la j-ème plus grande valeur soit plus petite que x. En d'autres termes, il s'agit de la probabilité qu'au moins i des variables aléatoires soient plus petites que x. Cette probabilité est obtenue en sommant selon une distribution binômiale. Un cas particulier de la loi beta est la distribution uniforme pour $\alpha = \beta = 1$. Soit X une variable aléatoire suivant une loi beta ayant pour paramètre α et β, alors son espérance et sa variance valent :

$$E(X) = \frac{\alpha}{\alpha + \beta}$$

et

$$var(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

La loi beta est très utilisée en statistique bayésienne puisqu'elle fournit une famille de distributions conjuguées *a priori* pour les distributions binômiales.

C

Le logiciel R (R Development Core Team, 2005)

Nous avons mené nos expérimentations avec le logiciel R que nous allons présenter maintenant. R a été initialement créé par Robert Gentleman et Ross Ihaka du département de statistique de l'Université d'Auckland en Nouvelle Zélande. Depuis 1997, c'est une équipe (la «R Core Team ») qui développe R. C'est à la fois un langage et logiciel qui permet de réaliser des analyses statistiques. R possède :

- un système efficace de manipulation et de stockage des données,
- différents opérateurs pour le calcul sur tableaux,
- un grand nombre d'outils pour l'analyse des données et les méthodes statistiques,
- des moyens graphiques pour visualiser les analyses,
- un langage de programmation simple et performant comportant : conditions, boucles, moyens d'entrées sorties, possibilité de définir des fonctions récursives.

Le langage R s'est construit à partir de deux autres langages :

- S qui est un langage développé par les AT&T Bell Laboratories. S est à la fois un langage haut niveau et un environnement pour l'analyse des données et les représentations graphiques. Il est utilisable à travers le logiciel SPlus qui est commercialisé par la société Insightful. C'est l'un des logiciels de statistiques les plus populaires et il s'est imposé comme une référence dans le milieu statistique.
- Scheme de Sussman qui est un langage fonctionnel dont le principe fondamental est la récursivité. L'exécution et la sémantique de R en sont dérivées. Le noyau de R est écrit en langage machine interprété qui a une syntaxe simillaire au langage C, mais qui est réellement un langage de programmation avec des capacités identiques au langage Scheme.

La plupart des fonctions accessibles par l'utilisateur dans R sont écrites en R. Pour les tâches plus coûteuses les langages C, C++ et Fortran ont été utilisés et liés pour une meilleure efficacité. L'utilisateur peut créer de nouvelles fonctions en R ou en C pour

manipuler directement des objets R. Toutes les grandes méthodes statistiques classiques sont implémentées dans le logiciel R : les modèles linéaires, les modèles linéaires généralisés, la régression nonlinéaire, les séries chronologiques, les tests paramétriques et non paramétriques classiques, ...R peut facilement étendre ses fonctions par l'intermédiaire de librairies. Les modules de base sont fournis avec la distribution de R mais d'autres sont disponibles par l'intermédiaire du CRAN (Comprehensive R Archive Network). Le CRAN est un ensemble de sites qui fournissent ce qui est nécessaire à la distribution de R, ses extensions, sa documentation, ses fichiers sources et ses fichiers binaires. Les extensions ont été créées pour des buts spécifiques et proposent une large gamme de statistiques modernes : analyse descriptive des données multidimensionnelles, arbres de régression et de classification, graphiques en trois dimensions, ...

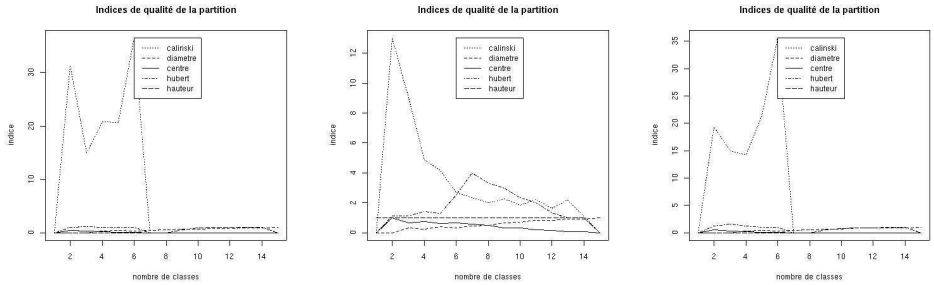
Enfin, R est développé pour pouvoir être utilisé avec les systèmes d'exploitation Unix, GNU/Linux, Windows et MacOS. C'est un logiciel libre qui est distribué sous les termes de la GNU Public Licence et il fait partie intégrante du projet GNU. Le logiciel étant donc dans le domaine public, son point fort est le développement d'applications, de modules qui sont mis à la disposition de tous les utilisateurs et développeurs qui peuvent proposer des extensions. R est donc en perpétuelle évolution et son potentiel est donc très grand.

