UNIVERSITÉ PARIS 13

THÈSE

Présentée par

Sébastien GUÉRIF

pour obtenir le titre de

Docteur de l'Université Paris 13

Spécialité : Informatique

Réduction de dimension en Apprentissage Numérique Non Supervisé

Soutenue publiquement le 11 décembre 2006

devant le jury composé de

Directeur	:	Pr. Younès BENNANI,	LIPN, Université Paris 13
Rapporteurs	:	Pr. Cyrille BERTELLE	LITIS, Université du Havre
		Pr. Gilles VENTURINI	LIUT, Ecole Polytechnique de l'Université de Tours
Examinateurs	:	Pr. Pascale KUNTZ	LINA, Ecole Polytechnique de Nantes
		Pr. Magnus S. MAGNUSSON	HBL, University of Iceland
		Pr. Jean-Daniel ZUCKER	LIM&Bio, Université Paris 13
Invité	:	M. Emmanuel ECOSSE	INSERM, Paris
		M. Eric JANVIER	Numsight, Boulogne Billancourt

RÉDUCTION DE DIMENSION EN APPRENTISSAGE NUMÉRIQUE NON SUPERVISÉ

Dimension Reduction for Unsupervised Numerical Learning

Sébastien GUÉRIF

⋈

favet neptunus eunti

Université de Paris Nord

Sébastien GUÉRIF

Réduction de dimension en Apprentissage Numérique Non Supervisé

xii+116 p.

Remerciements

J’adresse toute ma reconnaissance à Younès Bennani qui m’a permis de réaliser cette thèse pour sa disponibilité, ses encouragements, ses conseils et sa confiance.

Je remercie Claude Baudoin, professeur à l’Université Paris 13, pour sa disponibilité, ses encouragements et nos échanges toujours très enrichissants.

J’adresse mes sincères remerciements à Monsieur Cyrille Bertelle, professeur à l’Université du Havre, et Monsieur Gilles Venturini, professeur à l’Université de Tours, qui ont accepté d’évaluer ce travail.

Je remercie également Pascale Kuntz, professeur à l’Université de Nantes, Magnus Magnusson, professeur à l’Université d’Islande, et Jean-Daniel Zucker, professeur à l’Université Paris 13, d’avoir accepté de participer à mon jury de thèse.

Je tiens à consacrer quelques lignes aux personnes sans qui cette aventure n’aurait vraisemblement jamais commencé : Mohamed Quafafou, qui avait accepté d’encadrer mon mémoire de maîtrise et qui m’a fait connaître l’Université Paris 13, Daniel Kayser et Henry Soldano pour leur soutien lorsque cette thèse n’était encore qu’un projet lointain.

J’adresse ma gratitude à la société Numsight qui a financé la deuxième moitié de ce travail, et mes remerciements à mes anciens collègues : Eric Janvier, Marc Kerslake, Thierry Couronne, Emmanuel Ecosse et tous les autres qui sont trop nombreux pour être tous cités.

La réalisation d’une thèse s’appuie aussi sur un environnement qui est essentiel et qui va au-delà des murs de notre laboratoire ; je tiens à remercier tous les membres de notre laboratoire, mais également Colette et Daniel du service de reprographie, Faiza pour ces encourageants quotidiens. Je consacre une mention particulière à Hakima et Anass pour leur soutien à un moment où j’en avais grand besoin, à Sophie pour sa compagnie nocturne et dominicale, à Céline et Dominique pour ces nombreuses pauses café qui sont toujours l’occasion d’échanges tant personnels que scientifiques et à Françoise, Touria et Antoine pour avoir accepté de partager leur bureau avec l’horrible bavard que je suis.

“Last but not least”, je remercie ma famille et mes amis pour leur soutien et leurs encourageants.

Trop nombreux sont ceux que je n’ai pu nommés, qu’ils trouvent ici l’expression de ma gratitude.

Résumé

La classification automatique - *clustering* - est une étape importante du processus d'extraction de connaissances à partir de données (ECD). Elle vise à découvrir la structure intrinsèque d'un ensemble d'objets en formant des regroupements - *clusters* - qui partagent des caractéristiques similaires. La complexité de cette tâche s'est fortement accrue ces deux dernières décennies lorsque les masses de données disponibles ont vu leur volume exploser. En effet, le nombre d'objets présents dans les bases de données a fortement augmenté mais également la taille de leur description. L'augmentation de la dimension des données a des conséquences non négligeables sur les traitements classiquement mis en oeuvre : outre l'augmentation naturelle des temps de traitements, les approches classiques s'avèrent parfois inadaptées en présence de bruit ou de redondance. Dans cette thèse, nous nous intéressons à la réduction de dimension dans le cadre de la classification non supervisée. Différentes approches de sélection ou de pondération de variables sont proposées pour traiter les problèmes liés à la présence d'attributs redondants ou d'attributs fortement bruités :

- Nous proposons d'abord l'algorithme μ-SOM qui limite l'effet de la présence d'attributs redondants en calculant une pondération des attributs à partir d'une classification simultanée des objets et des attributs.
- Nous présentons ensuite une approche intégrée – *embedded* – de sélection de variables pour la classification automatique qui permet de découvrir à la fois le nombre de groupes d'objets présents dans les données mais aussi un sous-ensemble d'attributs pertinents.
- Nous terminons en présentant l'algorithme ω^{β}-SOM qui introduit une pondération des attributs dans la fonction de coût des cartes auto-organisatrices - *Self Organizing Maps* - qui est ensuite optimisée itérativement en alternant trois étapes : optimisation des affectations, optimisation des prototypes et optimisation des poids. La pondération obtenue après convergence est ensuite utilisée pour proposer une approche filtre - *Filter* - de sélection de variables.

Nous concluons cette thèse en indiquant les limites des approches proposées et envisageant quelques axes à développer lors de la poursuite ces recherches.

Sommaire

CHAPITRE 1
Introduction

La classification automatique - *clustering* - est une étape importante du processus d'extraction de connaissances à partir de données (ECD). Elle vise à découvrir la structure intrinsèque d'un ensemble d'objets en formant des regroupements - *clusters* - qui partagent des caractéristiques similaires. La complexité de cette tâche s'est fortement accrue ces deux dernières décennies lorsque les masses de données disponibles ont vu leur volume exploser. La taille des données peut être mesurée selon deux dimensions, le nombre de variables et le nombre d'exemples. Ces deux dimensions peuvent prendre des valeurs très élevées, ce qui peut poser un problème lors de l'exploration et l'analyse de ces données. Pour cela, il est fondamental de mettre en place des outils de traitement de données permettant une meilleure compréhension de la valeur des connaissances disponibles dans ces données. La réduction des dimensions est l'une des plus vieilles approches permettant d'apporter des éléments de réponse à ce problème. Son objectif est de sélectionner ou d'extraire un sous-ensemble optimal de caractéristiques pertinentes pour un critère fixé auparavant. La sélection de ce sous-ensemble de caractéristiques permet d'éliminer les informations non-pertinentes et redondantes selon le critère utilisé. Cette sélection/extraction permet donc de réduire la dimension de l'espace des exemples et rendre l'ensemble des données plus représentatif du problème. En effet, les principaux objectifs de la réduction de dimension sont :

- faciliter la visualisation et la compréhension des données,
- réduire l'espace de stockage nécessaire,
- réduire le temps d'apprentissage et d'utilisation,
- identifier les facteurs pertinents.

Les algorithmes d'apprentissage artificiel requièrent typiquement peu de traits - *features* - ou de variables - attributs - très significatifs caractérisant le phénomène étudié. Dans le domaine de la reconnaissance des formes et de la fouille de données, il pourrait encore être bénéfique d'incorporer un module de réduction de la dimension dans le système global avec comme objectif d'enlever toute information inconséquente et redondante. Cela a un effet important sur la performance du système. En effet le nombre de caractéristiques utilisées est directement lié à l'erreur finale. L'importance de chaque caractéristique dépend de la taille de la base d'apprentissage - pour un échantillon de petite taille, l'élimination d'une caractéristique importante peut diminuer l'erreur. Il faut aussi noter que des caractéristiques individuellement peu pertinentes peuvent être très informatives si on les utilise conjointement.

La réduction de la dimension est un problème complexe qui permet de réduire le volume d'informations à traiter et faciliter le processus de l'apprentissage.

Nous pouvons classer toutes les techniques mathématiques de réduction des dimensions en deux grandes catégories :

- la sélection de variables : qui consiste à choisir des caractéristiques dans l'espace de mesure, (figure 1.1)
- et l'extraction de traits : qui vise à sélectionner des caractéristiques dans un espace transformé - dans un espace de projection (figure 1.2)

Dans cette thèse, nous nous intéressons à la réduction de dimension dans le cadre de la classification non supervisée. Il s'agit d'un domaine de recherche encore peu exploré qui est plus difficile que dans

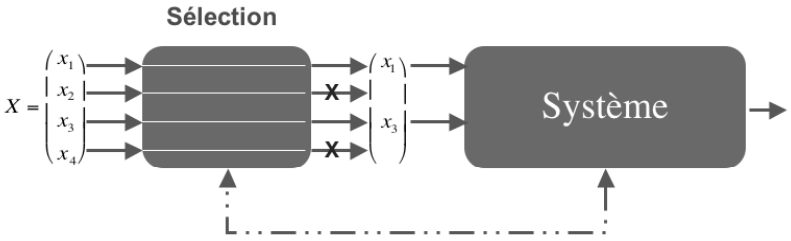

Figure 1.1 – Principe de la sélection de variables.

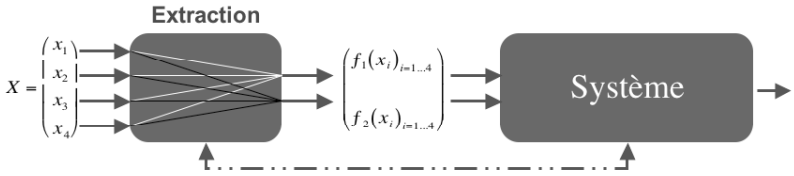

Figure 1.2 – Principe de l'extraction de caractéristiques.

le contexte de l'apprentissage supervisé où l'on dispose d'information pouvant guidée la procédure de réduction de dimension. Différentes approches de sélection ou de pondération de variables sont proposées pour traiter les problèmes liés à la présence d'attributs redondants ou d'attributs fortement bruités :

- Nous proposons d'abord l'algorithme μ-SOM qui limite l'effet de la présence d'attributs redondants en calculant une pondération des attributs à partir d'une classification simultanée des objets et des attributs.
- Nous présentons ensuite une approche intégrée – *embedded* – de sélection de variables pour la classification automatique qui permet de découvrir à la fois le nombre de groupes d'objets présents dans les données mais aussi un sous-ensemble associé d'attributs pertinents.
- Nous terminons en présentant l'algorithme ω^{β}-SOM qui introduit une pondération des attributs dans la fonction de coût des cartes auto-organisatrices - *Self Organizing Maps* - qui est ensuite optimisée itérativement en alternant trois étapes : optimisation des affectations, optimisation des prototypes et optimisation des poids. La pondération obtenue après convergence est ensuite utilisée pour proposer une approche filtre - *Filter* - de sélection de variables.

Nous concluons cette thèse en indiquant les limites des approches proposées et en envisageant quelques axes à développer lors de la poursuite ces recherches.

PARTIE I

Etat de l’art

CHAPITRE 2

Classification non-supervisée

La classification non supervisée ou classification automatique - *clustering* - est une étape importante de l'analyse de données ; elle consiste à identifier des groupes d'objets ou d'individus similaires - *clusters* - à partir d'un ensemble de données sans en connaître au préalable la structure. Elle ne doit pas être confondue avec la classification supervisée ou classement - *classification* - qui consiste à déterminer les règles qui ont permis de séparer un ensemble d'individus en classes connues à priori. L'objectif de ce chapitre est d'introduire les concepts et les notions nécessaires à la compréhension du reste du manuscrit au travers d'un survol rapide du domaine. Le lecteur intéressé est invité à consulter l'une des nombreuses références disponibles [Ber02, Fun01, JD88, JMF99, XW05] pour appronfondir son étude.

Nous commençons par rappeler quelques concepts et définitions avant de présenter quelques approches utilisées en classification automatique. La classification sous contrainte est ensuite présentée comme un moyen d'introduire des connaissances à priori aux algorithmes de classification automatique. Nous terminons ce chapitre sur la question de l'évaluation d'une classification à l'aide de critère de validité.

2.1 Concepts et définitions utiles

2.1.1 Qu'est-ce qu'une classification ?

Le concept de classification est étroitement lié à la notion de partition d'un ensemble fini et nous utiliserons ces deux termes de manière interchangeable tout au long de ce manuscrit. La définition qui suit correspond à la notion de **classification dure** mais ce qualificatif ne sera plus précisé dans la suite du document.

Définition 2.1.1 (Partition d'un ensemble fini) *Étant donné un ensemble fini d'objets Ω, on appelle partition de Ω toute famille de parties non vides de Ω disjointes deux à deux dont l'union forme l'ensemble Ω. Ainsi, si $\mathcal{C}$ est une partition de Ω, alors :*

$$\mathcal{C} = \left\{ \mathcal{C}_i \in \mathcal{P}(\Omega) \setminus \{\varnothing\} : \bigoplus_{i=1}^{K} \mathcal{C}_i = \Omega \right\}$$

La définition précédente impose deux contraintes fortes ; d'une part, tous les objets doivent appartenir à une classe et d'autre part, cette classe doit être unique. Lorsqu'on autorise certains objets à rester sans affectation, on parle de **classification partielle**. Ensuite, si la deuxième contrainte est relâchée, un objet peut alors se trouver dans différentes classes et on parle de **classification douce**. Enfin, en ajoutant la

notion de degré d'appartenance à une classe, on se place dans le contexte des ensembles flous et on parle de **classification floue**. Avant de définir le concept de classification hiérarchique, commençons par introduire une relation d'ordre $\prec$ sur les partitions .

Définition 2.1.2 (Relation d'ordre $\prec$) *On dit qu'une partition $\mathcal{C}$ est plus fine ou égale à une partition $\mathcal{C}'$, si chacune de ses parties $\mathcal{C}_i$ est incluse dans une partie $\mathcal{C}'_j$ de $\mathcal{C}'$ et on note $\mathcal{C} \preccurlyeq \mathcal{C}'$.*

$$\mathcal{C} \preccurlyeq \mathcal{C}' \Leftrightarrow (\forall\, \mathcal{C}_i \in \mathcal{C}) \left(\exists\, \mathcal{C}'_j \in \mathcal{C}' \ :\ \mathcal{C}_i \subseteq \mathcal{C}'_j\right)$$

Si de plus les partitions $\mathcal{C}$ et $\mathcal{C}'$ sont différentes, on note $\mathcal{C} \prec \mathcal{C}'$.

Une **classification hiérarchique** est une suite de partitions emboîtées $\mathcal{C}^{(0)} \prec \mathcal{C}^{(1)} \prec \ldots \prec \mathcal{C}^{(N)} = \{\Omega\}$ dont le premier terme $\mathcal{C}^{(0)}$ est la partition la plus fine qui ne contient que des singletons et dont le dernier terme est la partition la plus grossière qui ne comporte qu'une seule partie. La figure 2.1 illustre ce concept dans le cas d'un ensemble de quatre objets.

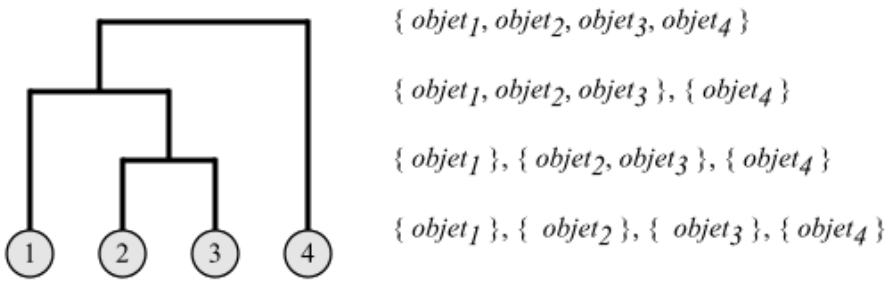

Figure 2.1 – Exemple de classification hiérarchique d'un ensemble de quatre objets. La base de la hiérarchie correspond à la classification la plus fine et on monte d'un niveau en fusionnant deux parties.

2.1.2 Qu'est-ce qu'un groupe d'objets similaires ?

Comme nous l'avons mentionné au début de ce chapitre, la classification automatique vise à former des groupes d'individus similaires. Cette notion de similarité est un élément essentiel de la classification automatique et l'exemple ci-dessous rappelle que c'est le biais introduit par la mesure de similarité qui permet de former des groupes.

Exemple 2.1.1 *Considérons un ensemble de quatre animaux : une baleine, un bar, une poule et une vache. Selon le point de vue adopté, toutes les partitions de cet ensemble sont acceptables comme classification : on peut vouloir distinguer les petits des gros animaux, les mammifères des ovipares, les animaux terrestres des animaux marins, etc.*

Il est commun de définir le concept de similarité à l'aide de la notion duale de dissimilarité ; on dit de deux individus qu'ils sont d'autant plus similaires qu'ils sont proches au sens d'une mesure de dissimilarité. Nous rappelons ci-dessous la définition générale d'une mesure de dissimilarité (définition 2.1.3) avant de considérer le cas des métriques et des ultramétriques qui sont deux types de mesures particulières.

Définition 2.1.3 (Mesure de dissimilarité) *On appelle indice ou mesure de dissimilarité sur un ensemble Ω, une application $d : \Omega \times \Omega \rightarrow \mathbf{R}_+$ qui vérifie les propriétés suivantes pour tout couple $(x, y) \in \Omega \times \Omega$:*

$$\begin{array}{rcll} d(x,y) & = & d(y,x) & \textit{(symétrie)} \\ d(x,y) = 0 & \Leftrightarrow & x = y & \textit{(séparabilité)} \end{array}$$

Définition 2.1.4 (Métrique) *On appelle métrique sur un ensemble Ω, une application $d : \Omega \times \Omega \rightarrow \mathbf{R}_+$ qui vérifie les propriétés suivantes pour tout couple $(x, y) \in \Omega \times \Omega$:*

$$\begin{array}{rcll} d(x,y) & = & d(y,x) & \textit{(symétrie)} \\ d(x,y) = 0 & \Leftrightarrow & x = y & \textit{(séparabilité)} \\ d(x,y) & \leq & d(x,z) + d(z,y) & \textit{(inégalité triangulaire)} \end{array}$$

Définition 2.1.5 (Ultramétrique) *On appelle ultramétrique sur un ensemble Ω, une application d : $\Omega \times \Omega \rightarrow \mathbf{R}_+$ qui vérifie les propriétés suivantes pour tout couple $(x, y) \in \Omega \times \Omega$:*

$$\begin{array}{rcll} d(x,y) & = & d(y,x) & \textit{(symétrie)} \\ d(x,y) = 0 & \Leftrightarrow & x = y & \textit{(séparabilité)} \\ d(x,y) & \leq & \max\{d(x,z), d(z,y)\} & \textit{(inégalité ultramétrique)} \end{array}$$

L'homogènéïté des individus regroupés au sein d'un groupe est souvent évaluée à l'aide d'un critère statistique appelée **variance** dont la définition est rappelée ci-dessous.

Définition 2.1.6 (Variance) *On définit la variance $V(\mathcal{C}_i)$ d'un groupe d'objets $\mathcal{C}_i$ ainsi :*

$$V(\mathcal{C}_i) = \frac{1}{N_i} \sum_{x_j \in \mathcal{C}_i} d^2(x_j - \mu_i)$$

où N_i et μ_i sont respectivement le nombre d'objets et le centroïde du groupe $\mathcal{C}_i$.

Dans le contexte de la classification automatique, on distingue généralement **la variance intra-classe** V_{intra}, que l'on souhaite minimiser, de la **variance inter-classe** V_{inter}, que l'on cherche à maximiser :

$$\begin{aligned} V_{intra} &= \frac{1}{N} \sum_{\mathcal{C}_i \in \mathcal{C}} N_i \times V(\mathcal{C}_i) \\ V_{inter} &= \frac{1}{N} \sum_{\mathcal{C}_i \in \mathcal{C}} N_i \times (\mu_i - \mu)^2 \end{aligned}$$

où N_i et μ_i sont respectivement le nombre d'objets et le centroïde du groupe $\mathcal{C}_i$, et de manière analogue, N et μ désignent respectivement le nombre d'objets et le centroïde de Ω. La première évalue l'homogénéïté moyenne des groupes d'une partition et la seconde permet de quantifier la différence entre les groupes. La **formule de König-Huyghens** permet de relier la variance intra-classe et inter-classe à la variance totale $V_{totale} = V(\Omega)$:

$$V_{totale} = V_{intra} + V_{inter}$$

2.1.3 Comment représenter un objet ?

Comme le suggère l'exemple 2.1.1, le concept de similarité ou le concept dual de dissimilarité repose sur la notion de représentation des objets. Si seule une matrice de similarité (ou de dissimilarité) entre les objets pris deux à deux est disponible, on parle de **représentation implicite**[1]. Lorsqu'une représentation est disponible, on parle de **représentation explicite** et lorsque celle-ci n'apporte pas toute l'information souhaitée on parle de **représentation incomplète**. De nombreux formalismes ont été développés et la

[1] La méthode de positionnement multidimensionnel - *Multi Dimensional Scaling* (MDS) - permet de construire une représentation vectorielle explicite à partir d'une matrice de dissimilarité (voir section 4.3.1.3)

représentation des connaissances est encore un domaine de recherche actif ; le lecteur intéressé trouvera une introduction à ce domaine dans [Kay97]. Une représentation peut prendre diverses formes plus ou moins complexes (tables, arbres, graphes, etc.) mais nous ne considérerons dans cette thèse que la représentations des données sous forme de table qui est la plus largement répandue dans les applications de la fouille de données[2].

Un tableau de données peut contenir des **variables continues**, qui servent à mesurer un caractère **quantitatif**, et des **variables discrètes** qui spécifient un caractère **qualitatif**. On distingue généralement les **variables discrètes ordinales**, dont les différentes valeurs ou **modalités** sont ordonnées, des **variables discrètes nominales** pour lesquelles aucun ordre n'est défini.