D

Autres expérimentations effectuées

D.1 Comparaison des indices de qualité d'une partition

Ce chapitre complète la section 6.3 en présentant l'évolution des écarts pour les indices de qualité.

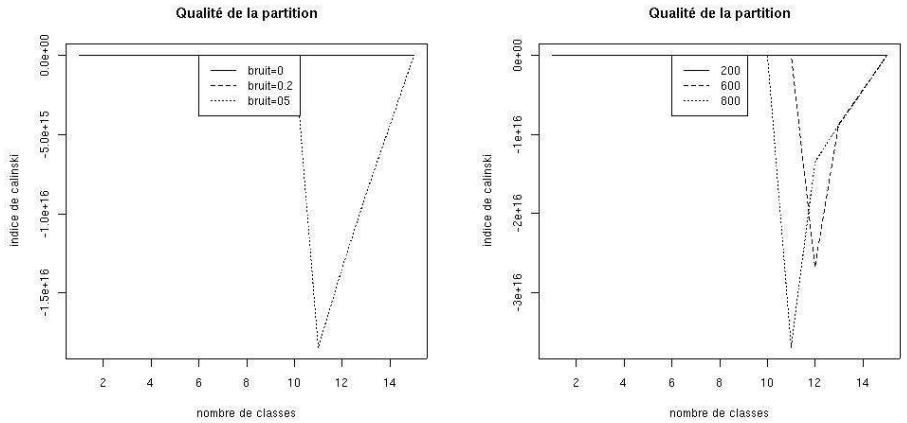


(a) Estimation classique. (b) Approche à la Lerman. (c) Approche probabiliste.

FIG. D.1 – *Evolution des indices de qualité d'une partition sur des données de 1600 individus.*

D.2 Résistance au bruit et à la taille des données de l'indice de qualité

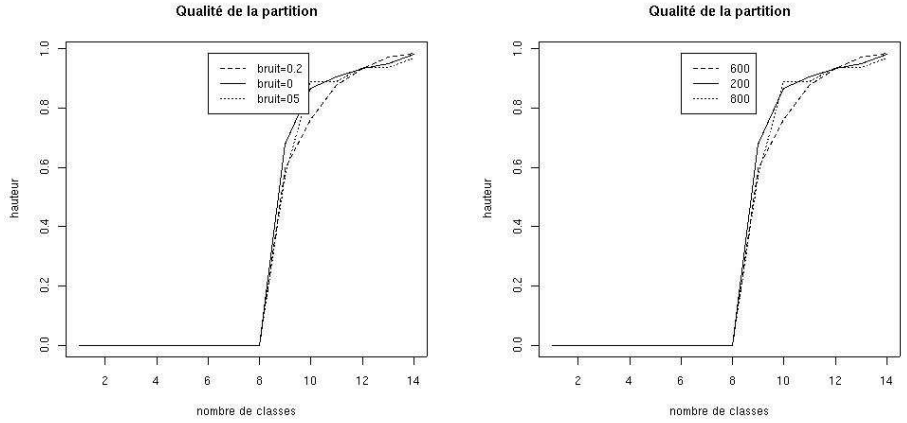
D.2.1 Indice de Calinski



(a) Résistance au bruit. (b) Résistance à la taille.

FIG. D.2 – *Robustesse de l'indice de Calinski pour une classification basée sur une information mutuelle estimée classiquement.*

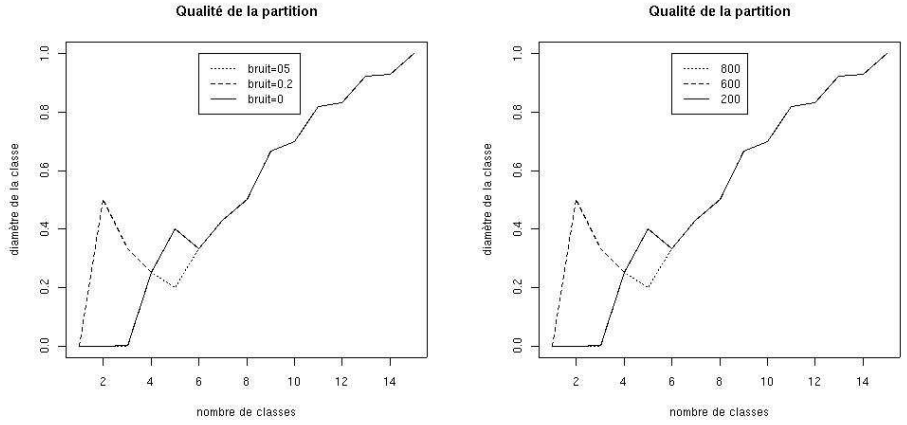
D.2.2 Indice de la hauteur



(a) Résistance au bruit. (b) Résistance à la taille.

FIG. D.3 – *Robustesse de l'indice de la hauteur pour une classification basée sur une information mutuelle estimée avec une approche bayésienne.*

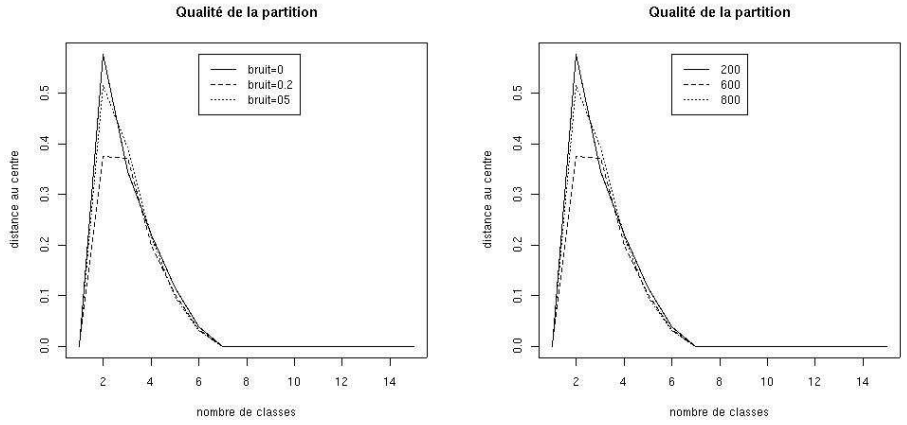
D.2.3 Indice du diamètre moyen



(a) Résistance au bruit. (b) Résistance à la taille.

FIG. D.4 – *Robustesse de l'indice du diamètre moyen pour une classification basée sur une information mutuelle estimée avec une approche bayésienne.*

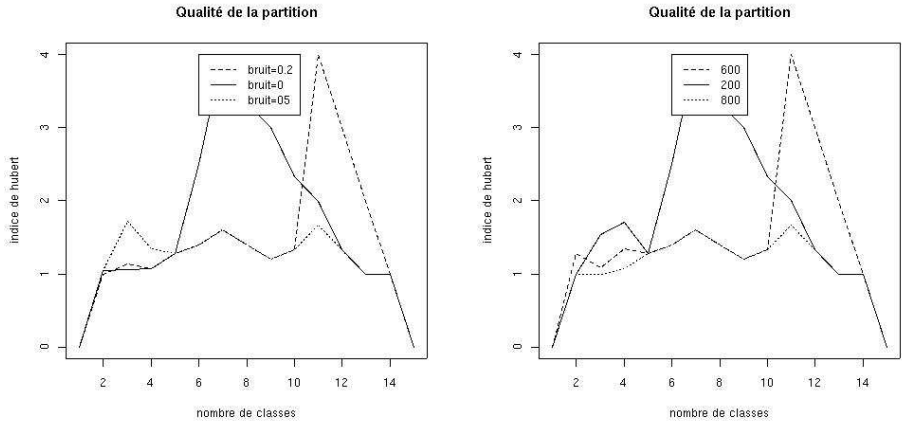
D.2.4 Indice de la distance au centre de la classe



(a) Résistance au bruit. (b) Résistance à la taille.