Exemple 2.1.2 *Prenons pour exemple le tableau de données suivant qu'un vétérinaire pourrait tenir à jour pour le suivi de ces "patients" :*

Nom	Race	Groupe	Hauteur	Poids	Taille
Belle	Montagne des Pyrénées	II	71 cm	45 kg	grand
Bilitis	Berger Allemand	I	59 cm	32 kg	grand
Ector	Boxer	II	63 cm	38 kg	grand
⋮	⋮	⋮	⋮	⋮	⋮
Hindy	Berger Belge Malinois	I	56 cm	20 kg	moyen
Milou	Fox Terrier à poil dur	III	33 cm	8 kg	petit
Nimbus	Yorkshire	III	20 cm	1 kg	petit

De prime abord, le type des différents attributs présents dans le tableau peut sembler évident, mais il est en fait souvent discutable. L'attribut Nom sera généralement considéré comme une variable nominale bien qu'elle puisse être porteuse d'une information concernant l'âge relatif des sujets dans le cas des chiens de races ; en effet, la première lettre du nom correspond la plus souvent à l'année de naissance et dans ce cas nous sommes en présence d'une variable ordinale. Une discussion analogue du caractère nominal de la variable Race est plus difficile et comme le caractère continu des variables Hauteur et Poids, le caractère ordinal de l'attribut Taille sera moins souvent remis question. En revanche, l'attribut Groupe ne doit pas être considéré comme ordinal mais comme nominal car il correspond au groupe d'utilisation des races canines que nous rappelons dans le tableau ci-dessous :

Groupe	Description
I	Les bergers et les bouviers
II	Les pinshers, les shnauzers et les molosses
III	Les terriers
IV	Les teckels
V	Les chiens nordiques et les spitz
VI	Les chiens courants
VII	Les chiens d'arrêt
VIII	Les leveurs de gibiers, les retrievers et les chiens d'eau
IX	Les chiens de compagnie
X	Les lévriers

[2] Source : résultats d'enquêtes disponibles sur *http ://www.kdnuggets.com*

2.2 Quelques approches classiques

2.2.1 Méthodes hiérarchiques

Au début de ce chapitre, nous avons défini une classification hiérarchique comme une suite ordonnée de partitions emboîtées $\left(\mathcal{C}^{(n)}\right)$ dont le premier terme $\mathcal{C}^{(0)}$ est la partition la plus fine qui ne contient que des singletons, et le dernier terme est la partition la plus grossière qui ne comporte qu'une seule partie. On distingue deux types d'approches de classification hiérarchique : les méthodes descendantes - *divisive* - et les méthodes ascendantes - *agglomerative*.

2.2.1.1 Méthodes descendantes

Elles considèrent l'ensemble des observations Ω et procèdent par division successive jusqu'à obtenir une partition formée de singletons. Nous ne détaillerons pas d'avantage ces méthodes qui sont trop coûteuses pour être utilisées sur les volumes de données manipulés aujourd'hui. En effet, la division d'une partie à N éléments nécessitent l'évaluation des $(2^{N-1}-1)$ divisions possibles.

2.2.1.2 Méthodes ascendantes

Elles commencent avec la partition de l'ensemble Ω la plus fine et procèdent ensuite par fusion progressive des parties jusqu'à obtention de la partition la plus grossière. On obtient ainsi un arbre binaire dont la racine correspond à la partition ne comportant qu'une seule partie et dont les feuilles s'identifient aux différents singletons. Les différents noeuds intermédiaires correspondent à la fusion de deux parties. La Classification Ascendante Hiérarchique (CAH) est sans nulle doute la méthode la plus largement utilisée de cette catégorie. Différents indices d'agrégation de groupes ont été proposés :

- L'indice du saut minimum est défini comme la distance minimale qui sépare deux éléments issus de groupes différents.
- L'indice du saut maximum correspond à la distance maximale qui sépare deux éléments issus de groupes différents.
- L'indice du saut moyen est l'espérance de la distance qui sépare deux éléments issus de groupes différents.
- La distance entre les centroïdes des groupes qui se calcule au plus en temps linéaire $O(N)$ contrairement aux indices précédents dont la complexité est quadratique $\theta(N^2)$.
- L'indice de Ward est défini comme l'augmentation de la variance intra-classe résultant de la fusion des deux groupes considérés.

Comme l'illustre les figures 2.3 et 2.4, il convient de souligner que le résultat d'une CAH est fortement conditionné par le choix du critère d'agrégation. Par ailleurs, on souhaite généralement que la hiérarchie obtenue, indicée par la valeur du critère d'agrégation soit monotone[3], cette propriété n'est pas vérifiée lorsqu'on utilise la distance entre les centroïdes comme critère d'agrégation. Rappelons enfin que d'autres méthodes de classification hiérarchique ont été proposées ; le lecteur intéressé trouvera notamment une présentation des algorithmes BIRCH (*Balanced Iterative Reducing and Clustering using Hierarchies*) et CURE (*Clustering Using REpresentative*) dans [Azz05].

[3] On peut associer une suite $(r_i \in \mathbf{R})_{0\leq i\leq N}$ à une hiérarchie de partition $\mathcal{C}^{(0)} \prec \ldots \prec \mathcal{C}^{(N)}$. On dit alors que la hiérarchie $\left(\mathcal{C}^{(i)}\right)_{0\leq i\leq N}$ indicée par la suite $(r_i \in \mathbf{R})_{0\leq i\leq N}$ est monotone si cette suite d'indice est soit croissante, soit décroissante.

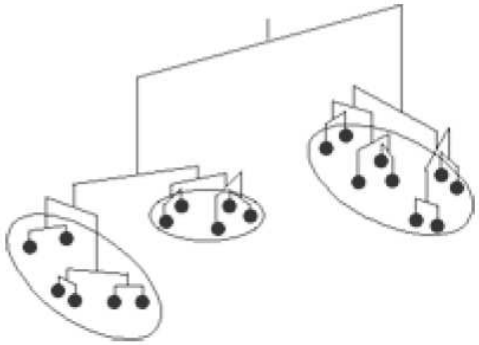

Figure 2.2 – Classification Ascendante Hiérarchique.

2.2.2 Nuées dynamiques

Les différentes partitions obtenues par les méthodes hiérarchiques présentées au paragraphe précédent sont représentées explicitement. Dans le cas des méthodes de type “nuées dynamiques”, chaque groupe est représenté par un prototype, encore appelé centre, et chaque objet est affecté au groupe dont il est le plus proche (figure 2.5). La partition obtenue est alors représentée implicitement par le pavage de Voronoï engendré par l’ensemble des prototypes. Nous commençons par introduire l’algorithme des K-moyennes avant d’en présenter une extension aux classifications floues.

2.2.2.1 K-moyennes

L’algorithme des K-moyennes consiste à choisir aléatoirement des centres initiaux et améliorer la partition obtenue de manière itérative en alternant les deux étapes suivantes jusqu’à stabilisation :

- **étape d’affectation :** chaque objet $x \in \Omega$ est affecté au centre le plus proche, noté $\phi(x)$,
- **étape d’optimisation :** chaque centre est remplacé par le barycentre de l’ensemble des objets qu’il représente.

Le critère optimisé par cet algorithme est défini par :

$$R_{K-moyennes} = \sum_{x \in \Omega} \|x - \phi(x)\|^2 \tag{2.1}$$

Bien que beaucoup plus rapide que la CAH, cet algorithme est très instable et converge vers des minima locaux. On choisit généralement la meilleure solution obtenue après plusieurs exécutions de l’algorithme sans toutefois avoir de garantie d’optimalité globale de la partition retenue. Néanmoins, de nombreuses modifications de l’algorithme initial ont été proposées pour essayer de palier à ces problèmes. L’algorithme des K-moyennes globales - *global kmeans* - proposée dans [LVV03] commence en considérant le barycentre des objets comme centre initial. Ensuite, l’objet qui maximise la diminution de l’erreur est ajouté comme nouveau prototype après chaque convergence de l’algorithme qui s’arrête lorsque le nombre de groupes souhaité est atteint. Bien que les solutions obtenues par cette approche soient stables, [HNCM05] montrent qu’elles ne sont en général pas optimales.

Outre les problèmes d’instabilité et d’optimalité que nous venons de soulever, cette approche nécessite de connaître à priori le nombre de centres. En pratique, on ignore souvent le nombre de groupes présents dans l’ensemble des objets et il est donc nécessaire d’exécuter l’algorithme pour différentes valeurs de ce paramètre. Notons que le critère $R_{K-moyennes}$ décroît lorsque le nombre de groupes augmente

et qu'il n'est donc pas adapté pour choisir le nombre de groupe optimal. Nous verrons au paragraphe 2.5 qu'il convient d'utiliser à cet effet l'un des nombreux critères de qualité proposés dans la littérature. Malgré les multiples exécutions requises par l'utilisation de la méthode des K-moyennes, cette approche conserve l'avantage sur la CAH lorsque le nombre de centres K reste faible devant le nombre d'objets ; sa complexité est en $\theta(N.K)$ contre une complexité en $O(N^2)$ pour la CAH.

Il convient de remarquer qu'en faisant appel à la notion de barycentre, l'algorithme décrit ci-dessus suppose implicitement que les objets sont représentés par un ensemble de valeurs continues. Lorsque les objets sont décrits par des variables nominales, ou plus généralement, lorsque l'utilisateur souhaite qu'un prototype corresponde à un objet observable le barycentre utilisé pour la mise à jour des prototypes peut être remplacé par l'objet médian ou l'objet le plus proche du barycentre ; ces alternatives sont appelées respectivement **K-médianes** et **K-médoïdes**.

2.2.2.2 K-moyennes floues

L'algorithme des K-moyennes présenté ci-dessus conduit à une partition dure et Dunn en a proposé une extension qui conduit à une partition floue. Celle-ci minimise la fonction de coût suivante :

$$R_{K-moyennes-floues} = \sum_{x\in\Omega}\sum_{i=1}^{K}(\mu_i(x))^f \times \|x-\omega_i\|^2 \tag{2.2}$$

où K, $\mu_i(x)$ et ω_i sont respectivement le nombre de centres, le degré d'appartenance de l'objet x au groupe $\mathcal{C}_i$, le centre du groupe $\mathcal{C}_i$. Le paramètre $f > 1$ permet d'ajuster le niveau d'importance accordé aux degrés d'appartenance.

Pour optimiser le critère donné ci-dessus, on utilise une procédure itérative similaire à celle utilisée dans le cas des K-moyennes ; les deux phases suivantes sont répétées :

- **calcul des degrés d'appartenance :** le degré d'appartenance d'un objet x au groupe $\mathcal{C}_i$ est définie par :
$$\mu_i(x) = \left[\sum_{k=1}^{K}\left(\frac{\|x-\omega_i\|}{\|x-\omega_k\|}\right)^{\frac{2}{f-1}}\right]^{-1} \tag{2.3}$$
- **étape d'optimisation :** chaque centre est remplacé par le barycentre pondéré par les degrés d'appartenance de l'ensemble des objets.

2.2.3 Modèles de mélange

2.2.3.1 Principe

On suppose que l'ensemble d'objets dont on dispose a été obtenu en fusionnant plusieurs sous-populations qui suivent chacune une loi de probabilité propre. La probabilité qu'un objet x soit issu de ce mélange de paramètres $\theta = (\alpha_1, \theta_1, \ldots, \alpha_i, \theta_i, \ldots)$ est alors donnée par :

$$p(x|\theta) = \sum_i \alpha_i \times p_i(x|\theta_i) \tag{2.4}$$

où les coefficients de mélange α_i satisfont $\sum_i \alpha_i = 1$, et où les densités de probabilité de chaque sous-population $\mathcal{C}_i$ sont données par les lois $p_i(x|\theta_i)$ de paramètres θ_i. Rappelons que toute distribution continue peut être approximée à l'aide d'un modèle de mélange dès lors que ses composantes sont assez nombreuses et que leurs paramètres sont bien choisis.

L'estimation du nombre et des paramètres de composantes est un problème difficile et dans la plupart des applications seuls les mélanges de lois normales sont considérés. Lorsqu'on impose de plus que toutes les lois normales du mélange aient la matrice identité comme matrice de covariance, on retrouve le cas des k-moyennes.

2.2.3.2 Algorithme EM

L'algorithme le plus répandu pour estimer les paramètres d'un mélange est l'algorithme EM - *Expectation Maximization* - introduit par Dempster et al. en 1977 [DHG01, MB88]. Il consiste à itérer les deux phases suivantes jusqu'à ce que l'amélioration de la log vraisemblance du modèle soit inférieure à un seuil $\epsilon > 0$ fixé :

1. **Estimation :** on suppose fixés les paramètres $\hat{\theta} = \left(\hat{\alpha}_1, \hat{\theta}_1, \hat{\alpha}_2, \hat{\theta}_2, \ldots\right)$ du modèle et on calcule la probabilité $p(x|\hat{\theta}_i)$ qu'un objet $x \in \Omega$ ait été généré par la composante correspondant à la sous-population $\mathcal{C}_i$:
$$p(x|\hat{\theta}_i) = \frac{\alpha_i \times p(x|\hat{\theta}_i)}{\sum_k \alpha_k \times p(x|\hat{\theta}_k)} \tag{2.5}$$
2. **Maximisation :** on suppose cette fois fixée la partition floue de l'ensemble des objets $x \in \Omega$ dont les degrés d'appartenance sont donnés par les probabilités $p(x|\hat{\theta}_i)$. On cherche alors les paramètres $\widetilde{\theta}$ du modèle qui maximisent sa log vraisemblance
$$\log L(\theta|\Omega) = \sum_{x\in\Omega} p(x|\theta) \tag{2.6}$$
$$\widetilde{\theta} = \arg\max_{\theta} \{\log L(\theta|\Omega)\} \tag{2.7}$$
Les coefficients optimaux du mélange sont définis par :
$$\widetilde{\alpha_i} = \frac{1}{N} \sum_{x\in\Omega} x \times p(x|\hat{\theta}_i) \tag{2.8}$$
où N est le nombre d'objets présents dans Ω. Et dans le cas d'un mélange de lois normales $N(\mu_i, \Sigma_i)$, les paramètres optimaux $\widetilde{\theta_i} = (\widetilde{\mu_i}, \widetilde{\Sigma_i})$ sont obtenus ainsi :
$$\widetilde{\mu_i} = \frac{1}{N \times \widetilde{\alpha_i}} \times \sum_{x\in\Omega} x \times p(x|\hat{\theta}_i)$$
$$\widetilde{\Sigma_i} = \frac{1}{N \times \widetilde{\alpha_i}} \times (x - \widetilde{\mu_i})(x - \widetilde{\mu_i})^T \times p(x|\hat{\theta}_i)$$

2.3 Approche neuromimétique : les cartes auto-organisées de Kohonen

2.3.1 Sources historiques et principes

L'algorithme des "cartes auto-organisées", ou "cartes topologiques" - *Self-Organizing Maps (SOM)* - a été introduit par Kohonen au début des années 80 pour modéliser un phénomène, couramment observé dans le cerveau : la formation de "cartes".

Dans le cortex cérébral, on peut remarquer une organisation en régions qui correspondent à différentes modalités sensorielles : pour chaque région corticale, la structure topologique est la même que la

structure topologique du capteur correspondant. On a ainsi des cartes rétinotopiques, somato-sensorielles, etc. Ces cartes se distinguent par la propriété commune suivante : pour un espace de données fixé, par exemple les signaux lumineux sur la rétine, la carte corticale est une représentation à deux dimensions telle que des données “voisines” aient des représentations voisines. Par exemple, la structure spatiale des réponses des cellules dans le cortex auditif correspond à la fréquence des sons perçus. Un certain nombre des fonctions sensorielles sont donc dépendantes de la réalisation d’applications qui conservent la topologie entre l’espace des données (sur les capteurs) et l’espace des représentations (dans la zone corticale correspondante).

Du point de vue informatique, on peut traduire cette propriété de la façon suivante : supposons que l’on dispose de données que l’on désire classifier. On cherche un mode de représentation tel que des données voisines soient classées dans la même classe ou dans des classes voisines. L’algorithme proposé par Kohonen produit un réseau qui a cette propriété : on obtient grâce au réseau une “représentation” de l’ensemble d’apprentissage telle que des exemples proches, mesurés dans le référentiel d’entrée, aient des représentations proches, mesurées dans le réseau. C’est une technique d’apprentissage non supervisé : les exemples sont présentés au réseau qui réorganise progressivement “de lui-même” ses poids de façon à produire l’organisation recherchée.

2.3.2 Description

Le procédé d’auto-organisation proposé par Kohonen cherche à transformer des signaux de départ de dimension quelconque, en général assez grande, en signaux à une ou deux dimensions. Le but principal du réseau est ici de reproduire en sortie du réseau les corrélations qui sont présentes dans les données présentées à l’entrée. D’une manière générale, les cartes auto-organisatrices vont projeter les données initiales sur un espace discret et régulier de faible dimension (en général 1 ou 2). Les espaces utilisés sont des treillis réguliers dont chacun des noeuds est occupé par un automate (neurone formel), la notion de voisinage entre neurones découle alors directement de la structure et définie une topologie de la carte. Grâce au procédé d’auto-organisation, la topologie qui lie les données initiales est conservée au niveau des réponses proposées par le réseau. La localisation des neurones actifs reproduit les liens existants au niveau des données initiales. La plupart du temps, puisqu’il s’agit d’un procédé d’apprentissage non supervisé, les relations de voisinages entre formes d’entrée sont inconnues. C’est l’observation des voisinages produits par la carte qui vont permettre l’interprétation des données initiales. En particulier, ils vont définir la notion de formes proches dans l’espace initial.

Les réseaux SOM sont constitués de deux couches (figure 2.6) :

- la couche d’entrée où les données à classer sont présentées. Les états de tous les neurones de cette couche sont forcés aux valeurs des caractéristiques décrivant les formes d’entrées ;
- la couche (topologique) d’adaptation est composée du treillis de neurones selon une géométrie prédéfinie.

Chaque neurone i de la couche topologique est totalement connecté aux neurones de la couche d’entrée. Le vecteur poids $\omega_{.i} = (\omega_{1i}, \ldots, \omega_{ni})$ de ces connexions forme le “référent” ou le prototype associé au neurone, il est de la même dimension que les formes d’entrée.

Pendant la phase d’apprentissage, le processus d’auto-organisation permet de concentrer l’adaptation des poids des connexions essentiellement sur la région de la carte la plus “active”. Cette région d’activité est choisie comme étant le voisinage associé au neurone dont l’état est le plus actif. Le critère de sélection du neurone le plus actif est de chercher celui dont le vecteur de poids est le plus proche au sens de la distance euclidienne de la forme présentée. Il s’agit d’un critère qui est à l’heure actuelle utilisé dans l’algorithme de ces cartes topologiques. C’est l’utilisation de cette notion de voisinage qui introduit les

contraintes topologiques dans la géométrie finale de la carte. Les recherches effectuées par les neuro-physiologistes dans l'étude du système visuel humain ont montré l'existence de ce type de phénomène au niveau des cellules du cortex et le rôle important qu'il joue dans la vision humaine.

2.3.3 Algorithme d'apprentissage

Différents algorithmes d'apprentissage ont été proposés pour l'adaptation des poids de la carte, nous ne présentons que le plus simple d'entre eux et renvoyons à [Koh01] pour les variantes. Nous commencerons d'abord par définir la notion de voisinage sur la carte topologique. Le voisinage V_i d'un neurone i est composé des neurones de la carte qui se situent à l'intérieur d'une zone d'influence. C'est le choix de la fonction h (une fonction noyau positive et symétrique de type gaussien) qui permet d'introduire des zones d'influence autour de chaque neurone. La fonction de voisinage h peut être de la forme suivante :

$$h_{rs} = \frac{1}{\lambda(t)} \exp\left(\frac{d^2(r,s)}{\lambda^2(t)}\right) \tag{2.9}$$

où $\lambda(t)$ est la fonction température modélisant l'étendue du voisinage :

$$\lambda(t) = \lambda_i \left(\frac{\lambda_f}{\lambda_i}\right)^{\frac{t}{T_{max}}} \tag{2.10}$$

avec λ_i et λ_f sont respectivement la température initiale et la température finale (par exemple $\lambda_i = 2$ et $\lambda_f = 0,5$) et T_{max} le nombre maximum attribué au temps (nombre d'itérations x nombre d'exemples d'apprentissage), et la distance de Manhattan d_1 est définie, entre deux neurones de la carte r et s de coordonnées respectives (k,m) et (i,j) par :

$$d_1(r,s) = |i-k| + |j-m| \tag{2.11}$$

La fonction h qui est une gaussienne introduit pour chaque neurone de la carte un voisinage global. La taille de ce voisinage est limitée par l'écart type $\lambda(t)$ de la gaussienne. Les neurones se trouvant au-delà de cette étendue ont une influence négligeable mais non nulle sur le neurone considéré. L'étendue $\lambda(t)$ est une fonction décroissante dans le temps, la fonction voisinage h aura donc la même évolution avec un écart type décroissant dans le temps. L'apprentissage sera réalisé par la minimisation de la distance, entre formes d'entrées et prototypes de la carte, pondérée par la fonction de voisinage h_{ij}. On pourra employer pour cela un algorithme de gradient.

Le critère à minimiser dans ce cas est défini par :

$$R_{SOM} = \sum_{x_i \in \Omega} \sum_{\omega_j \in U} h_{b(i)j} \times \|x_i - \omega_j\|^2 \tag{2.12}$$

où M représente le nombre de neurones de la carte, $b(i)$ est le neurone dont le référent est le plus proche de la forme d'entrée x_i, et h la fonction de voisinage. La version stochastique de l'algorithme d'apprentissage de ce modèle se déroule essentiellement en trois phases :

- la phase d'initialisation où des valeurs aléatoires sont affectées aux poids des connexions (référents ou prototypes) de chaque neurone de la carte ;
- la phase de compétition pendant laquelle, pour toute forme d'entrée x_i, un neurone $b(i)$, de voisinage $V_{b(i)}$, est sélectionné comme gagnant. Ce neurone est celui dont le vecteur de poids est le plus proche au sens de la distance euclidienne de la forme présentée :

$$b(i) = \underset{1 \leq j \leq M}{\arg\min} \|\omega_j - x_i\|^2 \tag{2.13}$$

- la phase d'adaptation où les poids de chaque neurone de la carte sont mis à jour selon les règles d'adaptation suivantes : si $\omega_{.j} \in V_{b(i)}$ ajuster les poids selon la formule :

$$\omega_{.j} \leftarrow \omega_{.j} - \epsilon h_{b(i)j}(\omega_{.j} - x_i) \tag{2.14}$$

Ce processus d'adaptation est répété jusqu'à stabilisation de l'auto-organisation.

Une version *batch* de cet algorithme a été proposée : les vecteurs poids ne sont mis à jour qu'après la présentation de toutes les formes d'entrées et on remplace alors le prototype des neurones par le barycentre pondéré à l'aide de la fonction de voisinage des formes d'entrées qui les ont activés.

2.4 Connaissances du domaine et contraintes

La classification non supervisée permet de former des groupes d'objets susceptibles d'être intéressants pour l'utilisateur. Notons qu'il est fréquemment possible de construire différentes partitions d'un même ensemble d'objets et en absence d'informations complémentaires, le choix de l'une ou l'autre est nécessairement arbitraire. La prise en compte des attentes de l'utilisateur est donc un facteur de succès déterminant de l'application de ce type de méthodes. Nous rappelons dans cette section différentes approches proposées à cet effet ; elles procèdent par introduction de contraintes qui portent sur les groupes, sur les objets ou encore sur les attributs.

2.4.1 Contraintes sur les groupes : forme et taille

2.4.1.1 Contraintes de forme

La forme des groupes est très souvent imposée par le choix de l'algorithme et de la mesure de (dis)similarité. Ainsi, l'algorithme des K-moyennes utilisant une distance euclidienne a tendance à former des groupes hyper-sphériques. Plus généralement, les modèles de mélange permettent d'imposer la forme du nuage de points des différentes sous population en contraignant les paramètres des différentes lois ; dans le cas d'un mélange de lois normales, il est commun d'imposer à la matrice de covariance d'être diagonale : les groupes formés sont ainsi hyper-ellipsoïdaux.

Dans certaines applications, les données revêtent un caractère spatial et il est parfois nécessaire d'obtenir des groupes contigus. La dimension spatiale peut être utilisée soit en ajoutant des variables de position que l'on traite ensuite comme les autres descripteurs, soit en utilisant une phase d'extraction de l'information spatiale en remplaçant par exemple la valeur d'un attribut par sa moyenne dans le voisinage (au sens spatial) de l'objet. L'application de ces approches à la segmentation d'image est illustrée dans [Amb96] et [BLP05] présente l'algorithme Geo-SOM en montrant la plus value apportée dans le cadre des Systèmes d'Informations Géographiques (SIG).