FIG. D.5 – *Robustesse de l'indice de la distance au centre de la classe pour une classification basée sur une information mutuelle estimée avec une approche bayésienne.*

D.2.5 Indice de Hubert



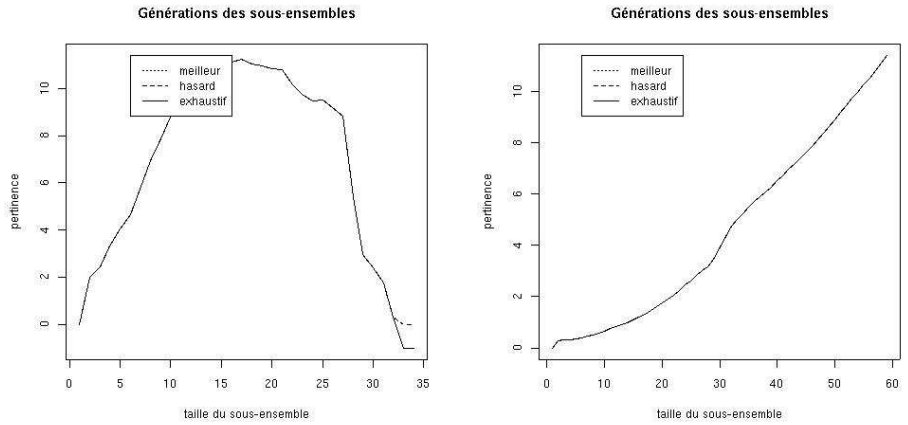
(a) Résistance au bruit.

(b) Résistance à la taille.

FIG. D.6 – *Robustesse de l'indice de Hubert pour une classification basée sur une information mutuelle estimée avec une approche probabiliste.*

D.3 Comparaison des trois algorithmes de génération de sous-ensembles

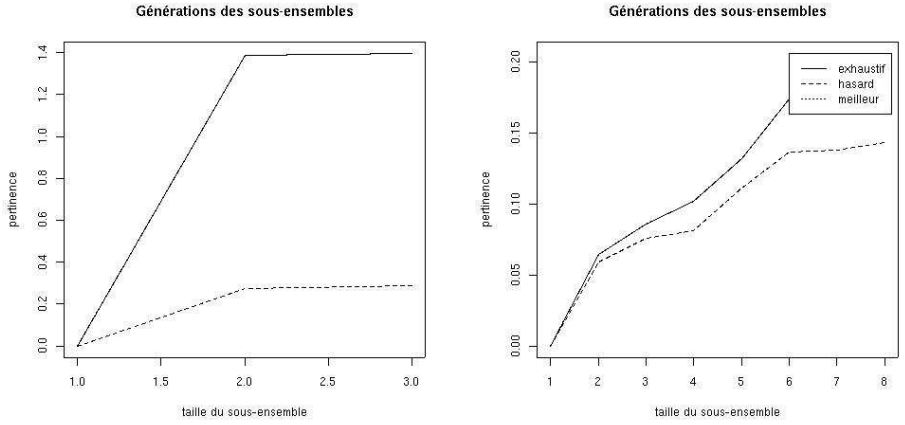
D.3.1 Indice de la hauteur



(a) Données Soybean. (b) Données Splice.

FIG. D.7 – *Pertinence des sous-ensembles des données Soybean et Splice retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de la hauteur.*