2.4.1.2 Contraintes de taille

Il est parfois souhaitable d'obtenir des groupes de taille plus ou moins homogène et il est possible d'introduire des contraintes sur la taille des clusters. Dans cet esprit une version modifiée de l'algorithme des K-moyennes est proposée dans [BBD00] et [VA99] propose une adaptation de l'algorithme de Kohonen pour les cartes auto-organisées. Il est relativement simple d'imposer une contrainte de ce type à une hiérarchie de partitions, cela revient en effet à définir une hauteur maximale de coupure à chaque branche du dendrogramme. Pour finir, dans le contexte des modèles de mélanges une telle contrainte peut être appliquée en ajoutant un terme de régularisation qui fixe une probabilité à priori maximale qu'aucune classe ne peut dépasser.

2.4.2 Contraintes sur les objets

2.4.2.1 Fusion et Exclusion

Du point de vue de l'utilisateur, un moyen simple de préciser la partition qu'il attend consiste à indiquer quels sont les objets qui doivent être regroupés et quels sont ceux qui doivent s'exclure mutuellement. Ainsi, lors d'un apprentissage actif, l'utilisateur peut affiner progressivement le résultat d'une classification automatique en précisant progressivement les regroupements ou séparations d'objets qu'il considère comme des anomalies. Deux versions modifiées de l'algorithme des K-moyennes sont ainsi proposées dans [Wag02] pour intégrer ce type de contraintes : l'algorithme *COP-KMeans* qui applique strictement l'ensemble des contraintes de fusion et d'exclusion mutuelle d'objets lors de la phase d'affectation des objets à un groupe et l'algorithme *SCOP-KMeans* qui utilise une version relaxée en ajoutant un terme de pénalisation à la fonction de coût optimisée.

2.4.2.2 Etiquetage partiel

Lorsqu'une partition d'un sous-ensemble des objets est connue, il est possible d'appliquer les approches décrite ci-dessus ou d'adopter l'une des approches développées dans [Bas05]. Dans l'algorithme *Seeded-KMeans* les prototypes initiaux ne sont pas choisis aléatoirement mais comme étant les barycentres des classes connues. L'algorithme des K-moyennes est ensuite appliqué normalement sans tenir compte des étiquettes connues. L'algorithme *Constrained-KMeans* initialise les prototypes de la même manière mais ne modifie pas l'affectation des objets dont la classes est connue pendant l'apprentissage.

2.4.3 Contraintes sur les attributs

Il est parfois souhaitable d'obtenir des groupes dans lesquels les valeurs prises par un attribut restent dans un intervalle de faible amplitude. C'est dans ce cadre que [DLC03] propose deux versions de la CAH utilisant l'indice du saut maximum comme critère d'agrégation. La première version, *Constrained Clustering with Complete-Link*, qui procède aux regroupements des objets en respectant la contrainte, est sensible à l'ordre des regroupements et laissent généralement de coté certains objets qui ne peuvent être ajoutés à aucun groupe sans violer la contrainte. Une deuxième version, *Progressive Constraint Relaxation Technique*, est proposée pour corriger ce problème. La contrainte imposée à l'intervalle de valeur dépend alors du niveau auquel le regroupement intervient dans la hiérarchie ; elle est relâchée progressivement.

2.5 Evaluation et critères de validité

Dans le contexte de la classification automatique, il est naturel de s'interroger sur la validité de la partition obtenue. Les groupes découverts correspondent-ils à nos connaissances à priori ? Correspondent-ils vraiment à l'ensemble d'objets dont on dispose ? De deux classifications, laquelle est la plus pertinente ? Ces différentes questions permettent de distinguer trois catégories de critères : les critères externes, les critères internes et les critères relatifs.

Les **critères externes** permettent de répondre à la première question et de mesurer l'adéquation entre une partition et les connaissances à priori dont on dispose. Nous ne les détaillerons pas ici car on se rapproche alors de la classification sous contraintes évoquée à la section 2.4 ou du problème de comparaison de partitions auquel nous consacrons le chapitre 3. Les **critères internes** quantifient l'adéquation entre une partition et l'idée subjective que l'on se fait d'une "bonne" classification. Ainsi, les propriétés les

plus communément recherchées sont la compacité et la séparabilité des groupes découverts. Les **critères relatifs** s'intéressent à la troisième et dernière question et à défaut de donner une appréciation absolue de la validité d'une partition, ils permettent d'ordonner plusieurs classifications et d'en choisir "une meilleure".

2.5.1 Erreur Quadratique Moyenne

L'erreur quadratique moyenne - *Mean Squared Error, MSE* - est une mesure de compacité très répandue, elle est notamment équivalente à la fonction de coût de l'algorithme de K-moyennes présenté au paragraphe 2.2.2.1 :

$$MSE = \frac{1}{N} \times \sum_{i=1}^{N} \sum_{j=1}^{K} c_{ij} \times \|x_i - \omega_j\|^2 \tag{2.15}$$

où K est le nombre de groupes et où $c_{ij} = \mathbf{1}|_{\mathcal{C}_j}(i)$ indique si $x_i \in \mathcal{C}_j$. Lorsqu'on étend cette mesure au cas des partitions floues, on retrouve (à un coefficient multiplicateur près) la fonction de coût optimisée par l'algorithme des K-moyennes floues donnée par l'équation (2.2) :

$$FMSE = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} (\mu_j(x_i))^f \times \|x_i - \omega_j\|^2 \tag{2.16}$$

2.5.2 Indice de Dunn

Dans le cas d'une classification dure, l'indice de Dunn tient compte à la fois de la compacité et de la séparabilité des groupes : la valeur de cet indice est d'autant plus faible que les groupes sont compacts et bien séparés. Notons que la complexité de l'indice de Dunn devient prohibitive dès qu'on manipule de grands ensembles d'objets ; il est par conséquent rarement utilisé.

$$I_{Dunn} = \frac{\min\{D_{min}(\mathcal{C}_i, \mathcal{C}_j) \ : \ i \neq j\}}{\max\{S_{max}(\mathcal{C}_i)\}} \tag{2.17}$$

où $D_{min}(\mathcal{C}_i, \mathcal{C}_j)$ est la distance minimale qui sépare un objet du groupe $\mathcal{C}_i$ d'un objet du groupe $\mathcal{C}_j$ et où $S_{max}(\mathcal{C}_i)$ est la distance maximale qui sépare deux objets du groupe $\mathcal{C}_i$:

$$\begin{aligned} D_{min}(\mathcal{C}_i, \mathcal{C}_j) &= \min \{\|x - y\| \ : \ x \in \mathcal{C}_i \text{ et } y \in \mathcal{C}_j\} \\ S_{max}(\mathcal{C}_i) &= \max \{\|x - y\| \ : \ (x, y) \in \mathcal{C}_i \times \mathcal{C}_i\} \end{aligned}$$

2.5.3 Indice de Davies-Bouldin

Dans le cas d'une classification dure, l'indice de Davies-Bouldin [DB79] tient compte à la fois de la compacité et de la séparabilité des groupes : la valeur de cet indice est d'autant plus faible que les groupes sont compacts et bien séparés. Cet indice dont la complexité en $\theta\left(K \times (N + K)\right)$ est raisonnable favorise les groupes hypersphériques et il est donc particulièrement bien adapté pour une utilisation avec la méthode des K-moyennes.

$$I_{DB} = \frac{1}{K} \sum_{k=1}^{K} \max_{l \neq k} \left\{ \frac{S_c(\mathcal{C}_k) + S_c(\mathcal{C}_l)}{D_{ce}(\mathcal{C}_k, \mathcal{C}_l} \right\} \tag{2.18}$$

où $S_c(\mathcal{C}_i)$ est la distance moyenne entre un objet du groupe $\mathcal{C}_i$ et son centre, et où $D_{ce}(\mathcal{C}_i, \mathcal{C}_j)$ est la distance qui sépare les centres des groupes $\mathcal{C}_i$ et $\mathcal{C}_j$:

$$\begin{aligned} S_c(\mathcal{C}_i) &= \frac{1}{N_i} \sum_{i=1}^{N_i} \|x - \omega_i\| \\ D_{ce}(\mathcal{C}_i, \mathcal{C}_j) &= \|\omega_i - \omega_j\| \end{aligned}$$

2.5.4 Indice de compacité Wemmert et Gançarski

L'indice Wemmert et Gançarski considère à la fois la compacité et la séparabilité des groupes et s'appuie sur le rapport entre deux distances [Bla06] : la distance d'un objet au centre de son groupe et la distance minimale au centre d'un autre groupe. Il se définit ainsi pour un groupe :

$$I_{WG}(\mathcal{C}_i) = \max \left\{ 0 \; ; \; 1 - \frac{1}{N_i} \sum_{x \in \mathcal{C}_i} \frac{\|x - \omega_i\|}{\min\{\|x - \omega_j\| \; : \; j \neq i\}} \right\} \tag{2.19}$$

et la valeur de cet indice pour une partition correspond à la moyenne pondérée de l'indice de chacun des groupes :

$$I_{WG} = \frac{1}{N} \sum_{i=1}^{K} N_i \times I_{WG}(\mathcal{C}_k) \tag{2.20}$$

2.5.4.1 Indice de Xie et Beni

Dans le cas d'une classification floue, il est fréquent d'utiliser l'indice de Xie et Beni pour prendre en considération à la fois la compacité et la séparabilité des groupes. On le définit à partir de l'erreur quadratique moyenne floue *FMSE* pour une valeur du paramètre $f = 2$ de la manière suivante :

$$I_{XB} = \frac{FMSE}{\min\{\|x - y\|^2 \; : \; (x, y) \in \mathcal{C}_i \times \mathcal{C}_j\}} \tag{2.21}$$

Notons que ce critère peut également être utilisé avec une classification dure en remplaçant l'erreur quadratique floue par l'erreur quadratique moyenne.

2.5.5 Indices propres aux cartes auto-organisées

De nombreux indices de qualité ont été développés pour les cartes auto-organisées et nous n'introduisons ici que les plus utilisés ; le lecteur intéressé est invité à consulter [PÖ4] pour approfondir cette question.

2.5.5.1 Erreur de quantification

Les cartes auto-organisées font partie des méthodes de quantification vectorielle et il semble donc naturel de les évaluer à l'aide de l'erreur de quantification moyenne que l'on définit ainsi :

$$Q_{err} = \frac{1}{N} \times \sum_{i=1}^{N} \left\| x_i - \omega_{b(i)} \right\| \tag{2.22}$$

où $b(i)$ est l'indice du prototype le plus proche de l'observation x_i.

2.5.5.2 Taux d'erreurs topologiques

Les cartes auto-organisées sont aussi une méthode de projection de données multidimensionnelles sur un espace de faible dimension et le taux d'erreurs topologiques permet de quantifier la conservation de la topologie locale de l'espace des observations par la carte. On considère qu'il y a une erreur topologique chaque fois que les deux prototypes les plus proches d'une observation ne sont pas voisins sur la carte. Le taux d'erreur topologique peut se définir ainsi :

$$T_{err} = 1 - \frac{1}{N} \times \sum_{i=1}^{N} \mathbf{1}|_{\mathcal{N}(b(i))} \left(\arg\min_{j \neq i} \|x - \omega_j\| \right) \tag{2.23}$$

où $\mathbf{1}|_{\mathcal{N}(b(i))}$ est la fonction indicatrice de l'ensemble des voisins du prototype le plus proche de l'observation x_i.

2.5.5.3 Mesure de distortion

La mesure de distortion permet de prendre en considération les deux aspects évoqués dans les paragraphes précédents (qualité de la quantification et conservation de la topologie locale) et elle s'apparente à l'erreur quadratique floue où les degrés d'appartenance seraient remplacés par la fonction de voisinage :

$$distortion = \sum_{i=1}^{N} \sum_{j} h_{b(i)j} \times \|x - \omega_j\|^2 \tag{2.24}$$

où $h_{b(i)j}$ est la fonction de voisinage. Rappelons que cette mesure peut être décomposée en trois termes [VSH03] qui correspondent à la variance des données dans la région de Voronoï de chaque unité, à la qualité topologique de la carte et à la pression liée au compromis entre quantification et conservation topologique.

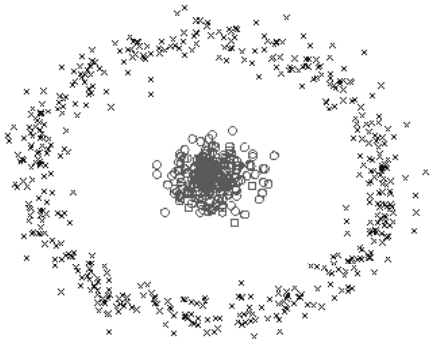

Figure 2.3 – Cas d'une couronne : dans le cas d'une couronne, une CAH utilisant l'indice du saut minimum identifiera parfaitement les deux groupes, en revanche l'utilisation de la distance entre les centroïdes conduira à une classification sans réel intérêt.

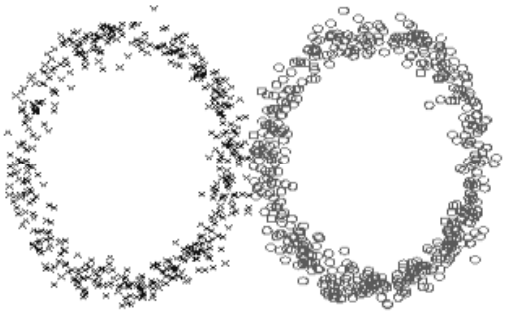

Figure 2.4 – Cas de deux anneaux : lorsque les groupes ne sont pas suffisamment séparés, l'utilisation de l'indice du saut minimum est à proscrire car elle conduirait à ce qu'on appelle "effet de chaîne" : les groupes sont fusionnés de proche en proche et la CAH se révèle incapable de mettre en exergue les deux anneaux. La distance entre les centroïdes ou le critère de Ward conduisent dans ce cas à des classifications plus pertinentes.

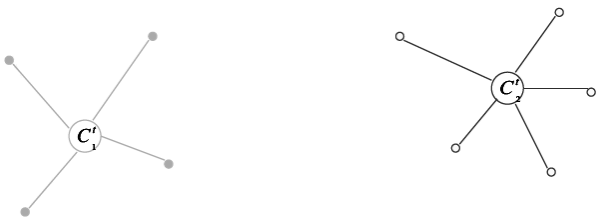

Figure 2.5 – Algorithme des K-moyennes : chaque groupe est représenté par un prototype, encore appelé centre, et chaque objet est affecté au groupe dont il est le plus proche.

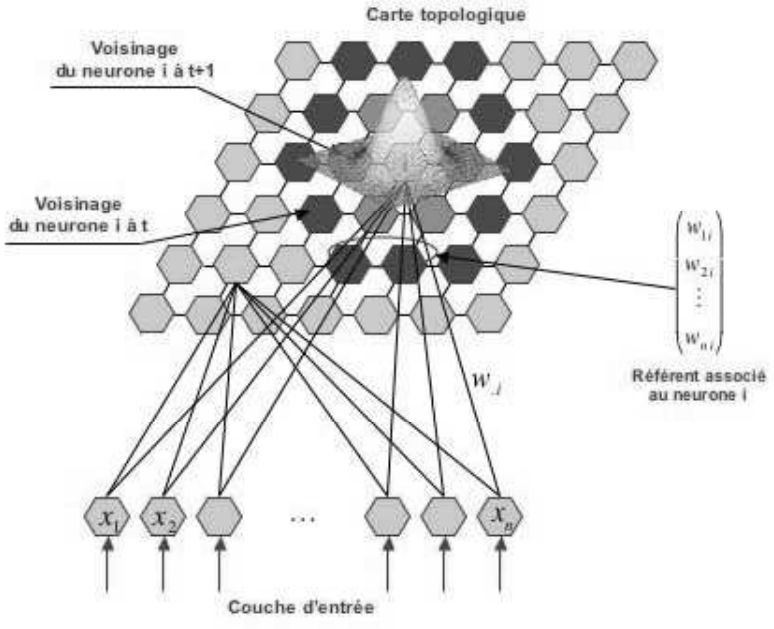

Figure 2.6 – Architecture du réseau pour l'algorithme des cartes topologiques.

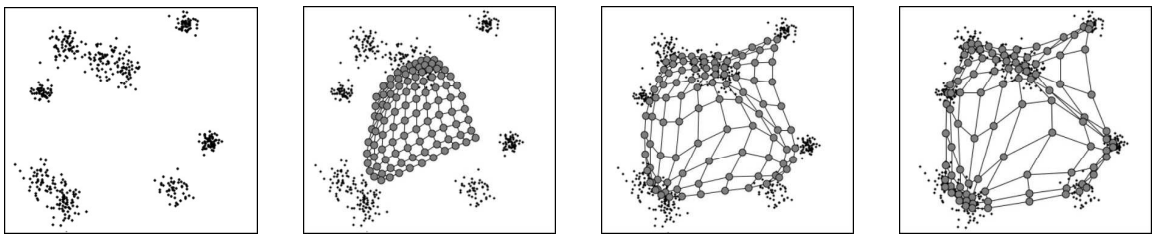

Figure 2.7 – La répartition des observations dans l'espace des formes est donnée par la figure la plus à gauche. Les 3 autres figures montrent le déroulement de l'apprentissage et de l'auto-organisation des référents associés aux neurones de la carte topologique.

CHAPITRE 3

Comparaison de partitions

La comparaison de partitions est un problème clef de la classification automatique. Elle est notamment à la base des critères externes d'évaluation de partitions évoqués au chapitre précédent et elle permet également d'évaluer la stabilité d'un algorithme de classification automatique. Nous lui consacrons ce chapitre qui synthétise et complète les travaux récents de Marina Meilă [Mei03, Mei05, Mei06].

3.1 Espace des partitions

3.1.1 Quelques définitions

La notion de partition peut être abordée selon deux approches complémentaires : on peut soit adopter une vision ensembliste, soit se placer dans le cadre de la théorie des graphes. Dans le dernier cas, l'ensemble des objets Ω est représenté par un graphe complet non orienté $G = (V, E)$ dont l'ensemble des sommets V est en bijection avec l'ensemble des objets. Dans un soucis de simplification des notations, on identifie les sommets aux objets et on a ainsi : $V = \Omega$ et $E = \Omega \times \Omega$.

Définition 3.1.1 (Partition) *Une partition $\mathcal{C}$ d'un ensemble Ω est une famille finie de parties non vides de Ω disjointes deux à deux dont l'union est Ω. Ceci s'exprime formellement de la manière suivante :*

$$\mathcal{C} = \left\{ \mathcal{C}_i \in \mathcal{P}(\Omega) \setminus \{\varnothing\} : \bigoplus_{i=1}^{K} \mathcal{C}_i = \Omega \right\} \tag{3.1}$$

Du point de vue de la théorie des graphes, une partition $\mathcal{C}$ de l'ensemble des objets Ω est représentée par la fermeture transitive[1] *d'un graphe partiel de $G = (\Omega, \Omega \times \Omega)$ que l'on notera $\gamma(\mathcal{C})$.*

Définition 3.1.2 (Raffinement) *Une partition $\mathcal{C}'$ est un raffinement d'une partition $\mathcal{C}$ si elle est obtenue en divisant une partie $\mathcal{C}_i$ en deux sous-parties $\mathcal{C}'_{i'}$ et $\mathcal{C}'_{i''}$. Formellement, on a :*

$$\mathcal{C}' = \{\mathcal{C}_j \in \mathcal{C} : j \neq i\} \cup \{\mathcal{C}'_{i'}, \mathcal{C}'_{i''} : \mathcal{C}'_{i'} \oplus \mathcal{C}'_{i''} = \mathcal{C}_i\} \tag{3.2}$$

Le graphe $\gamma(\mathcal{C})$ est alors égal à la fermeture transitive d'un graphe obtenu en ajoutant une arrête unique au graphe $\gamma(\mathcal{C}')$.

L'extension par transitivité de la notion de raffinement introduite ci-dessus permet de définir une relation d'ordre partiel sur l'ensemble des partitions :

[1] La fermeture transitive d'un graphe est obtenue en saturant l'ensemble des arrêtes sans diminuer le nombre de composantes connexes.

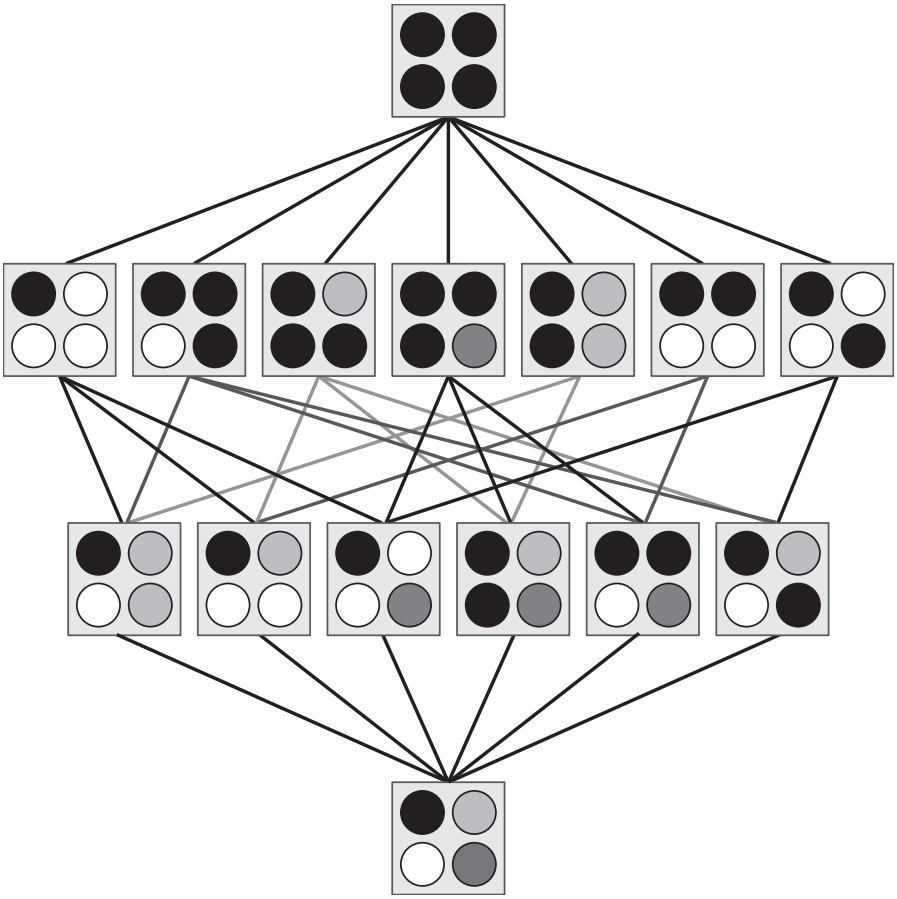

Figure 3.1 – Treillis des partitions d'un ensemble de données comportant quatre exemples.

Définition 3.1.3 (Ordre partiel sur l'ensemble des partitions $\prec$) *On dit qu'une partition $\mathcal{C}'$ est plus fine qu'une partition $\mathcal{C}$, si celle-ci est obtenue par raffinement successif de $\mathcal{C}$ et on note $\mathcal{C}' \prec \mathcal{C}$. Le graphe $\gamma(\mathcal{C}')$ est alors un sous-graphe de $\gamma(\mathcal{C})$.*

L'ensemble des partitions de Ω muni de la relation d'ordre partiel $\prec$ (cf. définition 3.1.3) est un treillis ; la figure 3.1 en donne une illustration pour le cas d'un ensemble de données comportant quatre exemples. Les bornes inférieure et supérieure de ce treillis sont notées respectivement $\hat{0}$ et $\hat{1}$; elles comportent respectivement tous les singletons de $\mathcal{P}(\Omega)$ et l'ensemble Ω.

Notations : Introduisons quelques notations additionnelles utilisées par la suite :

- $\mathcal{C}_i^{\hat{1}} = \{\Omega \setminus \{x_i\}, \{x_i\}\}$
- $\mathcal{C}_{\{i,j\}}^{\hat{0}} = \{\{\{x_k\} : k \notin \{i,j\}\}, \{x_i, x_j\}\}$

Définition 3.1.4 (Produit de partitions) *Le produit de p partitions $\mathcal{C}^{(i)}$ est la borne supérieure de l'ensemble des partitions qui sont simultanément plus fines que toutes les partitions $\mathcal{C}^{(i)}$:*

$$\prod_{i=1}^{p} \mathcal{C}^{(i)} = \sup \left\{ \bigcap_{i=1}^{p} \{\mathcal{C} : \mathcal{C} \prec \mathcal{C}^{(i)}\} \right\} \tag{3.3}$$

Autrement dit, le produit d'un ensemble de partitions $\mathcal{C}^{(i)}$ est la partition formée de l'union des intersections non vides des classes $\mathcal{C}_k^{(i)}$. Si $E^{(i)}$ est l'ensemble des arrêtes de $\gamma(\mathcal{C}^{(i)})$, alors $\gamma\left(\prod_{i=1}^{p}\mathcal{C}^{(i)}\right) = \left(\Omega, \bigcap_{i=1}^{p} E^{(i)}\right)$ est un sous-graphe de chaque $\gamma(\mathcal{C}^{(i)})$.

3.1.2 Outil de comparaison

Tableau de contingence

D'un point de vue ensembliste, pour comparer deux partitions $\mathcal{C}$ et $\mathcal{C}'$ d'un même ensemble de données Ω, on commence généralement par construire un tableau de contingence $C = (n_{ij})$, où n_{ij} est le nombre d'objets qui appartiennent simultanément à la classe $\mathcal{C}_i$ et $\mathcal{C}'_j$. Un exemple en est donné à la figure 3.2, où $n_{i.}$, $n_{.j}$ et N désignent respectivement les marges de la ligne i et de la colonne j, et la somme des marges.

	$\mathcal{C}'_1$	$\dots$	$\mathcal{C}'_j$	$\dots$	$\mathcal{C}'_{K'}$	
$\mathcal{C}_1$	n_{11}	$\dots$	n_{1j}	$\dots$	$n_{1K'}$	$n_{1.}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathcal{C}_i$	n_{i1}	$\dots$	n_{ij}	$\dots$	$n_{iK'}$	$n_{i.}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathcal{C}_K$	n_{K1}	$\dots$	n_{Kj}	$\dots$	$n_{KK'}$	$n_{K.}$
	$n_{.1}$	$\dots$	$n_{.j}$	$\dots$	$n_{.K'}$	N

Figure 3.2 – Tableau de contingence de deux partitions

Tableau de comptage des accords et des désaccords

Lorsqu'on se place dans le cadre de la théorie des graphes, pour comparer deux graphes $\gamma(\mathcal{C})$ et $\gamma(\mathcal{C}')$ on commence généralement par comptabiliser le nombre d'arêtes du graphe complet qui sont absentes ou présentes dans les deux graphes, ou encore celle qui n'apparaissent que dans un des deux graphes. On

		Partition $\mathcal{C}'$	
		1	0
Partition $\mathcal{C}$	1	$N_{11} = \#\, E \cap E'$	$N_{10} = \#\, E \cap \overline{E'}$
	0	$N_{01} = \#\, \overline{E} \cap E'$	$N_{00} = \#\, \overline{E} \cap \overline{E'}$

Figure 3.3 – Tableau de comptage des paires

obtient alors le tableau de la figure 3.3 où E, E', $\overline{E}$ et $\overline{E'}$ sont respectivement les ensembles d'arrêtées de $\gamma(\mathcal{C})$ et $\gamma(\mathcal{C}')$, et les ensembles d'arrêtées du graphe complet absentes de $\gamma(\mathcal{C})$ et $\gamma(\mathcal{C}')$. Notons que la somme des quatre valeurs N_{11}, N_{00}, N_{10} et N_{01} satisfait la relation suivante :

$$N_{11} + N_{00} + N_{10} + N_{01} = \frac{1}{2}N(N-1) \tag{3.4}$$

Relation entre les deux types de tableau

Il convient de rappeler que le tableau 3.3 peut être construit à partir du tableau de contingence 3.2 en utilisant les formules suivantes [HA85] :

$$N_{11} = \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}(n_{ij}-1) \tag{3.5}$$

$$N_{00} = \frac{1}{2}\left(n^2 + \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2 - \left(\sum_{i=1}^{K} n_{i.}^2 + \sum_{j=1}^{K'} n_{.j}^2\right)\right) \tag{3.6}$$

$$N_{01} = \frac{1}{2}\left(\sum_{j=1}^{K'} n_{.j}^2 - \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2\right) \tag{3.7}$$

$$N_{10} = \frac{1}{2}\left(\sum_{i=1}^{K'} n_{i.}^2 - \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2\right) \tag{3.8}$$

Les deux types de tableau introduits ci-dessus permettent d'apprécier qualitativement la similarité de deux partitions et de construire de nombreux critères quantitatifs de comparaison de partitions auquels sont consacrés les les deux prochaines sections.

3.2 Comparaison par comptage de paires et distances binaires

Lorsqu'on adopte une représentation des partitions sous forme de graphe, il convient de remarquer que les critères de comparaison que l'on peut construire à partir de N_{11}, N_{10}, N_{01} et N_{00} correspondent à des mesures de dissimilarité binaires dont un grand nombre peuvent s'exprimer sous la forme suivante [Li06] :

$$d_{\alpha,\delta} = \frac{N_{10}+N_{01}}{\alpha N_{11}+N_{10}+N_{01}+\delta N_{00}} \tag{3.9}$$

où α et δ sont deux paramètres qui permettent de pondérer la prise en compte respective des présences ou absences simultanées d'une arrête dans deux partitions. La table 3.1 rappelle la définition de quelques mesures et le lecteur intéressé en trouvera une présentation plus complète de ces mesures ou de leur propriété dans [JKV01, LLB04, Rou85, Li06].

3.2.1 Précision, Rappel et Critères associés

L'indice de précision et le coefficient de rappel sont des mesures asymétriques de similarité entre deux partitions dont l'une sert de référence.

Définition 3.2.1 (Indice de précision) *Lorsque la partition $\mathcal{C}$ sert de référence, l'indice de précision indique la probabilité que deux objets soient regroupés dans la partition $\mathcal{C}'$ s'ils le sont dans la partition $\mathcal{C}$:*

$$prec(\mathcal{C},\mathcal{C}') = \frac{N_{11}}{N_{11}+N_{01}} \tag{3.10}$$

Mesure	Similarité	Dissimilarité	α	δ	Métrique
Sokal & Sneath (I)	$\frac{\frac{1}{2}N_{11}}{\frac{1}{2}N_{11}+N_{10}+N_{01}}$	$\frac{N_{10}+N_{01}}{\frac{1}{2}N_{11}+N_{10}+N_{01}}$	$\frac{1}{2}$	0	oui
Rogers & Tanimoto	$\frac{\frac{1}{2}(N_{11}+N_{00})}{\frac{1}{2}(N_{11}+N_{00})+N_{10}+N_{01}}$	$\frac{N_{10}+N_{01}}{\frac{1}{2}(N_{11}+N_{00})+N_{10}+N_{01}}$	$\frac{1}{2}$	$\frac{1}{2}$	oui
Jaccard	$\frac{N_{11}}{N_{11}+N_{10}+N_{01}}$	$\frac{N_{10}+N_{01}}{N_{11}+N_{01}+N_{10}}$	1	0	oui
Simple corresp.	$\frac{N_{11}+N_{00}}{N_{11}+N_{10}+N_{01}+N_{00}}$	$\frac{N_{10}+N_{01}}{N_{11}+N_{10}+N_{01}+N_{00}}$	1	1	oui
Czekanowski-Dice	$\frac{2N_{11}}{2N_{11}+N_{10}+N_{01}}$	$\frac{N_{10}+N_{01}}{2N_{11}+N_{10}+N_{01}}$	2	0	non
Sokal & Sneath (II)	$\frac{2(N_{11}+N_{00})}{2(N_{11}+N_{00})+N_{10}+N_{01}}$	$\frac{N_{10}+N_{01}}{2(N_{11}+N_{00})+N_{10}+N_{01}}$	2	2	non
Kulczynski (II)	$\frac{1}{2}\left(\frac{N_{11}}{N_{11}+N_{10}}+\frac{N_{11}}{N_{11}+N_{01}}\right)$	$1-\frac{1}{2}\left(\frac{N_{11}}{N_{11}+N_{10}}+\frac{N_{11}}{N_{11}+N_{01}}\right)$	nd	nd	
Ochiai	$\frac{N_{11}}{\sqrt{(N_{11}+N_{10})(N_{11}+N_{01})}}$	$1-\frac{N_{11}}{\sqrt{(N_{11}+N_{10})(N_{11}+N_{01})}}$	nd	nd	
Russel & Rao	$\frac{N_{11}}{N_{11}+N_{10}+N_{01}+N_{00}}$	$1-\frac{N_{11}}{N_{11}+N_{10}+N_{01}+N_{00}}$	nd	nd	oui

Table 3.1 – Quelques mesures de similarité et de dissimilarité binaire.

Définition 3.2.2 (Coefficient de rappel) *Lorsque la partition $\mathcal{C}$ sert de référence, le coefficient de rappel indique la probabilité que deux objets soient regroupés dans la partition $\mathcal{C}$ s'ils le sont dans la partition $\mathcal{C}'$:*

$$rapp(\mathcal{C},\mathcal{C}') = \frac{N_{11}}{N_{11}+N_{10}} \tag{3.11}$$

Ces deux critères prennent leurs valeurs sur l'intervalle $[0;1]$, mais une valeur de 1 de l'un ou l'autre de ces indices ne doit pas être interprétée comme l'identité des partitions. Un moyen simple de combiner ces deux critères consistent à prendre leurs moyennes arithmétique, géométrique et harmonique. Nous obtenons ainsi respectivement le deuxième coefficient de Kulczynski, l'indice de Folkes & Mallows qui n'est autre que le coefficient de Ochiai et la F_1-mesure qui s'identifie au coefficient de Czekanowski-Dice, également appelé coefficient de Sørensen. Ces trois mesures sont symétriques, prennent leurs valeurs sur l'intervalle $[0;1]$ et sont égales à 1 si et seulement si les deux partitions sont identiques.

Définition 3.2.3 (2ème coefficient de Kulczynski) *Le deuxième coefficient de Kulczynski se définit comme la moyenne arithmétique de l'indice de précision et du coefficient de rappel :*

$$K(\mathcal{C},\mathcal{C}') = \frac{1}{2}\left(prec(\mathcal{C},\mathcal{C}') + rapp(\mathcal{C},\mathcal{C}')\right) \tag{3.12}$$

A l'origine proposé pour comparer deux classifications hiérarchiques [FM83], l'indice de Folkes & Mallows peut être utilisé pour comparer deux partitions d'un même ensemble d'objets. Dans un commentaire

de l'article original, David L. Wallace remarque qu'il s'exprime comme la moyenne géométrique du coefficient de rappel et de l'indice de précision [HA85, Mei03, Mei06, Wal83].

Définition 3.2.4 (Indice de Folkes & Mallows) *L'indice de Folkes & Mallows est défini comme la moyenne géométrique de l'indice de précision et du coefficient de rappel :*

$$FM(\mathcal{C},\mathcal{C}') = \sqrt{prec(\mathcal{C},\mathcal{C}') \times rapp(\mathcal{C},\mathcal{C}')} \tag{3.13}$$

Définition 3.2.5 (F-mesure) *La F-mesure est définie comme la moyenne harmonique de l'indice de précision et du coefficient de rappel :*

$$F(\mathcal{C},\mathcal{C}') = \frac{2 \times prec(\mathcal{C},\mathcal{C}') \times rapp(\mathcal{C},\mathcal{C}')}{prec(\mathcal{C},\mathcal{C}') + rapp(\mathcal{C},\mathcal{C}')} \tag{3.14}$$

En utilisant une moyenne harmonique pondérée, on définit la F_α-mesure qui généralise la F-mesure de la manière suivante :

$$F_\alpha(\mathcal{C},\mathcal{C}') = \frac{(1+\alpha) \times prec(\mathcal{C},\mathcal{C}') \times rapp(\mathcal{C},\mathcal{C}')}{\alpha \times prec(\mathcal{C},\mathcal{C}') + rapp(\mathcal{C},\mathcal{C}')} \tag{3.15}$$

où α est un coefficient de pondération strictement positif dont les valeurs les plus couramment utilisées sont 1, $\frac{1}{2}$ et 2. Notons que pour tout $\alpha \neq 1$, la F_α-mesure est asymétrique.

3.2.2 Indice de Rand & Métrique de Mirkin

L'indice de Rand, qui indique la proportion des paires d'objets pour lesquelles deux partitions sont concordantes, correspond à la mesure de similarité binaire "simple correspondance" et prend ainsi ses valeurs sur $[0;1]$.

Définition 3.2.6 (Indice de Rand) *L'indice de Rand qui prend ses valeurs sur l'intervalle $[0;1]$ est défini de la manière suivante :*

$$R(\mathcal{C},\mathcal{C}') = \frac{N_{11} + N_{00}}{N_{11} + N_{00} + N_{10} + N_{01}} \tag{3.16}$$

Il convient d'introduire ici la métrique de Mirkin qui est une forme normalisée de la mesure de dissimilarité associée à l'indice de Rand [Mei05, Mei06] :

Définition 3.2.7 (Métrique de Mirkin) *La métrique de Mirkin est définie comme le nombre d'arêtes qui n'existent que dans une seule des deux partitions :*

$$M(\mathcal{C},\mathcal{C}') = 2(N_{10} + N_{10}) \tag{3.17}$$

$$= N(N-1)\left[1 - R(\mathcal{C},\mathcal{C}')\right] \tag{3.18}$$

3.2.3 Similarité & hasard

Similarité due au hasard

Comme cela a été souligné par de nombreux auteurs [FM83, HA85, Mei06], les valeurs prises par les différents indices présentés au début de cette section ne prennent généralement pas toutes les valeurs de l'intervalle $[0;1]$ et une part de la similarité entre deux partitions peut être attribuée au hasard. Il est néanmoins possible de corriger la valeur d'un indice pour éliminer la part de similarité due au hasard :

$$indice = \frac{indice - E[indice]}{1 - E[indice]} \tag{3.19}$$

où $E[indice]$ désigne son espérance sous l'hypothèse d'indépendance des partitions comparées. On suppose alors que les deux partitions sont obtenues de façon indépendante et qu'elles sont choisies aléatoirement parmi l'ensemble des partitions respectant les sommes marginales $n_{i.}$ et $n_{.j}$ du tableau de contingence 3.2. Outre le fait que cette normalisation peut conduire théoriquement à des valeurs négatives de l'indice normalisé, la vraisemblance de l'hypothèse utilisée peut être remise en cause [Mei06, Wal83]. En effet, la plupart des algorithmes de classification supposent le nombre de classes connu mais ne permettent d'en spécifier les effectifs. Soulignons par ailleurs que dans le cadre d'une démarche exploratoire, il semblerait bien peu naturel de devoir indiquer la répartition des effectifs dans les différents groupes d'objets.

Test de Mc Nemar

Le test de Mc Nemar est un test non paramétrique qui peut être utilisé pour comparer l'égalité de deux proportions dans des échantillons appariés [You04]. En l'adaptant à l'ensemble des accords et désaccords entre deux partitions, on peut vérifier l'hypothèse nulle que les désaccords entre ces dernières sont le fruit de regroupements ou de séparations d'objets dûs au hasard. On obtient un nouveau critère de comparaison de partitions.

Définition 3.2.8 (Test de Mc Nemar) *Étant données deux partitions $\mathcal{C}$ et $\mathcal{C}'$, la statistique de Mc Nemar suit approximativement une loi normale sous l'hypothèse nulle et est définie ainsi*

$$MN = \frac{N_{10} - N_{01}}{\sqrt{N_{10} + N_{01}}} \tag{3.20}$$

En prenant la valeur absolue de la statistique, on obtient un critère de comparaison positif et symétrique dont la nullité ne doit pas être interprétée comme l'égalité des partitions.

3.3 Comparaison par mise en correspondance d'ensembles

La section précédente était consacrée à la présentation de mesures de similarité basées sur un comptage des paires d'objets regroupés (ou séparés) en accord ou en désaccord entre deux partitions. On s'intéresse dans cette section à une famille plus riche de mesures définies à partir du tableau de contingence $C = (n_{ij})$ où n_{ij} est le nombre d'objets qui appartiennent simultanément à la classe P_i et à la classe P'_j. On notera respectivement $n_{i.}$ et $n_{.j}$ les marges de la ligne i et de la colonne j. La figure 3.2 illustre notre propos et les équations (3.5) à (3.8) montrent comment calculer les indices de la section précédente à partir du tableau de contingence.

3.3.1 Critère de Larsen

Définition 3.3.1 (Critère de Larsen) *Pour chaque classe P_i, on recherche la classe P'_j qui maximise la moyenne harmonique de la part respective des objets de $P_i \cap P'_j$ dans les classes P_i et P'_j. Le critère de Larsen s'exprime alors comme la moyenne arithmétique de ces moyennes harmoniques maximales :*

$$L(P, P') = \frac{1}{K} \sum_{i=1}^{K} \max_{j=1,\ldots,K'} \frac{2n_{ij}}{n_{i.} + n_{.j}} \tag{3.21}$$

L'assymétrie du critère proposé par Larsen n'est pas sans poser de problème [Mei06]. Considérons la situation où la partition P comporte pour seule partie l'ensemble Ω de tous les objets et où la partition P' est obtenue à partir de P en séparant de Ω deux petites parties comportant chacune $N.f$ objets, avec $0 < f \ll \frac{1}{2}$. On obtient alors :

$$\begin{aligned} L(P, P') &= \frac{1-2f}{1-f} \\ &> (1-2f) \end{aligned}$$

ce qui apparaît raisonnable, en revanche :

$$\begin{aligned} L(P', P) &= \frac{1}{3} \times \frac{1+2f}{1-f} \\ \lim_{f \to 0} L(P, P') &= \frac{1}{3} \end{aligned}$$

Contre toute attente, le critère de Larsen converge vers $\frac{1}{3}$ lorsque f tend à s'annuler et que les partitions P et P' nous paraissent intuitivement de plus en plus semblables.

3.3.2 Critère de Meilă & Heckerman

Définition 3.3.2 (Critère de Meilă & Heckerman) *Le critère de Meilă & Heckerman est une métrique qui repose sur le taux d'erreurs de classement commises par une partition P' relativement à une partition P [Mei05, Mei06]. Il s'exprime ainsi :*

$$H = 1 - \frac{1}{N} \max_{\pi : P \to P'} \sum_{i=1}^{K} n_{i\pi(i)} \tag{3.22}$$

où π est une injection de P dans P'.

Signalons que la recherche de la mise en correspondance optimale des classes de P et de P' ne nécessite pas d'énumérer toutes les injections possibles ($\min\{K!, K'!\}$) mais qu'elle peut être calculée en temps polynomial [Mei05].

3.3.3 van Dongen

Définition 3.3.3 (Critère de van Dongen)

$$D(P, P') = 2N - \sum_{i=1}^{K} \max_{j=1,\ldots,K'} n_{ij} - \sum_{j=1}^{K'} \max_{i=1,\ldots,K} n_{ij} \tag{3.23}$$

3.3.4 Indice de Wemmert & Gançarski

Définition 3.3.4 (Coefficient de répartition) *Intuitivement, le coefficient de répartition ρ_i mesure la propension des objets de la classe P_i à se regrouper dans la partition P'. Plus formellement, il est défini comme la somme des parts de la classe P_i présentes dans la classe P'_j au carré :*

$$\rho_i^{P'} = \sum_{j=1}^{K'} \left(\frac{n_{ij}}{n_{i.}} \right)^2 \tag{3.24}$$

Le coefficient de répartition prend ses valeurs sur l'intervalle $[\frac{1}{K'};1]$; la valeur minimale est atteinte[2] lorsque les objets de la classe P_i sont répartis uniformément dans les différentes classes de la partition P' et une valeur de 1 indique que les objets de P_i sont regroupés au sein d'une même classe de la partition P'.

Définition 3.3.5 (Critère local de similitude) *Le critère local de similitude d'une classe P_i dans une partition P' évalue si P_i est similaire à l'une des classes de P'. Il est défini de la manière suivante :*

$$wg_i^{P'} = \rho_i^{P'} \times \max_{j=1,\ldots,K'} \left(\frac{n_{ij}}{n_{i.}} \right) \tag{3.25}$$

Le critère local de similitude prend ses valeurs sur l'intervalle $[\left(\frac{1}{K'}\right)^2;1]$; il prend sa valeur minimale[3] lorsque les objets de la classe P_i sont répartis uniformément dans les différentes classes de la partition P' et la valeur 1 si une des classes de la partition P' est égale à P_i.

Définition 3.3.6 (Indice de Wemmert & Gançarski) *L'indice de Wemmert & Gançarski est la moyenne des critères locaux de similitude des classes de la partition P dans la partition P' et des classes de P' dans la partition P ; il s'exprime ainsi :*

$$WG(P,P') = \frac{1}{2}\left(\frac{1}{K} \sum_{i=1}^{K} wg_i^{P'} + \frac{1}{K'} \sum_{j=1}^{K'} wg_j^{P} \right) \tag{3.26}$$

L'indice de Wemmert & Gançarski prend ses valeurs sur l'intervalle $[\frac{K^2+K'^2}{2K^2K'^2};1]$. La valeur minimale est atteinte lorsque les classes de P se répartissent uniformément dans P' et réciproquement ; cette situation est illustrée par la figure 3.4. La valeur de 1 est atteinte si les deux partitions sont identiques.

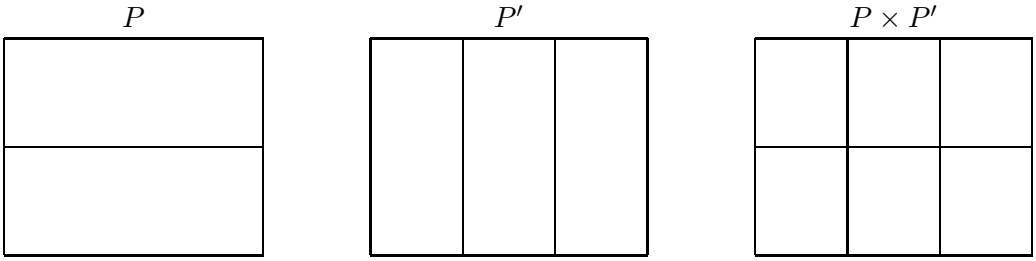

Figure 3.4 – Cas de deux partitions dont les classes se répartissent uniformément l'une dans l'autre.