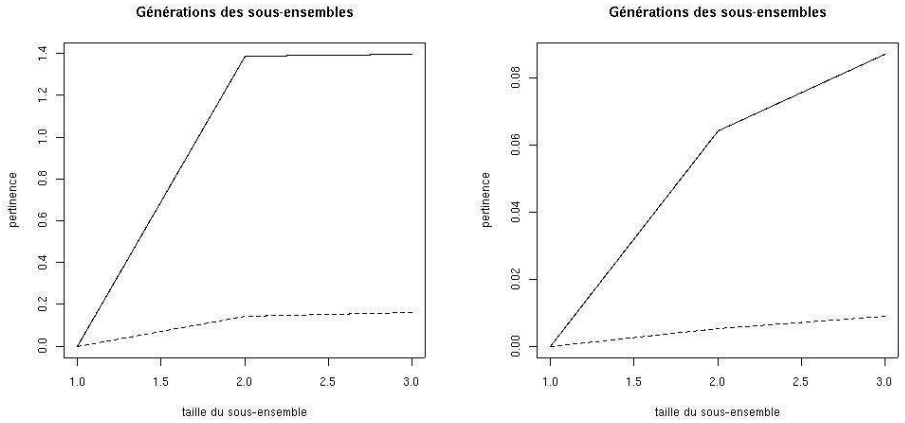
D.3.2 Indice du diamètre moyen



(a) Audiology. (b) Connect

FIG. D.8 – *Pertinence des sous-ensembles des données Audiology et Connect retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère du diamètre moyen.*

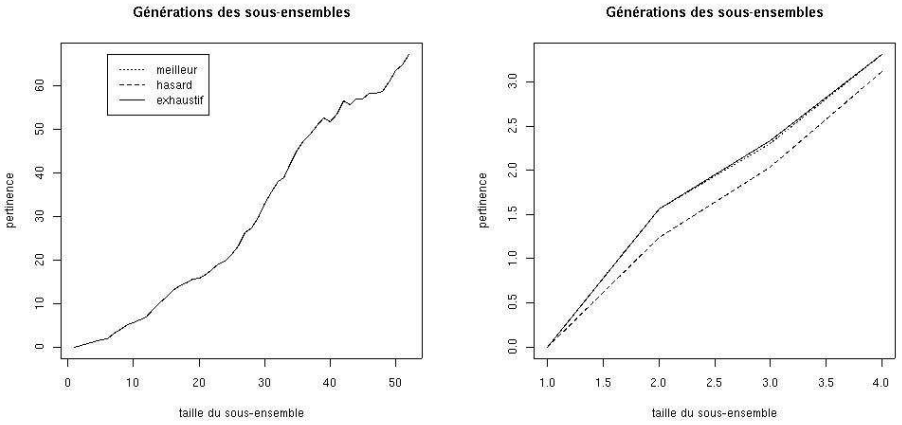
D.3.3 Indice de la distance au centre



(a) Audiology. (b) Connect.

FIG. D.9 – *Pertinence des sous-ensembles des données Audiology et Connect retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de la distance au centre de la classe.*

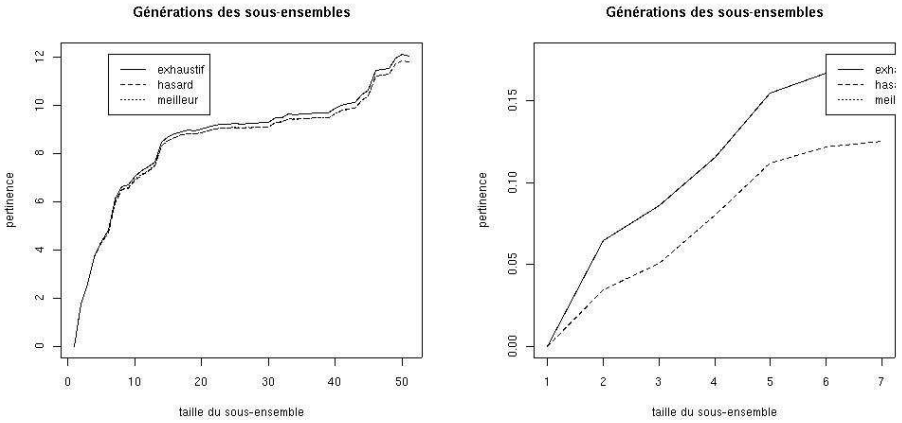
D.3.4 Indice de Calinski



(a) Lung Cancer. (b) Optdigit.

FIG. D.10 – *Pertinence des sous-ensembles des données Lung Cancer et Optdigit retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de Calinski.*

D.3.5 Indice de Hubert



(a) Audiology. (b) Connect.

FIG. D.11 – *Pertinence des sous-ensembles des données Audiology et Connect retournés par les trois parcours étudiés de l'espace de recherche : exhaustif, hasard, meilleures variables. La partition est choisie par le critère de Hubert.*

Bibliographie

N. Abramson. *Information Theory and Coding.* McGraw Hill, New-York, 1963.

A. Agresti. *Categorical Data Analysis.* Wiley, 2002. Second edition.

H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, volume 2, pages 547–552, Anaheim, California, 1991. AAAI Press.

H. Almuallim and T. G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2) :279–305, 1994.

R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transaction on Neural Networks*, 5(4) :537–550, 1994.

Y. Bennani and S. Guérif. Sélection de variables en apprentissage numérique non supervisé. In *In Cap'07 : conférence francophone sur l'apprentissage automatique*, 2007.

Y. Bennani, S. Guérif, and E. Janvier. mu-som : Weighting features during clustering. In *In International Workshop On Self-Organizing Mapsy*, 2005.

M. Boddy and T. Dean. Decision-theoretic deliberation scheduling for problem solving in time-constrained environments. *Artificial Intelligence*, 67(2) :245–286, 1994.

L. Breiman, J.H. Freidman, R.A. Olshen, and C.J. Stone. *Classification and Regression Tree.* Wadsworth, 1984.

F. Brucker and J.P. Barthélemy. *Eléments de classification.* Hermès-Lavoisier, 2007.

R. B. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3 :1–27, 1974.

C. Cardie. Using decision trees to improve case–based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 25–32. Morgan Kaufmann Publishers, Inc., 1993.

M. Carlyn. An assessment of the myers-briggs type indicator. *Joumal of Personality Assessment*, 41 :461–473, 1977.

T. Cover. The best two independent measurements are not the two best. *IEEE Trans. Systems, Man and Cybernictics*, 4 :116–117, 1974.

D. Cowan. An alternative to the dichotomous interpretation of jung's psychological functions : Developing more sensitive measurement technology. *Journal of Personality Assessment*, 53(3) :459–471, 1989.

M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(3), 1997.

C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5 :1531–1555, 2004.

A. M. Fraser and H.L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2) :1134–1140, 1986.

T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10) :906–914, 2000.

J. George and G. Jones. *Organizational Behavior*. Prentic Hall, Upper Saddler River, NJ, 3rd ed. 2004 edition, 2002.

L. R. Goldberg. Language and individual differences : The search for universals in personality lexicons. *Review of Personality and Social Psychology*, 2 :141–165, 1981.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3 :1157–1182, 2003.

S. Guérif. Unsupervised feature selection : when random rankings sound as irrelevancy. In *In JMLR Workshop and Conference Proceedings, New challenges for feature selection in data mining and knowledge discovery*, volume 4, pages 161–175, 2008.

B. Hanczar, J.D. Zucker, C. Henegar, and L. Saittaos. Feature construction from synergic pairs to improve microarray-based classification. *Bioinformatics*, page 429, 2007.

P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79 :191–215, 1997.

A. Hardy. On the number of clusters. *Computational Statistics and Data Analysis*, 23 : 83–96, 1996.

R. V. L Hartley. Transmission of information. *The Bell System Technical J.*, 7 :535–563, 1928.

L. J. Hubert and J. R. Levin. A general statistical framework for assessing categorical clustering in free recall. *Psychological bulletin*, 83 :1072–1080, 1976.

M. Hutter and M. Zaffalon. Distribution of mutual information from complete and incomplete data. *Computational Statistics and Data Analysis*, 48 :633–657, 2005.

H. Joe. Relative entropy measures of multivariate dependence. *J. Am. Statist. Assoc.*, 84 :157–164, 1989a.

H. Joe. Estimation of entropy and other functionals of a multivariate density. *Ann. Inst. Statist. Math.*, 41 :683–697, 1989b.

K. Kira and L. A. Rendell. The feature selection problem : traditional methods and a new algorithm. In *Proceedings of the Ninth National Conference on Machine Learning*, pages 129–134, 1992a.

K. Kira and L. A. Rendell. A practical approach to feature selection. In *ML '92 : Proceedings of the Ninth International Workshop on Machine Learning*, pages 249–256, San Francisco, CA, USA, 1992b. Morgan Kaufmann Publishers Inc.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2) : 273–324, 1997.