3.4 Propriétés souhaitables

Bien que l'espace des partitions d'un ensemble fini d'exemples soit fini, sa cardinalité est super exponentielle et sa structure est suffisamment complexe pour défier notre intuition. Marina Meilă propose différentes propriétés qui permettent de rendre plus intuitive une mesure de dissimilarité entre partitions notamment en l'alignant sur la structure de treillis définie à la section précédente [Mei05, Mei06].

[2] On suppose ici que le nombre d'objets $n_{.i}$ de chaque classe P_i est un multiple de K'.

[3] On suppose ici que le nombre d'objets $n_{.i}$ de chaque classe P_i est un multiple de K' et que le nombre d'objets $n_{j.}$ de chaque classe P'_j est un multiple de K.

Il est communément admis que l'esprit humain est généralement plus familier avec une métrique qu'avec une fonction quelconque de deux variables. Les propriétés d'une métrique, et tout particulièrement la symétrie et l'inégalité triangulaire, facilitent alors sa perception. Ensuite, l'inégalité triangulaire nous indique que deux éléments de l'espace proches d'un troisième ne peuvent pas être très éloignés l'un de l'autre. Cette particularité est intéressante pour concevoir des structures de données et des algorithmes efficaces. Enfin, cette propriété nous ne limite plus à la comparaison de deux classifications mais permet d'envisager une analyse fine d'un ensemble plus important de classifications.

Ensuite, dans l'optique de comparer les résultats obtenus sur différents jeux de données, par un ou plusieurs algorithmes de classification[4], il est nécessaire de disposer d'un critère dont la valeur ne dépend pas du nombre d'objets présents dans l'ensemble à partitionner. Cette considération nous amène à définir la propriété suivante :

Définition 3.4.1 (N-invariance) *Un critère d est N-invariant si sa valeur ne dépend pas directement du nombre total d'objets.*

Définissons maintenant trois propriétés d'additivité par rapport aux différentes opérations disponibles sur les partitions :

Définition 3.4.2 (Additivité par raffinement) *On dit qu'un critère d respecte la propriété d'additivité par raffinement si et seulement si pour toutes partitions $\mathcal{C}$, $\mathcal{C}'$ et $\mathcal{C}''$ telles que $\mathcal{C}'' \prec \mathcal{C}'$ et $\mathcal{C}' \prec \mathcal{C}$, on a :*

$$d(\mathcal{C}, \mathcal{C}'') = d(\mathcal{C}, \mathcal{C}') + d(\mathcal{C}', \mathcal{C}'') \tag{3.27}$$

Définition 3.4.3 (Additivité par jointure) *On dit qu'un critère d respecte la propriété d'additivité par jointure si et seulement si pour toutes partitions $\mathcal{C}$ et $\mathcal{C}'$ on a :*

$$d(\mathcal{C}, \mathcal{C}') = d(\mathcal{C}, \mathcal{C} \times \mathcal{C}') + d(\mathcal{C}', \mathcal{C} \times \mathcal{C}') \tag{3.28}$$

Définition 3.4.4 (Additivité par composition) *On dit qu'un critère d respecte la propriété d'additivité par composition si et seulement si pour toutes partitions $\mathcal{C}$ et $\mathcal{C}'$ on a :*

$$d(\mathcal{C}, \mathcal{C}') = \sum_{k=1}^{K} \frac{n_k}{N} \, d(\mathcal{C}_k, \mathcal{C}_k \times \mathcal{C}') \tag{3.29}$$

où n_k est le cardinal de la classe $\mathcal{C}_k$ et N est le nombre total d'objets.

Les trois propriétés d'additivité définies ci-dessus permettent calculer les critères entre différentes partitions prises deux à deux de manière incrémentale, ceci peut être particulièrement intéressant lorsque son calcul est coûteux et qu'on souhaite étudier le parcours d'un algorithme dans l'espace des partitions. Au delà de cet aspect purement calculatoire, ces propriétés s'appuient sur la structure de treillis et facilitent ainsi la compréhension du critère de comparaison et de la structure qu'il engendre.

3.5 Variation d'information

Un nouveau critère de comparaison de partitions issu de la théorie de l'information est proposé dans [Mei03, Mei05, Mei06] : la Variation d'Information (VI). La VI quantifie l'information que la connaissance d'une partition apporte sur une autre.

[4] en comparant par exemple le résultat à une partition de référence

3.5.1 Définitions

Définition 3.5.1 (Entropie associée à une partition) *L'entropie associée à une partition $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ mesure l'incertitude de la variable aléatoire X dont la valeur correspond à l'indice de la classe d'un objet prélevé aléatoirement dans l'ensemble Ω. Elle est définie ainsi :*

$$H(\mathcal{C}) = -\sum_{i=1}^{K} P(X = i) \log_2 P(X = i) \tag{3.30}$$

Notons que l'entropie d'une partition est toujours positive et prend la valeur 0 lorsqu'il n'y a aucune incertitude quant à l'appartenance d'un objet à une classe ; ce cas de figure se présente lorsque $\mathcal{C} = \hat{1}$. La valeur maximale de 1 est atteinte lorsque les objets se répartissent de manière uniforme dans deux classes différentes.

Définition 3.5.2 (Information mutuelle entre deux partitions) *L'information mutuelle entre deux partitions $\mathcal{C}$ et $\mathcal{C}'$ quantifie l'information apportée par la variable aléatoire X associée à $\mathcal{C}$ sur la variable aléatoire X' associée à $\mathcal{C}'$ et réciproquement. Elle se définit de la manière suivante :*

$$I(\mathcal{C}, \mathcal{C}') = \sum_{i=1}^{K} \sum_{i'=1}^{K'} P(X = i, X' = i') \log_2 \frac{P(X = i, X' = i')}{P(X = i) P(X' = i')} \tag{3.31}$$

D'après la définition ci-dessus, l'information mutuelle entre deux partitions est symétrique et toujours positive. Ajoutons qu'elle ne peut en aucun cas dépasser l'entropie de l'une ou l'autre des partitions.

$$I(\mathcal{C}, \mathcal{C}') \geq 0 \tag{3.32}$$
$$I(\mathcal{C}, \mathcal{C}') = I(\mathcal{C}', \mathcal{C}) \tag{3.33}$$
$$I(\mathcal{C}, \mathcal{C}') \leq \min\{H(\mathcal{C}), H(\mathcal{C}')\} \tag{3.34}$$

Notons que lorsque $\mathcal{C}' \preccurlyeq \mathcal{C}$ on a alors

$$I(\mathcal{C}, \mathcal{C}') = H(\mathcal{C}') \leq H(\mathcal{C}')$$

Ainsi, lorsque deux partitions $\mathcal{C}$ et $\mathcal{C}'$ sont égales, on a : $I(\mathcal{C}, \mathcal{C}') = H(\mathcal{C}') = H(\mathcal{C}')$

Définition 3.5.3 (Variation d'Information) *La variation d'information entre $\mathcal{C}$ et $\mathcal{C}'$ peut être vue comme la somme de l'information sur $\mathcal{C}$ que l'on perd et de l'information sur $\mathcal{C}'$ que l'on gagne lorsqu'on passe de la partition $\mathcal{C}$ à la partition $\mathcal{C}'$. Ceci est formulé de manière équivalente par les différentes expressions suivantes :*

$$VI(\mathcal{C}, \mathcal{C}') = H(\mathcal{C}|\mathcal{C}') + H(\mathcal{C}'|\mathcal{C}) \tag{3.35}$$
$$= [H(\mathcal{C}) - I(\mathcal{C}, \mathcal{C}')] + [H(\mathcal{C}') - I(\mathcal{C}, \mathcal{C}')] \tag{3.36}$$
$$= H(\mathcal{C}) + H(\mathcal{C}') - 2I(\mathcal{C}, \mathcal{C}') \tag{3.37}$$

3.5.2 Propriétés

Une métrique sur l'ensemble des partitions

Propriété 3.5.1 *La variation d'information est une métrique sur l'ensemble des partitions ; ainsi, pour toutes partitions $\mathcal{C}$, $\mathcal{C}'$ et $\mathcal{C}''$, elle présente les propriétés suivantes*

1. *Positivité :* $VI(\mathcal{C},\mathcal{C}')$ *est toujours positif*
2. *Séparabilité :* $VI(\mathcal{C},\mathcal{C}')$ *s'annule si et seulement si les deux partitions sont égales.*
3. *Symétrie :* $VI(\mathcal{C},\mathcal{C}') = VI(\mathcal{C}',\mathcal{C})$.
4. *Inégalité triangulaire :* $VI(\mathcal{C},\mathcal{C}') + VI(\mathcal{C}',\mathcal{C}'') \geq VI(\mathcal{C},\mathcal{C}'')$

Bornes supérieures

Propriété 3.5.2 *La valeur de* $VI(\mathcal{C},\mathcal{C}')$ *ne dépend que des tailles relatives des classes et non du nombre total d'objets.*

Propriété 3.5.3 *La borne supérieure suivante est atteinte quel que soit le nombre total d'objets* N *:*

$$V(\mathcal{C},\mathcal{C}') \leq \log N \tag{3.38}$$

Propriété 3.5.4 *Si* $\mathcal{C}$ *et* $\mathcal{C}'$ *sont deux partitions formées d'au plus* $K^\star$ *classes chacune, avec* $K^\star \leq \sqrt{N}$*, alors :*

$$VI(\mathcal{C},\mathcal{C}') = 2\log K^\star \tag{3.39}$$

Le voisinage local induit

Propriété 3.5.5 (Partage d'une classe) *La variation d'information entre une partition* $\mathcal{C}$ *et la partition* $\mathcal{C}'$ *obtenue en partageant la classe* $\mathcal{C}_i$ *en sous-groupes* $\mathcal{C}'_{i_1},\ldots,\mathcal{C}'_{i_k}$ *est égale à :*

$$VI(\mathcal{C},\mathcal{C}') = P(X=i)H_{|i} \tag{3.40}$$

où X *est la variable aléatoire associée à la partition* $\mathcal{C}$ *et où* $H_{|i}$ *est l'entropie associée à la partition* $\mathcal{C}'_{i_1},\ldots,\mathcal{C}'_{i_k}$ *de la classe* $\mathcal{C}_i$*.*

Corollaire 3.5.1 *D'après la propriété 3.5.5, nous avons :*

1. *Si la partition* $\mathcal{C}'$ *est obtenue en partageant la classe* $\mathcal{C}_i$ *en* k *sous-groupes de même taille, alors :*

$$VI(\mathcal{C},\mathcal{C}') = P(X=i)\log_2 k \tag{3.41}$$

   où X *est la variable aléatoire associée à la partition* $\mathcal{C}$*.*

2. *Si la partition* $\mathcal{C}'$ *est obtenue en séparant un point de la classe* $\mathcal{C}_i$ *pour former un singleton, alors*

$$VI(\mathcal{C},\mathcal{C}') = \frac{1}{N}\left[n_i\log_2 n_i - (n_i-1)\log_2(n_i-1)\right] \tag{3.42}$$

Propriété 3.5.6 (Additivité par jointure) *La variation d'information est additive par jointure, ainsi pour toute partition* $\mathcal{C}$ *et* $\mathcal{C}'$*, on a :*

$$VI(\mathcal{C},\mathcal{C}') = VI(\mathcal{C},\mathcal{C}\times\mathcal{C}') + VI(\mathcal{C}',\mathcal{C}\times\mathcal{C}') \tag{3.43}$$

Corollaire 3.5.2 *Le plus proche voisin* $\mathcal{C}'$ *d'une partition* $\mathcal{C}$ *quelconque est comparable avec celle-ci ; soit* $\mathcal{C}' \prec \mathcal{C}$ *soit* $\mathcal{C} \prec \mathcal{C}'$*.*

Corollaire 3.5.3 *Pour toutes partitions $\mathcal{C}$ et $\mathcal{C}'$,*

$$VI(\mathcal{C},\mathcal{C}') \geq VI(\mathcal{C},\mathcal{C}\times\mathcal{C}') \tag{3.44}$$

avec l'égalité si et seulement si $\mathcal{C}=\mathcal{C}'$.

Corollaire 3.5.4 *Pour toutes partitions $\mathcal{C}\neq\mathcal{C}'$,*

$$VI(\mathcal{C},\mathcal{C}') \geq \frac{2}{N} \tag{3.45}$$

avec l'égalité lorsque $\mathcal{C}'$ est obtenue en fusionnant deux classes de $\mathcal{C}$ ou l'inverse.

Propriété 3.5.7 (Additivité par composition) *Étant données trois partitions $\mathcal{C}=\{\mathcal{C}_1,\ldots,\mathcal{C}_K\}, \mathcal{C}' \prec \mathcal{C}$ et $\mathcal{C}'' \prec \mathcal{C}$, la variation d'information est additive par composition et vérifie :*

$$VI(\mathcal{C}',\mathcal{C}'') = \sum_{i=1}^{K} P(X=i) VI(\mathcal{C}_k\times\mathcal{C}', \mathcal{C}_k\times\mathcal{C}'') \tag{3.46}$$

Propriété 3.5.8 (Unicité) *La variation d'information est le seul critère de comparaison de partition d qui :*

- *est additif par composition,*
- *est additif par jointure,*
- *pour toute partition $\mathcal{C}$, vérifie $d(\hat{1},\mathcal{C})+d(\mathcal{C},\hat{0})=d(\hat{1},\hat{0})$*
- *lorsque la partition $\mathcal{C}_K^U$ avec K classes de même effectif existe, vérifie $d(\hat{1},\mathcal{C}_K^U)=\log K$.*

Remarque : La propriété qui s'énonce "pour toute partition $\mathcal{C}$, $d(\hat{1},\mathcal{C})+d(\mathcal{C},\hat{0})=d(\hat{1},\hat{0})$" peut être vue comme une version affaiblie de l'additivité par raffinement dont elle est un cas particulier. Néanmoins, on peut montrer qu'un critère qui vérifie également l'additivité par composition est additif par raffinement.

3.6 Conclusion

Au cours de ce chapitre consacré à la problématique de comparaison de partitions, nous avons introduit les notions nécessaires à l'appréhension des critères classiques qui ont été présentés. La liste des propriétés que Marina Meila a proposées comme étant intéressantes a ensuite été rappelée avant d'introduire la variation d'information qui nous semble un critère de comparaison très pertinent. Rappelons que la comparaison de partitions est à la base de nombreux critères externes d'évaluation de partitions et qu'elle permet également d'évaluer la stabilité d'un algorithme de classification automatique. Ce dernier point est particulièrement intéressant lorsqu'on utilise des techniques de rééchantillonnage pour fixer les paramètres d'un algorithme comme le nombre de classes.

CHAPITRE 4

Réduction de dimension

4.1 Introduction

La taille des données peut être mesurée selon deux dimensions, le nombre de variables et le nombre d'exemples. Ces deux dimensions peuvent prendre des valeurs très élevées, ce qui peut poser un problème lors de l'exploration et l'analyse de ces données. Pour cela, il est fondamental de mettre en place des outils de traitement de données permettant une meilleure compréhension de la valeur des connaissances disponibles dans ces données. La réduction des dimensions est l'une des plus vieilles approches permettant d'apporter des éléments de réponse à ce problème. Son objectif est de sélectionner ou d'extraire un sous-ensemble optimal de caractéristiques pertinentes pour un critère fixé auparavant. La sélection de ce sous-ensemble de caractéristiques permet d'éliminer les informations non-pertinentes et redondantes selon le critère utilisé. Cette sélection/extraction permet donc de réduire la dimension de l'espace des exemples et de rendre l'ensemble des données plus représentatif du problème. En effet, les principaux objectifs de la réduction de dimension sont :

- faciliter la visualisation et la compréhension des données,
- réduire l'espace de stockage nécessaire,
- réduire le temps d'apprentissage et d'utilisation,
- identifier les facteurs pertinents.

Les algorithmes d'apprentissage artificiel requièrent typiquement peu de traits - *features* - ou de variables - attributs - très significatives caractérisant le processus étudié. Dans le domaine de la reconnaissance des formes et de la fouille de données, il pourrait encore être bénéfique d'incorporer un module de réduction de la dimension dans le système global avec comme objectif d'enlever toute information inconséquente et redondante. Cela a un effet important sur la performance du système. En effet le nombre de caractéristiques utilisées est directement lié à l'erreur finale. L'importance de chaque caractéristique dépend de la taille de la base d'apprentissage - pour un échantillon de petite taille, l'élimination d'une caractéristique importante peut diminuer l'erreur. Il faut aussi noter que des caractéristiques individuellement peu pertinentes peuvent être très informatives si on les utilise conjointement.

La réduction de la dimension est un problème complexe qui permet de réduire le volume d'informations à traiter et de faciliter le processus de l'apprentissage.

Nous pouvons classer toutes les techniques mathématiques de réduction de dimension en deux grandes catégories :

- la sélection de variables : qui consiste à choisir des caractéristiques dans l'espace de mesure, (figure 4.1)
- et l'extraction de traits : qui vise à sélectionner des caractéristiques dans un espace transformé - dans un espace de projection (figure 4.2)

Définition 4.1.1 (Variables et Traits [Ben01]) *Nous appelons "variables" les descripteurs d'entrée et "traits" des caractéristiques construites à partir des variables d'entrée.*

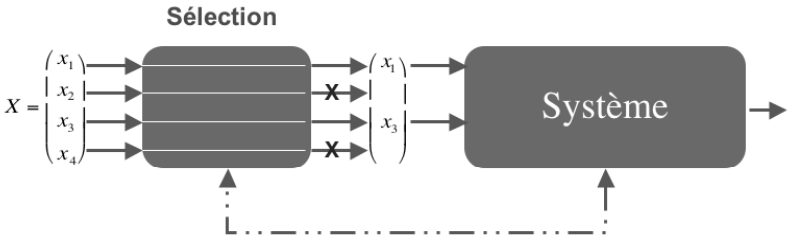

Figure 4.1 – Principe de la sélection de variables.

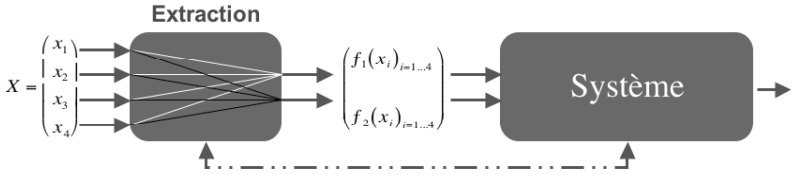

Figure 4.2 – Principe de l'extraction de caractéristiques.

La distinction est nécessaire dans le cas des méthodes à noyaux pour lesquelles les traits ne sont pas explicitement calculés.

La première catégorie est appropriée quand l'acquisition de mesures des formes est coûteuse. Ainsi l'objectif principal de la sélection de caractéristiques dans ce cas est de réduire le nombre de mesures requises. Par contre, les techniques d'extraction de traits (deuxième catégorie) utilisent toute l'information contenue dans les formes pour la compresser et produire un vecteur de plus petite dimension. Ces techniques projettent un vecteur forme de l'espace de représentation dans un espace de dimension plus petite. Les systèmes d'apprentissage connexionniste sont un bon exemple de cette catégorie. En effet, les modèles connexionnistes conçus pour une tache de discrimination fournissent un système avec des aptitudes intéressantes pour l'analyse du processus. Les cellules cachées d'un Perceptron multi-couches apprennent comment extraire les caractéristiques significatives du signal d'entrée.

4.2 Sélection de variables

La sélection de variable est un problème difficile qui a été étudié depuis les années 70. Il revient dans l'actualité scientifique avec l'apparition des grandes bases de données et les systèmes de fouille de données «Data Mining» [LM98, CB02, GGNZar].

La sélection de variables a fait l'objet de plusieurs recherches en statistique, et plus particulièrement dans des domaines comme la reconnaissance des formes, la modélisation de séries chronologiques et l'identification de processus. Dans le domaine de l'apprentissage, l'étude de la problématique de la sélection de variables est assez récente. En apprentissage symbolique, de nombreuses méthodes ont été proposées pour des tâches de classement - discrimination. Dans le domaine de l'apprentissage connexionniste [Ben01, Ben06], la sélection de variables a été abordée à partir d'un problème d'optimisation et de choix d'architectures des modèles, ainsi des approches très intéressantes ont émergé.

La sélection de variables est une problématique complexe et d'une importance cruciale pour les systèmes d'apprentissage. Afin de mettre en évidence les deux aspects du processus de la sélection de variables, difficulté et importance, nous allons présenter les éléments essentiels que nécessite généralement

ce processus. Une définition de la sélection de variables peut s'énoncer de la façon suivante :

Définition 4.2.1 (Sélection de variables [Ben01]) *La sélection de variables est un procédé permettant de choisir un sous-ensemble optimal de variables pertinentes, à partir d'un ensemble de variables original, selon un certain critère de performance.*

A partir de cette définition, on peut se poser trois questions essentielles :

- Comment mesurer la pertinence des variables ?
- Comment former le sous-ensemble optimal ?
- Quel critère d'optimalité utiliser ?

Ces trois questions définissent les éléments essentiels d'une procédure de sélection de variables. En effet, le problème de la sélection de variables consiste à identifier les variables permettant une meilleure séparation entre les différentes classes dans le cas d'un classement et une meilleure qualité de prédiction dans le cas d'une régression. On parle alors de "pouvoir discriminant" dans le premier cas et de "pouvoir prédictif" dans le deuxième cas, pour désigner la pertinence d'une variable. La réponse à la première question consiste à trouver une mesure de pertinence ou un **critère d'évaluation** $J(X)$ permettant de quantifier l'importance d'une variable ou d'un ensemble de variables X. La deuxième question évoque le problème du choix de la **procédure de recherche** ou de constitution du sous-ensemble optimal des variables pertinentes. La dernière question demande la définition d'un critère d'arrêt de la recherche. Le **critère d'arrêt** est généralement déterminé à travers une combinaison particulière entre la mesure de pertinence et la procédure de recherche.

4.2.1 Critères d'évaluation

L'amélioration des performances d'un système d'apprentissage par une procédure de sélection de variables nécessite dans un premier temps la définition d'une mesure de pertinence. Dans le cas d'un problème de classement, on teste, par exemple, la qualité de discrimination du système en présence ou en absence d'une variable. Par contre, pour un problème de régression, on teste plutôt la qualité de prédiction par rapport aux autres variables.

Commençons d'abord par définir ce qui est la pertinence d'une variable (ou d'un ensemble de variables).

Définition 4.2.2 (Pertinence d'une variable [Ben01]) *Une variable pertinente est une variable telle que sa suppression entraîne une détérioration des performances - pouvoir de discrimination en classement ou la qualité de prédiction en régression - du système d'apprentissage.*

Plusieurs critères d'évaluation ont été proposés, basés sur des hypothèses statistiques ou sur des heuristiques. Pour un problème de classement - discrimination -, les critères d'évaluation sont souvent basés sur les matrices de dispersion intra et inter classes. En effet, ces matrices sont directement liées à la géométrie des classes et donnent une information significative sur la répartition des classes dans l'espace des formes.

On trouve aussi des critères d'évaluation qui utilisent des distances probabilistes ou des mesures d'entropie. Le critère dans ce cas est basé sur l'information mutuelle entre le classement et l'ensemble de variables. Dans le cas des systèmes d'apprentissage connexionnistes, l'évaluation des variables se fait en fonction de l'importance des poids qui est définie comme le changement de l'erreur - de classement ou de régression - dû à la suppression de ces poids.