I. Kojadinovic. Relevance measures for subset variable selection in regression problems based on k-additive mutual information. *Computational Statistics and Data Analysis*, 49(4) :1205–1227, 2005.

D. Koller and M. Sahami. Toward optimal feature selection. In *International Conference on Machine Learning*, pages 284–292, 1996.

I. Kononenko. Estimating attributes : Analysis and extensions of RELIEF. In *European Conference on Machine Learning*, pages 171–182, 1994.

S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Stat.*, 22 : 79–86, 1951.

G.N. Lance and W.T. Williams. A general theory of classificatory sorting strategies 1 - hierarchical systems. *Computer Journal*, 9, 1967.

P. Langley. Selection of relevant features in machine learning. pages 140–144., 1994.

P. Leray and P. Gallinari. Feature selection with neural networks. *Behaviormetrika*, 26, 1999.

I.C. Lerman. *Classification et Analyse Ordinale de Données*. Dunod, Paris, 1981.

I.C. Lerman. Conception et analyse de la forme limite d'une famille de cooefficients statistiques d'association entre variables relationnelles. *Mathématiques Informatique et Sciences Humaines*, pages 75–100, 1992.

I.C. Lerman and P. Peter. Indice probabiliste de vraisemblance du lien entre objets quelconques. analyse comparative entre deux approches. *Revue de Statistique Appliquée*, pages 5–35, 2003.

D. D. Lewis. Feature Selection and Feature Extraction for Text Categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 212–217, San Mateo, California, 1992. Morgan Kaufmann.

H. Liu and H.Motoda. *Feature Selection for Knowledge Discovery and Data Mining.* Kluwer Academic Publishers, Norwell, MA, USA, 1998.

H. Liu and H. Motoda. *Feature selection for knowledge discovery and data mining.* Kluwer Academic Publishers, 1998.

H. Liu and W. Wen. Concept learning through feature selection. In *First Australian and New Zealand Conference on Intelligent Information Systems*, pages 293–297, 1993.

G. W. Miligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50 :159–179, 1985.

P. Mitra, C.A. Murthy, and S.K. Pal. Unsupervised feature selection using feature similarity. *IEEE Trans. On Patterns Analysis and Machine Learning*, 24-4, 2002.

K. Miyahara and M. J. Pazzani. Collaborative filtering with the simple bayesian classifier. In *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*, pages 679–689, 2000.

I. Myers. The myers-briggs type indicator. *Educational Testing Service*, 1962.

P. Myers. *Gifts Differing. Understanding Personality Type.* Davies-Black Publishing, 1995.

P.M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Trans. Computers*, 26(9) :917–922, 1977.

F. C. Nicolau and H. Bacelar-Nicolau. Some trends in the classification of variables. In C. Hayashi, N. Ohsumi, K. Yajima, Y. Tanaka, H. H. Bock, and Y. Baba, editors, *Data Science, Classification and Related Methods*, pages 89–98. Springer, 1998.

Stéphanie Noci. Le test Sosie : fiabiliser l'évaluation des personnes : Les outils d'évaluation des personnes. *Actualité de la formation permanente*, 186 :17–20, 2003.

K. Pearson. Mathematical contributions to the theory of evolution, xiii : on the theory of contingency and its relation to association and normal correlation. In *Drapers' Company Research Memoires (Biomeric Series I)*. London : University College, 1904. Reprinted in Early Statistical Papers (1948) by the Cambridge University Press, Cambridge, UK.

PerformanSe. Classeur de formation PerformanSe. Technical report, 2003.

V. Philippé, S. Baquedano, R. Gras, P. Peter, J. Juhel, P. Vrignaud, and Y. Forner. étude de validation : Performanse echo, performanse oriente. Technical report, Study realized with the collaboration of PerformanSe, Laboratoire COD de l'École Polytechniquede l'Université de Nantes, Laboratoire de Psychologie Différentielle de l'Université de Rennes 2, 2004.

J.-M. Poggi and C. Tuleau. Classification supervisée en grande dimension. application à l'agrément de conduite automobile. *Revue de Statistique Appliquée*, 4 :41–60, 2006.

W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recogn. Lett.*, 15(11) :1119–1125, 1994.

J. R. Quinlan. *C4.5 : programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

R Development Core Team. *R : A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. URL `http://www.R-project.org`.

C.A. Ratanamahatana and D.Gunopulos. Feature selection for the naive bayesian classifier using decision trees. *Applied artificial intelligence*, 5-6(17) :475–487, 2003.

G-C. Rota. On the foundations of combinatorial theory. I. Theory of Möbius functions. *Z. Wahrscheinlichkeitstheorie und Verw. Gebiete*, 2 :340–368 (1964), 1964.

G. Saporta. *Probabilités, Analyse de Données et Statistique, 2ème édition*. Editions Technip, Paris, 2006.

C.E. Särndal. A comparative study of association measures. *Psychometrika*, 39 :165–187, 1974.

R. Setiono and H. Liu. Neural-network feature selector. *IEEE Transactions on Neural Networks*, 8(3) :654–662, May 1997.

C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27 :379–623, 1948.

W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2 :197–220, 1988.

R.R. Sokal and P.H.A. Sneath. *Principles of numerical taxonomy*. Freeman, San Francisco, 1963.

B. S. Stein and J. D. Bransford. Constraints on effective elaboration : effects of precision and subject generation. *Journal of Verbal Learning andVerbal Behavior*, 18 :769–777, 1979.

K. Torkkola. Feature extraction by non parametric mutual information maximization. *J. Mach. Learn. Res.*, 3 :1415–1438, 2003.

J. S. Wiggins, editor. *The Five-Factor Model of Personality : Theoretical Perspectives*. Guilford, New York, 1996.

I. H. Witten and E.Frank. *Data Mining : Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, 2005.

H. H. Yang and J. Moody. Feature selection based on joint mutual information. *Advances in Intelligent Data Analysis (AIDA), Computational Intelligence Methods and Applications (CIMA), International Computer Science Conventions*, 1999.

Résumé

Le problème de la sélection de variables en discrimination se rencontre généralement lorsque le nombre de variables, pouvant être utilisées pour expliquer la classe d'un individu, est très élevé. Les besoins ont beaucoup évolué ces dernières années avec la manipulation d'un grand nombre de variables dans des domaines tels que les données génétiques, la chimie moléculaire ou encore le traitement de documents textes.

Une procédure de sélection de variables consiste à sélectionner un sous-ensemble de variables permettant d'expliquer la classe de façon optimale ou quasi-optimale. La nécessité de ce traitement est essentiellement due au fait que, généralement, un nombre de variables discriminantes trop élevé dans un modèle de discrimination détériore grandement sa capacité de généralisation et la compréhension de la relation modélisée.

Dans le cadre de ce travail, nous nous intéressons au cas où les variables potentiellement discriminantes sont toutes discrètes ou nominales et nous proposons une **procédure de sélection de variables** indépendante d'un modèle de données. Nos travaux s'orientent dans deux directions : une **mesure de pertinence peu coûteuse** grâce à l'utilisation d'une **troncature k-additive de l'information mutuelle** et une **réduction de l'espace de recherche** en structurant l'ensemble des variables avec une **classification ascendante hiérarchique**.

Notre algorithme a pu être expérimenté sur trois types de données : des jeux artificiels construits avec une structure connue, des jeux de données réelles classiques et enfin une application d'entreprise : une population de cadres à la recherche d'emploi décrite par des variables comportementales.

Mots-clés : troncature k-additive, sélection de variables, classification de variables

---

Summary

Subset variable selection algorithms are necessary when the number of features is too huge to provide a good understanding of the underlying process that generated the data. In the past few years, variable and feature selection have become the focus of much research because of domains, such as molecular chemistry or gene expression array analysis, with hundreds to tens of thousands of variables.

In the framework of subset variable selection for supervised classification involving only discret variables, we propose a selection algorithm using a computationally efficient relevance measure based on a k-additive truncation of the mutual information and involving an agglomerative hierarchical clustering of the set of potentially discriminatory variables in order to reduce the number of subsets whose relevance is estimated.

Keyword : k-additive troncation, feature selection, variable clustering