4.2.2 Procédures de recherche

En général, on ne connaît pas le nombre optimal m de variables à sélectionner. Ce nombre dépendra de la taille et de la qualité de la base d'apprentissage - la quantité et la qualité d'information disponible - et de la règle de décision utilisée - le modèle. Pour un ensemble de n variables il existe $(2^n - 1)$ combinaisons de variables possibles où 2 représente deux choix : sélectionner ou ne pas sélectionner une variable. La recherche d'un sous-ensemble de m variables parmi n engendre un nombre de combinaison égal à :

$$\binom{n}{m} = \frac{n!}{(n-m)!\, m!} \tag{4.1}$$

En grande dimension - très grande -, le nombre de combinaison à examiner devient très élevé et une recherche exhaustive n'est pas envisageable. La recherche d'un sous-ensemble optimal de variables est un problème NP-difficile. Une alternative consiste à utiliser une méthode de recherche de type Branch & Bound, [LM98]. Cette méthode de recherche permet de restreindre la recherche et donne le sous-ensemble optimal de variables, sous l'hypothèse de monotocité du critère de sélection J(X). Le critère $J(X)$ est dit monotone si :

$$X_i \subset X_2 \subset \ldots \subset X_m : J(X_1) < J(X_2) < \ldots < J(X_m) \tag{4.2}$$

où X_k est l'ensemble contenant k variables sélectionnées.

Cependant, la plupart des critères d'évaluation utilisés pour la sélection ne sont pas monotones et dans ce cas on a recours à la seule alternative basée sur des méthodes sous-optimales comme les procédures séquentielles :

- Stratégie ascendante : *Forward Selection (FS)*,
- Stratégie descendante : *Backward Selection (BS)*,
- Stratégie bidirectionnelle : *Bidirectional Selection (BiS)*.

La méthode *FS* procède par agrégations successives - par ajouts successifs de variables. Au départ l'ensemble des variables sélectionnées est initialisé à l'ensemble vide. À chaque étape k, on sélectionne la variable qui optimise le critère d'évaluation $J(X_k)$ et on la rajoute à l'ensemble des variables sélectionnées X_k. Soit X l'ensemble des variables, on sélectionne la variable x_i telle que :

$$J\left(X_k\right) = \max_{x_i \in X \setminus X_{k-1}} J\left(X_{k-1} \setminus \{x_i\}\right) \tag{4.3}$$

L'ordre d'adjonction des variables à l'ensemble des variables sélectionnées produit une liste ordonnée des variables selon leur importance. Les variables les plus importantes sont les premières variables ajoutées à la liste. Néanmoins, il faut aussi se rappeler que des variables individuellement peu pertinentes peuvent être très informatives si on les utilise conjointement.

La méthode *BS* est une procédure inverse de la précédente - par retraits successifs de variables. On part de l'ensemble X complet des variables et on procède par élimination. À chaque étape la variable la moins importante selon le critère d'évaluation est éliminée. Le procédé continue jusqu'à ce qu'il reste qu'une seule variable dans l'ensemble des variables de départ. À l'étape k, on supprime la variable x_i telle que :

$$J\left(X_k\right) = max_{x_i \in X_{k+1}} \setminus \{x_i\} \tag{4.4}$$

Une liste ordonnée selon l'ordre d'élimination des variables est ainsi obtenue. Les variables les plus pertinentes sont alors les variables qui se trouvent dans les dernières positions de la liste.

La procédure *BiS* effectue sa recherche dans les deux directions - *Forward* et *Backward* - d'une manière concurrentielle. La procédure s'arrête dans deux cas : (1) quand une des deux directions a trouvé

le meilleur sous-ensemble de variables avant d'atteindre le milieu de l'espace de recherche ; ou (2) quand les deux directions arrivent au milieu. Il est clair que les ensembles de variables sélectionnées trouvés respectivement par *FS* et par *BS* ne sont pas égaux à cause de leurs différents principes de sélection. Néanmoins, cette méthode réduit le temps de recherche puisque la recherche s'effectue dans les deux directions et s'arrête dès qu'il y a une solution quelle que soit la direction.

4.2.3 Critères d'arrêt

Le nombre optimal de variables n'est pas connu a priori, l'utilisation d'une règle pour contrôler la sélection-élimination de variables permet d'arrêter la recherche lorsque aucune variable n'est plus suffisamment informative. Le critère d'arrêt est souvent défini comme une combinaison de la procédure de recherche et du critère d'évaluation. Une heuristique, souvent utilisée, consiste à calculer pour les différents sous-ensembles de variables sélectionnées une estimation de l'erreur de généralisation par validation croisée. Le sous-ensemble de variables sélectionnées est celui qui minimise cette erreur de généralisation. Les différentes approches de sélection Il existe trois grandes familles d'approches :

Approches "Filtres" - *Filters* : ces méthodes sélectionnent les variables indépendamment de la méthode qui va les utiliser, elles se basent sur les caractéristiques de l'ensemble des données afin de sélectionner certaines variables et d'éliminer d'autres sous forme de pré-traitement des données.

Approches "Symbioses" - *Wrappers* : contrairement aux approches filtre qui ignorent totalement l'influence des variables sélectionnées sur la performance de l'algorithme d'apprentissage, les approches "enveloppantes" utilisent l'algorithme d'apprentissage comme une fonction d'évaluation.

Approches "Intégrées" - *Embedded* : ces méthodes exécutent la sélection variable pendant le processus de l'apprentissage. Le processus de la sélection de variables est effectué parallèlement au processus de classement - ou de la régression. Le sous-ensemble de variables ainsi sélectionnées sera choisi de façon à optimiser le critère d'apprentissage utilisé.

4.2.4 Sélection de variables et apprentissage connexionniste

La sélection de variables dans le domaine connexionniste est très attrayante et soulève de nombreux enjeux à la fois théoriques et applicatifs fondamentaux [Ben01, Ben06]. En effet, dans le cas des réseaux connexionnistes, le processus de la sélection de variables peut être effectué parallèlement au processus de classement - ou de la régression. Le sous-ensemble de variables ainsi sélectionnées sera choisi de façon à optimiser le critère d'apprentissage. En plus, le nombre de variables est directement lié à l'architecture et à la complexité de la fonction réalisable par le système connexionniste.

Dans le cas des systèmes d'apprentissage connexionniste, le nombre de variables est directement lié à l'architecture et à la complexité de la fonction réalisable par le modèle connexionniste. Plusieurs approches ont été proposées dans la littérature. La plupart de ces techniques emploient la première ou la deuxième dérivée de la fonction de coût par rapport aux poids pour estimer l'importance des connexions.

Les méthodes les plus largement employées sont : *Optimal Brain Damage (OBD)* proposée par Le Cun et al. [LCDS90], et *Optimal Brain Surgeon (OBS)* [HS93] par Hassibi et Stork qui est une amélioration de la précédente. Pedersen et al. ont proposé γOBD et γOBS [PHL96], où l'estimation de l'importance d'un poids est basée sur le changement associé dans l'erreur de généralisation si le poids est élagué. D'autres variantes d'*OBD* et d'*OBS* ont été proposées : *Early Brain Damage (EBD)* et *Early Brain Surgeon (EBS)* [TNZ96]. On peut citer aussi *Optimal Cell Damage (OCD)* développée par Cibas et al. dans [CFGR94] qui est une extension de *OBD* pour l'élagage des variables d'entrée. Ces méthodes se basent sur l'estimation systématique de l'importance d'une connexion qui est définie comme le changement de

l'erreur causé par la suppression de ce poids. L'emploi des dérivées premières pour la sélection de variables peut être trouvé par exemple dans [DPJ+96, Moo94, RRK90]. D'autres méthodes de sélection de variables utilisent les paramètres du système d'apprentissage. Certaines de ces méthodes emploient : des tests statistiques pour évaluer un intervalle de confiance pour chaque poids [CGG+95], l'information mutuelle pour évaluer un ensemble de caractéristiques et sélectionner un sous-ensemble pertinent [Bat94], des mesures heuristiques basées sur l'estimation de la contribution des variables dans la prise de décision du système [BB95, YB97]. Dans le cadre de l'apprentissage bayésien MacKay et Neal proposent une méthode de sélection de variables *Automatic Relevance Determination (ARD)* [Mac94, Nea94]. Cette méthode utilise des hypothèses de normalité sur la répartition des poids du réseau.

Dans les paragraphes qui suivent, nous allons détailler quelques méthodes en les regroupant par type.

Les méthodes connexionnistes de sélection de variables sont en général de type "backward". L'idée générale est de faire converger un réseau jusqu'à un minimum local en utilisant toutes les variables et de faire ensuite la sélection. L'étape de sélection consiste à trier les variables par ordre croissant de pertinence, supprimer la ou les variables les moins pertinentes et ré-entraîné le réseau avec les variables restantes. Ce processus continue tant qu'un certain critère d'arrêt n'est pas satisfait. Les méthodes qui suivent cette procédure comportent donc deux phases : une phase d'apprentissage et une phase d'élagage qui peuvent être alternées. On peut dire qu'une "vraie" procédure connexionniste de sélection de variables suit l'algorithme général suivant :

1. Atteindre un minimum local
2. Calculer la pertinence de chaque entrée
3. Trier les entrées par ordre croissant de pertinence
4. Supprimer les entrées dont la pertinence cumulée est inférieure à un seuil fixé
5. Recommencer en 1. Tant que les performances estimées sur une base de validation ne chutent pas

Les méthodes de sélection de variables en apprentissage connexionniste peuvent se regrouper en trois grandes familles :

- Les méthodes d'ordre zéro
- Les méthodes du premier ordre
- Les méthodes du second ordre

4.2.4.1 Méthodes d'ordre zéro

Pour estimer la pertinence d'une variable, les mesures d'ordre zéro utilisent les valeurs des paramètres du système d'apprentissage (les valeurs des connexions, la structure, ...). Par exemple la mesure de pertinence HVS [YB97] repose sur les paramètres et la structure du réseau connexionniste. Dans le cas d'un Perceptron multicouches à une seule couche cachée, cette mesure est définie par :

$$\begin{cases} \text{pertinence d'une variable} & \zeta_i = \sum\limits_{j \in Hidden} \left[\dfrac{|\omega_{ji}|}{\sum\limits_{i' \in Input} |\omega_{ji'}|} \times \sum\limits_{k \in Output} \dfrac{|\omega_{kj}|}{\sum\limits_{j' \in Hidden} |\omega_{kj'}|} \right] \\ \text{critère d'évaluation} & J(X_k) = \sum\limits_{x_i \in X_k} \zeta_i \\ \text{procédure de recherche} & \textit{Backward} + \text{réapprentissage} \\ \text{critère d'arrêt} & \text{test statistique} \end{cases}$$

Une autre méthode d'ordre zéro très efficace a été proposée par [Mac94] : *Automatic Relevance Determination (ARD)*. Dans cette méthode la pertinence d'une variable est estimée par la variance de ses poids : la variable est éliminée si la variance correspondante est faible.

4.2.4.2 Méthodes du premier ordre

La dérivée de la fonction ψ que représente un système d'apprentissage connexionniste - un réseau - par rapport à chacune de ses variables est très utilisée comme mesure de pertinence des variables. Si une dérivée est proche de zéro pour tous les exemples, alors la variable correspondante n'est pas utilisée par le réseau, et peut donc être supprimé.

Dans le cas des PMC - Perceptrons multicouches -, cette dérivée peut se calculer comme une extension de l'algorithme d'apprentissage. Comme ces dérivées peuvent prendre aussi bien des valeurs positives que négatives, produisant une moyenne proche de zéro, c'est la moyenne des valeurs absolues qui est généralement utilisée - ce sont les grandeurs des dérivées qui nous intéressent. On trouve beaucoup de mesures de pertinences basées sur cette approche.

La sensibilité de l'erreur à la suppression de chaque variable est utilisée par Moody dans [Moo94]. Une mesure de sensibilité est calculée pour chaque variable x_i pour évaluer la variation de l'erreur en apprentissage si cette variable est supprimée du réseau. Le remplacement d'une variable par sa moyenne supprime son influence sur la sortie du réseau. La définition de la pertinence est :

$$\begin{aligned} \zeta_i &= R(\omega) - \tilde{R}(\overline{x}_i, \omega) \\ \text{avec } \tilde{R}(\overline{x}_i, \omega) &= \frac{1}{N} \sum_{k=1}^{N} \left\| y^k - \psi(x_1^k, \ldots, \overline{x}_i^k, \ldots, x_n^k) \right\|^2 \end{aligned}$$

N est la taille de la base d'apprentissage. Quand cette taille est très grande, Moody propose d'utiliser une approximation qui donne la méthode de sélection suivante :

$$\begin{cases} \text{pertinence d'une variable} & \zeta_i \overset{N\to\infty}{\cong} \frac{1}{N} \sum_{k=1}^{N} \left(x_i^k - \overline{x}_i\right) \left(y^k - \psi(x^k, \omega)\right) \frac{\partial \psi(x^k, \omega)}{\partial x_i} \\ \text{critère d'évaluation} & J(X_k) = \sum_{x_i \in X_k} \zeta_i \\ \text{procédure de recherche} & \textit{Backward} \\ \text{critère d'arrêt} & \text{variation des performances en test} \end{cases}$$

Ruck et al. [RRK90] proposent la méthode suivante :

$$\begin{cases} \text{pertinence d'une variable} & \zeta_i = \sum_{k=1}^{N} \sum_{j \in Output} \left| \frac{\partial \psi_j(x^k, \omega)}{\partial x_i} \right| \\ \text{critère d'évaluation} & J(X_k) = \sum_{x_i \in X_k} \zeta_i \\ \text{procédure de recherche} & \textit{Backward} \\ \text{critère d'arrêt} & \text{seuil : moyenne des pertinences} \end{cases}$$

Refenes et al. [RZ99] utilisent l'élasticité moyenne de la sortie par rapport à chaque variable :

$$\begin{cases} \text{pertinence d'une variable} & \zeta_i = \frac{1}{N}\sum_{k=1}^{N}\left|\dfrac{\partial\psi(x^k,\omega)}{\partial x_i}\times\dfrac{x_i}{\psi(x^k,\omega)}\right| \\ \text{critère d'évaluation} & J(X_k)=\sum_{x_i\in X_k}\zeta_i \\ \text{procédure de recherche} & \textit{Backward} \\ \text{critère d'arrêt} & \text{seuil : moyenne des pertinences} \end{cases}$$

Dans le cas des réseaux à fonctions radiales RBF - *Radial Basis Functions* -, Dorizzi et al. [DPJ+96] utilisent le quantile à 95% de la distribution des valeurs absolues des dérivées de chaque variable.

$$\begin{cases} \text{pertinence d'une variable} & \zeta_i = q_{.95}\left[\left|\dfrac{\partial\psi(x,\omega)}{\partial x_i}\right|\right] \\ \text{critère d'évaluation} & J(X_k)=\sum\limits_{x_i\in X_k}\zeta_i \\ \text{procédure de recherche} & \textit{Backward} \\ \text{critère d'arrêt} & \text{seuil : moyenne des pertinences} \end{cases}$$

Pour un problème de discrimination, Fabrice Rossi propose de ne considérer que les exemples qui sont près des frontières interclasses [Ros96] :

$$x^k \in frontier \equiv \left\|\nabla_{x^k}\psi(x^k,\omega)\right\| > \epsilon$$

$$\begin{cases} \text{pertinence d'une variable} & \zeta_i = \frac{1}{|Output|}\sum\limits_{x^k\in frontier}\sum\limits_{j\in Output}\dfrac{\left|\frac{\partial\psi_j(x^k,\omega)}{\partial x_i}\right|}{\left\|\frac{\partial\psi_j(x^k,\omega)}{\partial x}\right\|} \\ \text{critère d'évaluation} & J(X_k)=\sum\limits_{x_i\in X_k}\zeta_i \\ \text{procédure de recherche} & \textit{Backward} \\ \text{critère d'arrêt} & \text{seuil : moyenne des pertinences} \end{cases}$$

4.2.4.3 Méthodes du second ordre

Pour estimer la pertinence d'une variable, les méthodes du second ordre calculent la dérivée seconde de la fonction de coût par rapport aux poids. Ces mesures sont des extensions des techniques d'élagage des poids. La technique d'élagage la plus populaire est *Optimal Brain Damage (OBD)* proposée par Le Cun et al. [LCDS90]. *OBD* est basée sur l'estimation de la variation de la fonction de coût $R(w)$ lorsqu'un poids est supprimé du réseau. Cette variation peut être approximée à l'aide d'un développement

en série de Taylor :

$$\delta\tilde{R}(\omega_i) = \sum_i \frac{\partial\tilde{R}(\omega)}{\partial\omega_i}\delta\omega_i + \frac{1}{2}\sum_i\sum_j \frac{\partial^2\tilde{R}(\omega)}{\partial\omega_i\partial\omega_j}\delta\omega_i\delta\omega_j + O\left(\delta\omega^3\right)$$

Sous l'hypothèse que le réseau connexionniste a atteint un minimum local, le premier terme de droite de cette formule est nul. Pour simplifier les calculs, Le Cun et al. [LCDS90] supposent en outre que la matrice Hessienne est nulle et le coût est localement quadratique. On obtient alors la formule simplifiée suivante :

$$\begin{aligned}\delta\tilde{R}(\omega_i) &\approx \frac{1}{2}\sum_i \frac{\partial^2\tilde{R}(\omega)}{\partial\omega_i^2}\delta\omega_i^2 + O\left(\delta\omega^3\right)\\ &\approx \frac{1}{2}H_{ii}\delta\omega_i^2\end{aligned}$$

La pertinence d'une connexion est alors estimée par :

$$pertinence(\omega_i) \approx \frac{1}{2}H_{ii}\omega_i^2$$

La méthode de sélection de variables *Optimal Cell Damage (OCD)* développée par Cibas et al. dans [CFGR94] est basée sur la mesure de pertinence ci-dessus. Dans *OCD*, l'importance de chaque variable s'obtient en sommant les importances des connexions qui partent de celle-ci :

$$\begin{cases}\text{pertinence d'une variable} & \zeta_i = \frac{1}{2}\displaystyle\sum_{j\in fan-Out(i)} \frac{\partial^2\tilde{R}(w)}{\partial\omega_{ji}^2}\omega_{ji}^2\\ \text{critère d'évaluation} & J(X_k) = \displaystyle\sum_{x_i\in X_k}\zeta_i\\ \text{procédure de recherche} & \textit{Backward}\\ \text{critère d'arrêt} & \text{test statistique}\end{cases}$$

où $fan - Out(i)$ est l'ensemble des neurones qui utilisent comme entrée la sortie du neurone i.

Dans *OBD* et *OBS*, la sensibilité d'un poids ne peut être évaluée correctement qu'autour d'un minimum local de la fonction de coût. Tresp et al. [TNZ96] proposent deux extensions d'*OBD* et d'*OBS* : *Early Brain Damage (EBD)* et *Early Brain Surgeon (EBS)*. *EBD* et *EBS* peuvent être utilisées avec le "*early stopping*" comme critère d'arrêt de l'apprentissage. Dans *EBD*, par exemple, la sensibilité d'un poids est donnée par la formule suivante :

$$pertinence(\omega_i) = \frac{1}{2}\frac{\partial^2\tilde{R}(w)}{\partial\omega_{ji}^2}\,\omega_{ji}^2 - \frac{\partial\tilde{R}(w)}{\partial\omega_{ji}}\,\omega_{ji} + \frac{\left(\dfrac{\partial\tilde{R}(w)}{\partial\omega_{ji}}\right)^2}{\dfrac{\partial^2\tilde{R}(w)}{\partial\omega_{ji}^2}} \tag{4.5}$$

A partir de cette définition de pertinence et de la même façon que *OCD*, [LG] propose la méthode *ECD (Early Cell Damage)* :

$$\begin{cases} \text{pertinence d'une variable} & \zeta_i = \frac{1}{2} \sum_{j \in fan-Out(i)} \frac{\partial^2 \tilde{R}(w)}{\partial \omega_{ji}^2} \omega_{ji}^2 - \frac{\partial \tilde{R}(w)}{\partial \omega_{ji}} \omega_{ji} + \frac{\left(\frac{\partial \tilde{R}(w)}{\partial \omega_{ji}} \right)^2}{\frac{\partial^2 \tilde{R}(w)}{\partial \omega_{ji}^2}} \\ \text{critère d'évaluation} & J(X_k) = \sum_{x_i \in X_k} \zeta_i \\ \text{procédure de recherche} & \textit{Backward} \\ \text{critère d'arrêt} & \text{test statistique} \end{cases}$$

Pour cette méthode on supprime les variables une par une et on peut utiliser la technique de *early stopping* pour arrêter l'apprentissage.

4.3 Extraction de caractéristiques

Les méthodes utilisées pour l'extraction de traits sont très variées. Nous rappellerons brièvement les principes des méthodes linéaires (ACP, MDS), puis nous décrirons quelques méthodes non linéaires qui ont fait l'objet de nombreuses études depuis cinq ans. Nous nous intéressons en particulier aux méthodes utilisant des graphes, comme Isomap, LLE et leurs variantes.

On considère un espace d'observations χ, qui n'est pas nécessairement $\mathbf{R}^n$, ce qui permet de généraliser les méthodes proposées aux cas où l'on ne dispose pas d'une représentation vectorielle des données à traiter, par exemple les données structurées (arbres ou graphes). L'espace de caractéristiques H est relié à l'espace d'observation par une application :

$$\begin{aligned} \Phi \quad : \quad & \chi \to H \\ & x \mapsto \phi(x) \end{aligned}$$

Les données d'apprentissage sont un ensemble fini de points x_i, ou bien, dans le cas de l'apprentissage supervisé, un ensemble fini de couples (point, étiquette) $\{(x_i, y_i)\}$.

4.3.1 Méthodes linéaires

Nous rappelons brièvement les principes de trois méthodes classiques d'analyse de données, qui sont le fondement de plusieurs méthodes non linéaires plus récentes.

4.3.1.1 Analyse en Composantes Principales

L'analyse en composantes principales (ACP) - *Principal Component Analysis (PCA)* - est une ancienne approche, qui effectue une réduction de dimension par projection des points originaux dans un sous-espace vectoriel de dimension plus réduite. L'ACP détermine des axes de projections orthogonaux,

qui maximisent la variance expliquée. Dans la base formée par ces axes, les coordonnées ne sont pas corrélées. L'ACP maximise la variance de la projection dans l'espace de caractéristiques, ce qui est équivalent à minimiser l'erreur quadratique moyenne de reconstruction.

L'ACP se calcule en diagonalisant la matrice de corrélations, le plus souvent en utilisant une décomposition en valeurs singulières (SVD). Elle est très utilisée car elle est simple à mettre en oeuvre. Elle est limitée par son caractère linéaire : il est facile d'imaginer des situations dans lesquelles l'ACP n'apporte aucune information utilisable (par exemple, des données réparties sur un tore en dimension n). A titre illustratif, la figure 4.3 présente les Iris de Fisher dans la base obtenue par une ACP sous forme de nuages de points.

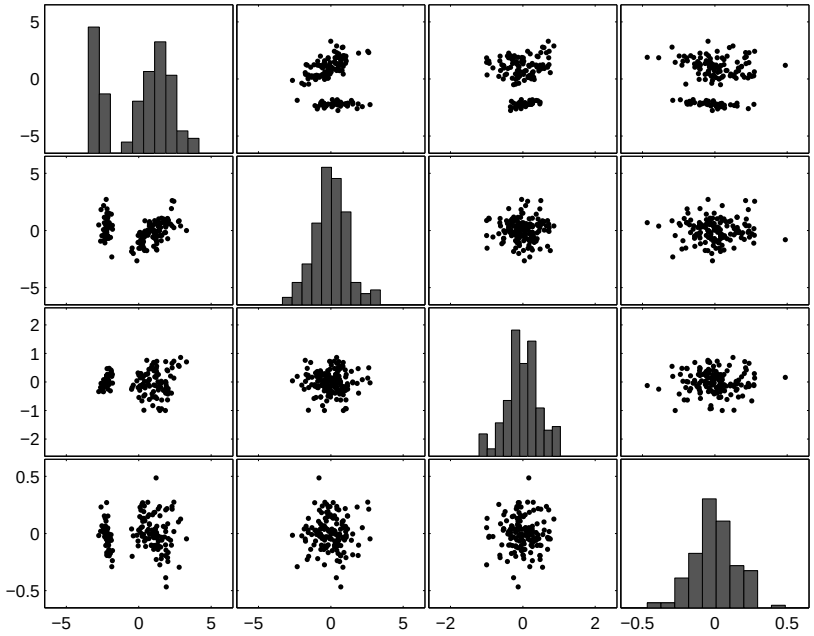

Figure 4.3 – Visualisation des Iris de Fisher sous forme de nuages de points dans la base fournie par l'ACP.

Plusieurs variantes de l'ACP ont été proposées pour faciliter l'interprétation de la projection obtenue ; ainsi, les méthodes *varimax*, *quartimax* et *equamax* s'appuient sur une rotation orthogonale des axes et les approches *oblimin* et *promax* utilisent des rotations obliques. La plus utilisée de ces variantes est sans nul doute la méthode *varimax* qui effectue une rotation orthogonale des axes pour obtenir des facteurs fortement corrélés à quelques variables et faiblement aux autres ; ainsi, chaque variable est identifiée à un - ou à un petit nombre de facteurs - et les axes sont facilement interprétables.

4.3.1.2 Analyse Discriminante

Proposée par Ronald A. Fisher en 1936 [Fis36], l'Analyse Factorielle Discriminante - *Fisher Discriminant Analysis (FDA)* - appelée aussi analyse discriminante linéaire de Fisher, s'applique lorsque les classes des individus sont connues. Elle consiste à chercher un espace vectoriel de faible dimension qui maximise la variance inter-classe. Une base de cet espace est obtenue en appliquant une Analyse en Composantes Principales sur les centroïdes des différentes classes pondérés par l'effectif de la classe correspondante avec Σ^{-1} comme métrique. On conservera, au plus, $(C-1)$ axes discriminants où C est

le nombre de classes.

4.3.1.3 Positionnement Multi-Dimensionnel

Dans de nombreux cas, on connaît les distances entre les points d'un ensemble d'apprentissage (on peut utiliser une mesure de similarité plus sophistiquée que la distance euclidienne, comme indiquée dans la section suivante), et on cherche à obtenir une représentation en faible dimension de ces points. La méthode de positionnement multidimensionnel[1] - *Multi-Dimensional Scaling (MDS)* - permet de construire cette représentation. L'exemple classique est d'obtenir la carte d'un pays en partant de la connaissance des distances entre chaque paire de villes. L'algorithme MDS est basé sur une recherche de valeurs propres

MDS permet de construire une configuration de m points dans $\mathbf{R}^d$ à partir des distances entre m objets. On observe donc $m(m-1)/2$ distances. Il est toujours possible de générer un positionnement de m points en m dimensions qui respecte exactement les distances fournies. MDS calcule une approximation en dimension $d < m$. L'algorithme est le suivant :

1. Moyennes des distances carrées par rangées : $\mu_i = \frac{1}{N}\sum_j d_{ij}$
2. Double centrage (distance carrée vers produit scalaire) : $P_{ij} = -\frac{1}{2}\left(d_{ij} - \mu_i - \mu_j + \sum_i \mu_i\right)$
3. Calcul des vecteurs propres v_j et valeurs propres λ_j principales de la matrice P (avec les λ_j les plus grands).
4. La i-ème coordonnée réduite de l'exemple j est $\sqrt{\lambda_j} v_{ij}$

Notons que la matrice de distance $D = (d_{ij})$ doit être semi définie positive.Les méthodes linéaires comme l'ACP et le MDS ne donnent des résultats intéressants que si les données sont situées sur un sous-espace linéaire. Elles ne peuvent traiter le cas où les données sont sur une variété très non linéaire.

4.3.2 Méthodes non linéaires

Les méthodes linéaires reposent (au moins implicitement) sur l'utilisation d'une distance euclidienne (liée au produit scalaire ordinaire). Dans de nombreuses applications, la distance euclidienne n'a pas grand sens ; elle suppose en particulier que toutes les variables sont comparables entre elles (elles doivent donc avoir été convenablement normalisées). La théorie des espaces de Hilbert permet de définir d'autres produits scalaires, basés sur des fonctions noyaux $k(x, y)$. k est alors une mesure de similarité entre les points de l'ensemble à traiter. Le noyau k défini implicitement une application de l'espace d'origine vers un "espace de caractéristiques" H. La dimension de l'espace H est éventuellement infinie. De nombreuses méthodes statistiques peuvent s'exprimer en ne recourant qu'à des produits scalaires entre les points à traiter et les exemples d'apprentissage. Si l'on remplace le produit scalaire habituel par un noyau k, on rend la méthode non-linéaire ; c'est le "truc du noyau" - *kernel trick* -, qui a fait l'objet de nombreuses recherches depuis son introduction par Vapnik [BGV92] dans le cadre des machines à vecteurs de support (SVM).

[1] On trouvera également dans la littérature les termes *Mise à l'échelle multidimentionnelle* et *Echelonnement multidimensionnel* ; en l'absence de consensus, nous avons retenu *Positionnement multidimensionnel* qui traduit le mieux l'objectif de la méthode.

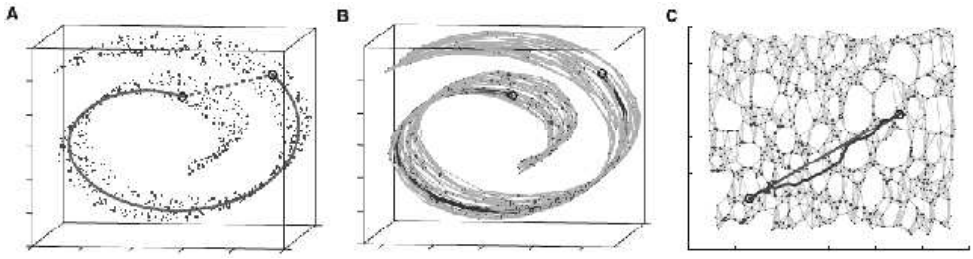

Figure 4.4 – Principe de l'algorithme Isomap. Les géodésiques sont construites en cherchant un chemin de proche en proche sur les points de l'échantillon (d'après [TdSL00]).

4.3.2.1 ACP "kernelisée"

La première approche permettant d'appliquer l'ACP au cas de données situées sur une variété non linéaire est d'effectuer des approximations locales : on calcule une ACP pour un groupe de points proches les uns des autres. Cette approche pose le problème de la définition des voisinages et du traitement des points nouveaux rencontrés loin des exemples connus.

Une autre approche, formalisée par B. Schölkopf en 1998, utilise le le "truc du noyau" - *kernel trick* - pour rendre non linéaire l'ACP traditionnelle. En effet, le calcul de l'ACP ne fait intervenir que des produits scalaires entre les points (pour le calcul de la matrice de covariance) et ne considère jamais les coordonnées d'un point isolé. Si l'on remplace le produit scalaire par un noyau, on calcule donc les composantes principales dans l'espace de caractéristiques H, et on peut ainsi accéder à des corrélations d'ordre supérieur entre les variables observées. Remarquons que l'on peut calculer la projection d'un point ne faisant pas partie de l'ensemble d'apprentissage, ce qui n'est pas le cas de toutes les méthodes de réduction de dimension non linéaires.

4.3.2.2 Isomap

Isomap [TdSL00] est une techniques de réduction de dimension qui comme la méthode de positionnement multidimensionnel (*MDS*) part de la connaissance d'une matrice de dissimilarités entre les paires d'individus. Le but est cette fois de trouver une variété (non linéaire) contenant les données. On exploite le fait que pour des points proches, la distance euclidienne est une bonne approximation de la distance géodésique sur la variété. On construit un graphe reliant chaque point à ses k plus proches voisins. Les longueurs des géodésiques sont alors estimées en cherchant la longueur du plus court chemin entre deux points dans le graphe. On peut alors appliquer *MDS* aux distances obtenues afin d'obtenir un positionnement des points dans un espace de dimension réduite.

4.3.2.3 Plongement localement linéaire

La méthode du plongement localement linéaire [RS00] - *Local Linear Embedded (LLE)* - a été présenté en même temps qu'Isomap et aborde le même problème par une voie différente. Chaque point est ici caractérisé par sa reconstruction à partir de ses plus proches voisins. *LLE* construit une projection vers un espace linéaire de faible dimension préservant le voisinage. Les différentes étapes de l'algorithme *LLE* sont rappelées à la figure 4.6.

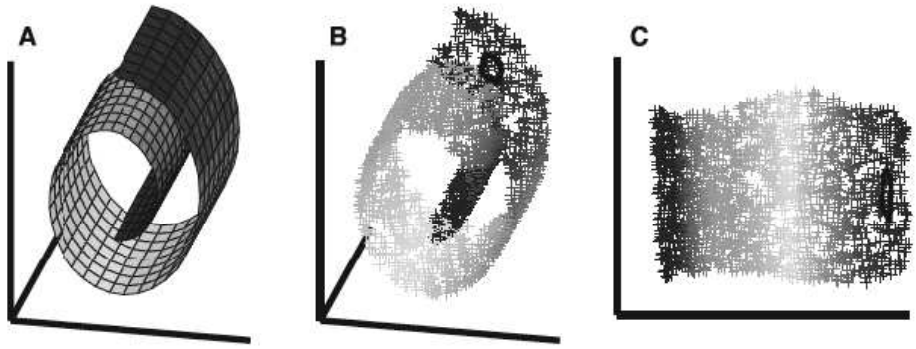

Figure 4.5 – Le problème de réduction de dimension : les points de l'échantillon, de dimension 3, (figure du milieu) sont situés sur la variété représentée à gauche. On cherche une représentation en deux dimensions (à droite) qui préserve la topologie (le voisinage de chaque point) (d'après [RS00]).

4.3.2.4 Approche neuromimétique : Réseaux de neurones auto-regressifs

Les réseaux de neurones auto-régressifs - *Auto-encoders* - sont parfois considérés comme une extension neuronale non linéaire de l'ACP. En effet, ils visent à minimiser l'erreur moyenne de reconstruction d'un individus à partir de sa projection sur un espace de dimension réduite. Comme l'illustre la figure 4.8, ce modèle neuronal comporte trois couches cachées :

- une couche d'encodage qui extrait une représentation non-linéaire des individus,
- une couche de compression qui compresse l'information,
- une couche de décodage qui permet de retrouver la représentation initiale d'un individus.

4.4 Conclusion

Au cours de ce chapitre, nous avons présenté les problématiques de la sélection de variables et de l'extraction de caractéristiques et nous avons rappelé les principes de quelques méthodes. Avant de poursuivre, rappelons que cette thèse s'inscrit dans le cadre de l'apprentissage non supervisé et que dans ce contexte, nous nous intéressons aux méthodes de réductions de dimensions pour la classification automatique. Les techniques d'extraction de caractéristiques non supervisées sont, soit limitées par leur caractère linéaire (ACP, MDS), soit difficilement utilisables à cause de leur complexité algorithmique lorsque l'on travaille sur de grandes bases données (LLE, Isomap). Bien que ce dernier point mérite d'être nuancé avec l'apparition de méthodes de calcul incrémental [BDL$^+$04], il nous semble malgré tout naturel de se focaliser sur les techniques de sélection de variables qui, à l'instar des méthodes d'extraction de caractéristiques, permettent de rester dans l'espace des observations et de ne pas imposer d'effort d'interprétation de nouvelles variables à l'utilisateur.

La sélection de variables en apprentissage non supervisé est un domaine encore peu exploré et les techniques existantes reposent pour beaucoup sur des mesures de similarité entre attributs ou sur des mesures de variances. Il s'agit d'un problème qui est plus difficile que dans le cas supervisé car aucune information n'est disponible pour guider la procédure. La détermination automatique du nombre de groupes est un problème associé très important et ces deux problèmes interfèrent l'un avec l'autres.

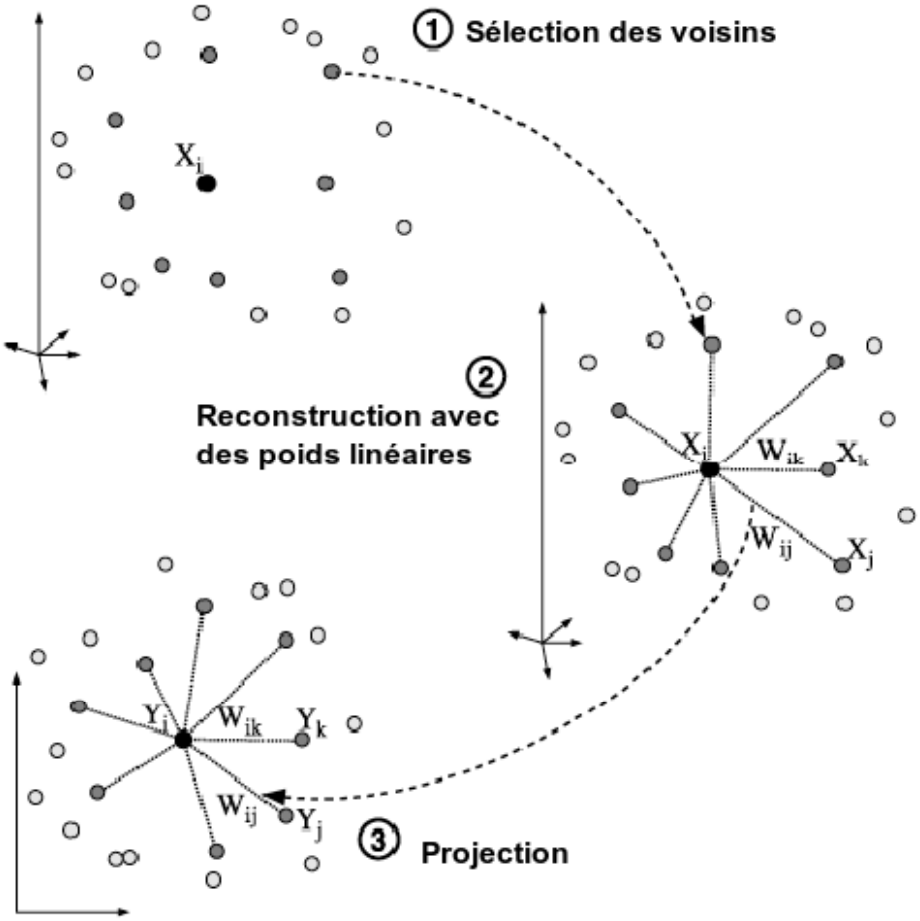

Figure 4.6 – Principe de fonctionnement de l'algorithme LLE (d'après [RS00]).

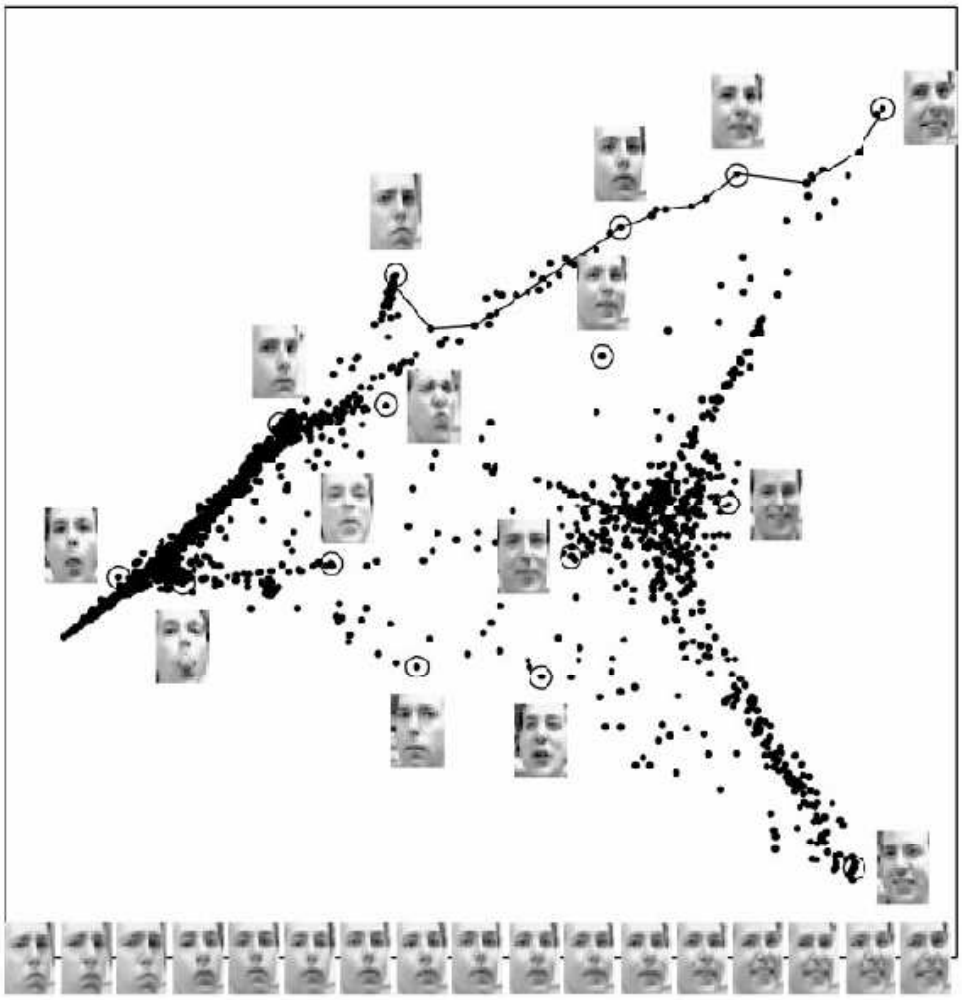

Figure 4.7 – Un exemple d'application de l'algorithme LLE : les points initiaux représentent des images de visages. Dans l'espace de dimension 2, ces images sont regroupées selon la position, l'éclairage et l'expression. Les images placées en bas de la figure correspondent aux points successifs rencontrés sur la ligne en haut à droite, balayant un continuum d'expression du visage. (d'après [RS00]).

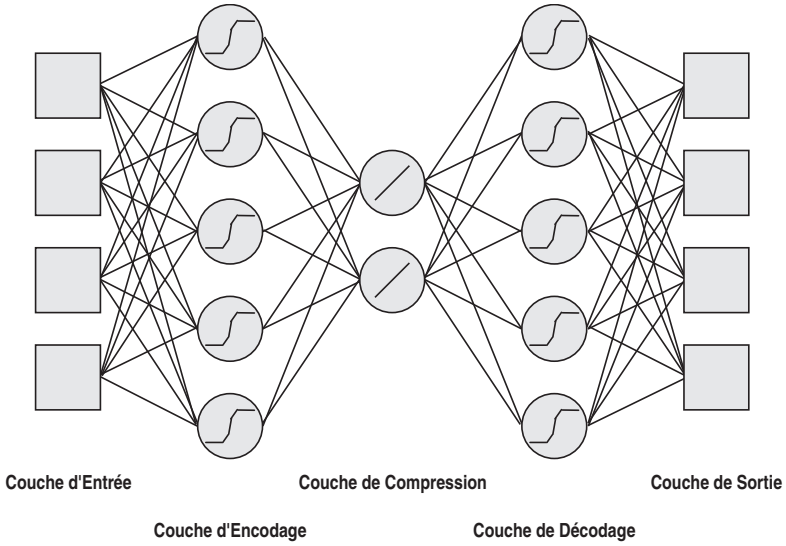

Figure 4.8 – Exemple de réseau auto-régressif : projection non-linéaire d'individus en 4 dimensions dans un espace de dimension 2.

PARTIE II

Approches proposées

CHAPITRE 5

Traitement des attributs redondants

5.1 Motivations

Les données utilisées par les applications réelles qui intègrent des techniques de fouille de données renferment souvent de nombreux attributs redondants. Si d'un coté cette redondance facilite la prise en compte de valeurs manquantes [CIL03] ou la détection de valeurs aberrantes, elle peut nuire par ailleurs à la découverte de structures intéressantes par les algorithmes de classification automatique basés sur l'utilisation de la distance euclidienne. Intuitivement, une information redondante, qui est représentée par de nombreux attributs, risque d'en occulter une autre qui bien qu'elle soit potentiellement pertinente est moins présente. Dans le pire cas, l'information pertinente est noyée parmi de nombreux attributs qui expriment tous une même idée sans intérêt pour l'utilisateur. Cette situation extrême risque de conduire à une classification sans réel intérêt pour l'utilisateur. Trois types d'approches sont généralement utilisées pour palier à ce problème : l'extraction de caractéristiques, la sélection et la pondération de variables.

Bien qu'elles soient souvent plus performantes que les méthodes de sélection de variables pour les problèmes de régression ou de prédiction, les méthodes d'extraction de caractéristiques imposent un effort important à l'utilisateur pour interpréter et comprendre la nouvelle représentation de ses données. La sélection de variables constitue donc une alternative très intéressante car l'utilisateur peut interpréter directement les résultats obtenus. Néanmoins, le fait d'éliminer complètement des variables complique la prise en compte des valeurs manquantes et nous n'avons pas retenu cette approche non plus. Nous nous sommes intéressés à la pondération des variables qui permet un ajustement plus fin de l'importance relative que l'on accorde aux différents attributs.

Nous proposons dans cette partie une nouvelle approche baptisée μ-SOM basée sur une classification simultanée des individus et des variables à l'aide de cartes auto-organisées qui sont connues pour permettre une bonne représentation de données en grande dimension. Un mécanisme de pondération s'appuyant sur la classification des variables est intégré à l'algorithme d'apprentissage pour diminuer l'influence des attributs redondants.

5.2 Approche proposée

5.2.1 Principes et algorithmes

J. Vesanto et J. Ahola [VA99] ont proposé une méthode de détection visuelle des corrélations entre variables basée sur l'algorithme des cartes auto-organisées proposé par Teuvo Kohonen [Koh01]. Leur approche débute par la construction d'une carte des observations dont sont ensuite extraits des profils de variables : chaque variable est représentée par le vecteur qui contient les valeurs qu'elle prend au niveau de chaque unité. Outre la robustesse aux valeurs aberrantes de cette représentation des variables,

Algorithm 1 Algorithme d'apprentissage μ-SOM

/* Initialisation */
$\mu_i \leftarrow \frac{1}{n},\ \forall i \in \{1, \ldots, n\}$,
$\omega_i \in \mathbb{R}^n,\ \forall i \in U^{(obs)} = \{1, \ldots, M\}$,
$\pi_i \in \mathbb{R}^M,\ \forall i \in U^{(var)} = \{1, \ldots, m\}$

/* Apprentissage grossier */
Apprentissage grossier de la carte des observations $SOM^{(obs)}$
Extraction des profils des variables fp_i à partir de $SOM^{(obs)}$
Apprentissage grossier de la carte des variables $SOM^{(var)}$
Calcul de la pondération correspondante μ_i^{new}
Mise à jour de la pondération $\mu_i \leftarrow \alpha_0.\mu_i + (1 - \alpha_0).\mu_i^{new}$

/* Apprentissage fin */
pour $t = 1, \ldots, T_{max}$ **faire**
  Epoque(s) d'apprentissage de la carte des observations $SOM^{(obs)}$
  Extraction des profils des variables à partir de $SOM^{(obs)}$
  Epoque(s) d'apprentissage de la carte des variables $SOM^{(var)}$
  Calcul de la pondération correspondante μ_i^{new}
  Mise à jour de la pondération $\mu_i \leftarrow \alpha_t.\mu_i + (1 - \alpha_t).\mu_i^{new}$
fin pour

les auteurs en montrent également la pertinence sur différents jeux de données artificiels et réels. Dans la perspective de détecter visuellement les corrélations entre variables, les auteurs proposent de présenter à l'utilisateur les différentes composantes en les réorganisant selon leurs corrélations ; ils construisent pour cela une carte auto-organisée dont chaque unité représente au plus une variable et affichent les différentes composantes en respectant l'ordre topologique ainsi "découvert".

Dans notre approche, la carte des observations et la carte des variables sont construites simultanément sans imposer de contrainte d'effectif pour la carte des variables. La carte des variables est ensuite utilisée pour calculer le poids de chaque dimension en tenant compte de leur redondance : un poids potentiel est attribué à chaque unité en fonction de l'homogénéité des prototypes dans son voisinage et ces poids potentiels sont ensuite partagés entre les variables qui se projettent dans le voisinage de l'unité correspondante.

L'algorithme 1 rappelle les grandes lignes de l'algorithme d'apprentissage μ-SOM. La carte des observations $SOM^{(obs)}$ est constituée de l'ensemble de M unités noté $U^{(obs)} = \{1, \ldots, M\}$. De manière analogue, la carte des variables $SOM^{(var)}$ se compose des m unités notées $U^{(var)} = \{1, \ldots, m\}$. Les prototypes respectifs des unités $i \in U^{(obs)}$ et $j \in U^{(var)}$ sont désignés par $\omega_i \in \mathbf{R}^n$ et $\pi_j \in \mathbf{R}^M$. Précisons maintenant quelques points de l'algorithme proposé :

- La recherche de l'unité gagnante sur la carte des observations s'effectue à l'aide de la distance euclidienne pondérée $d^{(obs)}(x, \omega_j) = \sqrt{\sum_{i=1}^{n} \mu_i (x_i - \omega_{ji})^2}$, avec $\sum_{i=1}^{n} \mu_i = 1$.
- Le paramètre α_t permet une prise en compte progressive de la pondération induite par la carte des variables au fur et à mesure qu'elle devient plus représentative.
- Les profils des différentes variables sont donnés par les lignes de la matrice dont les colonnes sont les prototypes des unités de la carte des observations.

5.2.2 Mécanisme de pondération proposé

Le mécanisme de pondération proposé repose sur le partage de l'importance des différentes variables $F = \{1, \ldots, n\}$ en fonction de leur similarité. On commence par attribuer une importance potentielle à chaque unité de la carte des variables que l'on partage ensuite entre les différents attributs ; les détails de ces deux étapes sont donnés ci-dessous.

5.2.2.1 Importance potentielle

L'objectif de notre approche est de diminuer l'importance relative des dimensions très redondantes, nous souhaitons donc attribuer une importance potentielle plus faible aux régions de la carte où les prototypes sont très similaires car elles correspondent aux zones de forte densité dans l'espace des variables. A cet effet, nous avons retenu l'indice d'auto-corrélation spatiale locale de Geary pour sa capacité à mesurer l'homogénéité relative des prototypes dans le voisinage d'une unité de la carte des variables. Chaque unité de la carte des variables se voit attribuer une importance potentielle qui correspond à la part de sa contribution à l'indice d'auto-corrélation spatiale de Geary [Zan05]. Cet indice est approximativement égal au rapport de la variance locale sur la variance globale et la contribution de l'unité i se définit ainsi :

$$\gamma_i = \frac{(m-1)}{\sum_{j=1}^{m} c_{ij}} \times \frac{\sum_{j=1}^{m} c_{ij} \|\pi_i - \pi_j\|^2}{\sum_{j=1}^{m} \|\pi_i - \pi_j\|^2} \tag{5.1}$$

où $c_{ij} \in \{0, 1\}$ indique si les unités i et j sont voisines ou non. Généralement, on fixe un seuil et on considère que deux unités sont voisines lorsque la distance qui les sépare est inférieure à ce seuil : $c_{ij} = \left(d^{(var)}(i,j) < \sigma\right)$, où $d^{(var)}(i,j)$ est la distance qui sépare les unités $i \in U^{(var)}$ et $j \in U^{(var)}$ sur la carte des variables et σ est la taille du voisinage pris en compte. On calcule ensuite la contribution de chaque unité i :

$$\widetilde{\gamma_i} = \frac{\gamma_i}{\sum_{j=1}^{m} \gamma_j} \tag{5.2}$$

5.2.2.2 Partage des importances potentielles

Pendant la phase d'apprentissage, la contribution de chaque profil de variable aux prototypes des différentes unités est contrôlée par la fonction de voisinage ν. On peut alors considérer que la partition des profils de variables utilisée pour la mise à jour des référents est la partition floue dont chaque partie correspond au support d'une unité de la carte des variables. Le degré d'appartenance $\delta_j(i)$ d'une variable i au support de l'unité j peut être calculé de la manière suivante :

$$\delta_j(i) = \frac{\nu_{b(i)j}}{\sum_{k=1}^{m} \nu_{b(i)k}} \tag{5.3}$$

où $b(i)$ correspond au référent de la variable i. L'importance potentielle de chaque unité $j \in U^{(var)}$ est ensuite répartie entre les variables $i \in F$ au prorata de leurs degrés d'appartenance aux supports des différentes unités :

$$\mu_i^{new} = \sum_{j=1}^{m} \widetilde{\gamma_j} \times \frac{\delta_j(i)}{\sum_{k=1}^{n} \delta_j(k)} \tag{5.4}$$

5.3 Evaluation

5.3.1 Données

Pour valider notre approche, nous avons utilisé différents jeux de données mis à la disposition de la communauté d'apprentissage artificiel par l'université de Californie à Irvine (UCI) [DNM98], ainsi qu'une base de données issue du domaine du marketing.

- **Isolet**[1] **:** Cette base issue du domaine de la reconnaissance de la parole comporte près de 7800 exemples qui sont décrits par 617 attributs et issus de 26 classes équiprobables.
- **Waveform :** Ce jeu de données artificielles comporte 5000 exemples répartis en trois classes obtenues par combinaison de deux des trois "vagues de base" et ajout d'un bruit gaussien de moyenne nulle et de variance 1 à chacune des 21 variables originales. Dans leur version bruitée, les vagues de Breiman comportent 19 dimensions supplémentaires qui suivent une loi normale de moyenne nulle et de variance 1.
- **Marketing :** Ce jeu de données comporte les réponses d'un millier de consommateurs interrogés sur leur appréciation d'une centaine de produits et sur leurs attentes. Cette base contient également des informations d'ordre socio-démographique comme l'âge, le sexe ou la catégorie socio-professionnelle des individus interrogés.

5.3.2 Amélioration de la qualité topologique de la carte des observations

Une procédure de validation croisée a été utilisée pour comparer la qualité des cartes obtenues par notre approche à celle auxquelles conduit l'algorithme de Kohonen. Les jeux de données *waveform* et *isolet* on été séparés en cinq parties dont quatre ont été utilisées pour l'apprentissage et la dernière pour l'évaluation de la qualité de la carte à l'aide de l'erreur moyenne de quantification, du taux d'erreurs topologiques et de la mesure de distortion.

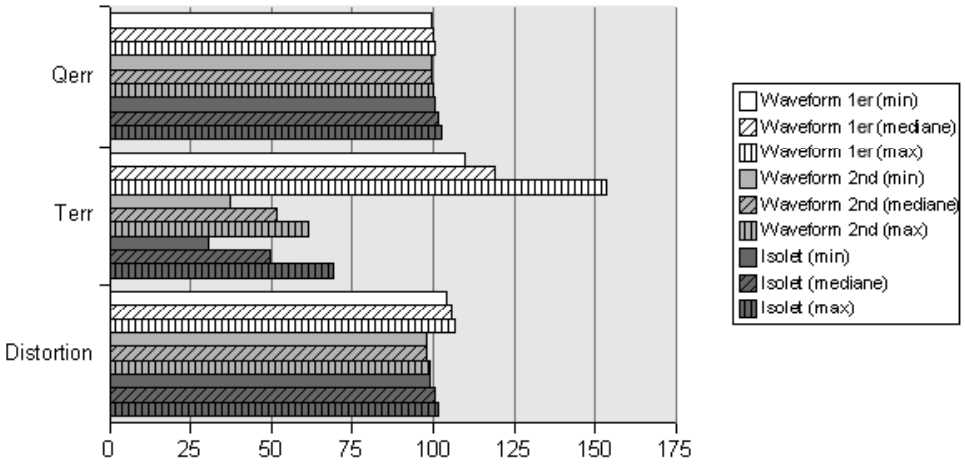

Figure 5.1 – Qualité relative des cartes construites par μ-SOM (indice 100 pour SOM). $Qerr$, $Terr$ et $Distortion$ correspondent respectivement à l'erreur de quantification (2.22), le taux d'erreurs topologiques (2.23) et la mesure de distortion (2.24)

[1] Isolated Letter Speech Recognition

Lors de notre première expérimentation avec le jeu de données *waveform*, la carte des variables comportait plus d'unités que de variables et était inutilisable pour identifier des corrélations intéressantes. Nous avons donc mené une deuxième série d'expérience en diminuant la taille de la carte. La figure 5.1 montre les valeurs relatives des critères de qualité que nous avons obtenues ; l'indice 100 correspond aux cartes construites par l'algorithme de Kohonen. On n'observe qu'il n'y a pas de différences significatives en ce qui concerne l'erreur de quantification moyenne et la mesure de distortion mais qu'en revanche le taux d'erreurs topologiques chute de manière significative.

5.3.3 Détection du bruit

Au cours de nos expérimentation avec la version bruitée des "vagues de Breiman", nous avons noté que les variables additionnelles qui correspondent à un bruit gaussien étaient regroupées au centre de la carte des variables. Nous n'avons pas étudié en détail ce phénomène, mais nous pouvons néanmoins en donner une explication probable. Une dimension bruitée ne participe pas à l'établissement de l'ordre topologique de la carte des observations et sa moyenne est quasiment nulle dans chacune des régions de Voronoï des unités de la carte des observations. Ainsi, les profils suivent approximativement une loi normale $N(\overrightarrow{0}, \epsilon\, I)$ où I est la matrice identité et $0 < \epsilon \ll 1$. Ensuite, le processus d'auto-organisation de la carte des variables conduit à un gradient spatial des valeurs et les prototypes du centre de la carte sont approximativement égaux au vecteur nul. Enfin, le référent qui a la plus grande probabilité d'être le plus proche d'un profil de variable qui ne correspond à aucune structuration de la carte des observations est le vecteur nul.

5.3.4 Application aux données marketing

Nous avons appliqué l'algorithme μ-SOM à une base de données issue du domaine du marketing afin d'identifier d'une par des segments de consommateurs et d'autre part des catégories de produits et d'attentes. Les cartes obtenues ont été découpées à l'aide de la méthode des k-moyennes et le nombre de classes a été sélectionné à l'aide de l'indice de Davies Bouldin.

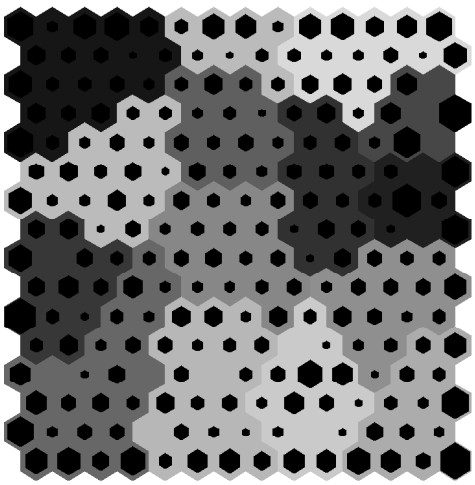

Figure 5.2 – Répartition des consommateurs sur la carte des observations segmentée.

Les figures 5.5 et 5.2 montrent respectivement les catégories de produits et les segments de consommateurs mis en évidence. Ensuite, la figure 5.6 indique la répartition des poids sur la carte des variables.

Enfin, les figures 5.3 et 5.4 montrent des anomalies de regroupement de variables.

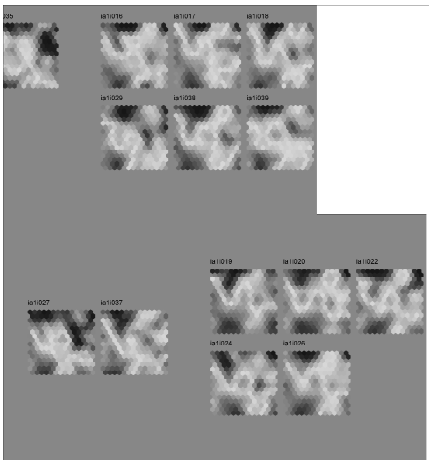

Figure 5.3 – Carte des variables : zoom sur la région située en haut à droite de la figure 5.5.

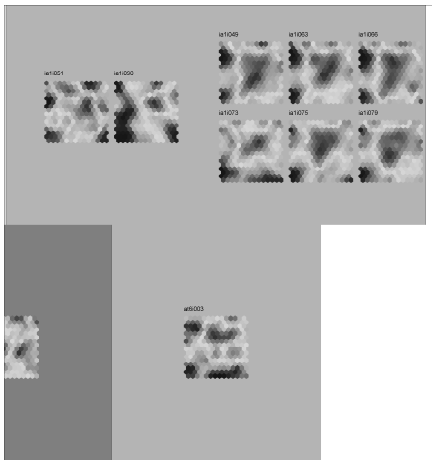

Figure 5.4 – Carte des variables : zoom sur la région située à mi hauteur et à droite de la figure 5.5.

5.4 Discussion

5.4.1 Distances entre profils de variables

Pour construire la cartes des variables, nous avons transformé les profils de variables pour que les valeurs de chaque dimension soient dans l'intervalle $[0; 1]$ et nous avons ensuite utilisé une distance euclidienne ; cela nous a conduits à observer un certain nombre d'anomalies et nous pensons que ce point fort criticable de l'algorithme mérite d'être amélioré. En outre, la distance euclidienne ne permet pas de rapprocher deux variables très corrélées négativement comme le ferait par exemple le coefficient de corrélation de Pearson. Ensuite, les profils des variables sont extraits de la carte des observations et une mesure de dissimilarité appropriée devrait également prendre en compte l'organisation spatiale de cette dernière.

Les remarques précédentes nous amènent à revoir notre définition de la similarité de deux profils de variable ; ainsi, en considérant que deux variables sont d'autant plus proches qu'elles induisent des découpages similaires de la carte des observations, on pallie ainsi aux deux lacunes majeures de la distance euclidienne énoncées au paragraphe ci-dessus. Nous proposons de ramener ce problème de comparaison de profils de variable au problème de comparaison des partitions qu'ils induisent. On procède alors au découpage de la carte des observations selon les différentes dimensions prises une à une et on mesure la dissimilarité entre les partitions obtenues. Il convient de souligner ici qu'en procédant ainsi, il est également possible de mesurer la dissimilarité entre deux sous-ensembles non vides quelconques de variables. On utilisera par exemple une classification ascendante hiérarchique et la variation d'information.

5.4.2 Importance potentielle

Il est important de souligner ici un biais induit par la structure de la carte : les unités du bord de la carte sont défavorisées par rapport aux autres. En effet, elles ont moins de voisines et la variance locale

des prototypes sur les bords est plus faibles : l'importance potentielle qui en résulte est donc plus faible. Il conviendrait donc d'ajouter un terme de pénalisation à l'expression (5.1) pour remédier à ce problème.

Ensuite, nous avons mis en évidence au paragraphe précédent que la distance euclidienne n'est vraisemblablement pas la mesure de dissimilarité optimale pour notre problème. Ceci nous conduit à critiquer également l'utilisation d'un indice statistique qui l'utilise dans sa définition. Ainsi, la mesure de l'homogénéité des prototypes dans une région de la carte des variables mérite également notre attention ; on pourra par exemple définir une mesure basée sur la variation d'information entre les partitions induites par les différents prototypes.

Enfin, le mode de calcul de l'importance potentielle que nous avons utilisé ne s'intéresse qu'aux aspects liés à la redondance et ne prend pas en compte explicitement la pertinence des variables. La notion de pertinence d'une variable n'est pas clairement définie dans le cadre de l'apprentissage non supervisé mais nous pouvons proposé ici de considérer qu'une variable est d'autant plus pertinente qu'elle met en exergue une structure spatialement marquée sur la carte des observations qui soit en cohérence avec la structure globale émergente. Un indice d'auto-corrélation spatiale locale d'une variable prototype peut mettre en évidence l'existence d'une structure spatiale marquée mais ne permet pas de vérifier sa cohérence avec la structure globale émergente ; nous proposons une fois de plus d'utiliser la variation d'information à cet effet pour comparer les partitions induites par les différentes composantes à la structure globale émergente.

5.4.3 Algorithme d'optimisation

L'optimisation des prototypes des deux cartes est réalisée à l'aide de la version *batch* de l'algorithme de Kohonen qui, à l'instar de sa version stochastique, présente l'avantage d'être déterministe. Mais bien que motivé, ce choix ne conduit pas à des résultats optimaux pour des raisons inhérentes à la version l'algorithme *batch* d'une part [FLC02] et à une optimisation séparée de deux fonctions de coût différentes d'autre part. La fonction de coût de notre algorithme μ-SOM peut s'écrire comme la somme des fonctions de coût des deux cartes auto-organisées

$$R_{\mu-SOM} = \sum_{i=1}^{N}\sum_{j=1}^{M} h_{b(i)j} \sum_{k=1}^{n} \mu_k \left(x_{ik} - \omega_{jk}\right)^2 + \sum_{k=1}^{n}\sum_{l=1}^{m} \nu_{b(k)l} \sum_{j=1}^{M} \left(\omega_{jk} - \pi_{lj}\right)^2 \tag{5.5}$$

où h et ν sont respectivement les fonctions de voisinage de la carte des observations et de la carte des variables, et où les poids μ_k des différents attributs sont déterminés à l'aide de l'équations (5.4). On pourra alors utiliser le formalisme lagrangien d'optimisation de systèmes modulaires proposé dans [BG91, Bot91] pour optimiser les paramètres des deux cartes simultanément.

5.5 Conclusion

Une approche originale baptisée μ-SOM et basée sur une classification simultanée des individus et des variables à l'aide de cartes auto-organisées a été présentée au cours de ce chapitre. Elle intègre un mécanisme de pondération s'appuyant sur la classification des variables pour diminuer l'influence des attributs redondants pendant l'apprentissage. Bien que l'application de cette méthode à des données réelles issues du domaine du marketing nous aie donné satisfaction, elle a aussi permis de mettre en évidence un certain nombre d'anomalies. Ce dernier point, discuté à la fin du chapitre, a été l'occasion d'envisager différentes améliorations possibles de notre méthode.

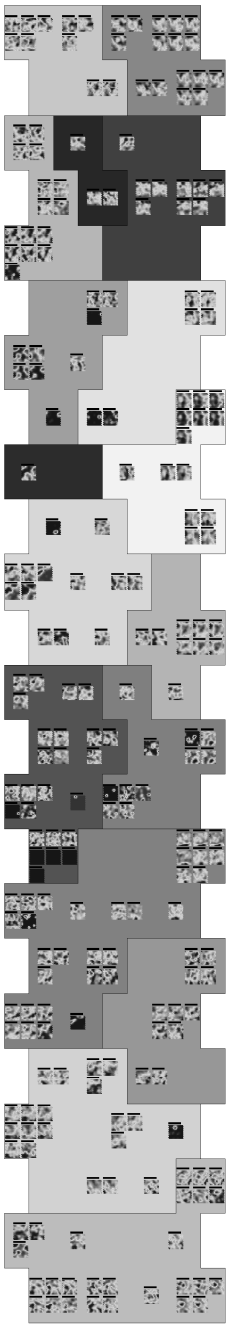

Figure 5.5 – Répartition des attributs et des catégories sur la carte.

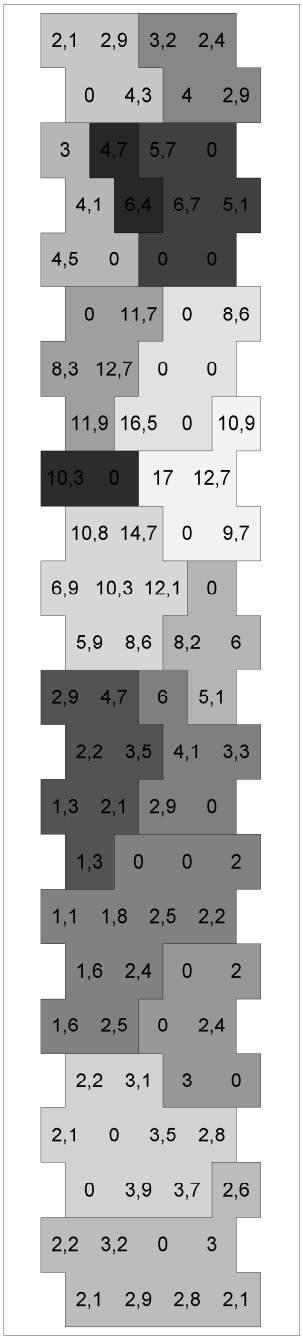

Figure 5.6 – Répartition des poids ($\times 10^{-3}$) des attributs.

CHAPITRE 6

Sélection de variables et du nombre de groupes

6.1 Motivations

La fouille de données est avant tout une démarche exploratoire et l'utilisateur n'a généralement d'idée précise ni sur le nombre de groupes présents dans ses données, ni sur les attributs qui les décrivent au mieux. S'il existe d'une part des approches filtres de sélection de variables non supervisée [ML01, MMP02] et d'autre part de nombreux critères pour choisir une meilleure classification parmi plusieurs classifications possibles [HBV01], la sélection simultanée du nombre de groupes et d'un sous-ensemble d'attributs pertinents demeure un des nombreux défis de la classification automatique.

Nous proposons dans ce chapitre une approche originale de sélection simultanée du nombre de groupes et d'un sous-ensemble de variables pertinentes au regard des groupes identifiés. Celle-ci repose sur une classification à deux niveaux et utilise deux mesures de pertinence basées sur l'indice de Davies-Bouldin : la première quantifie la pertinence individuelle de chaque variable et la seconde permet de tenir compte de la pertinence mutuelle des variables.

6.2 Approche proposée

6.2.1 Principes et algorithmes

Nous avons rappelé dans la section 4.2 qu'une procédure de sélection de variables se compose de trois éléments essentiels : une mesure d'évaluation, une stratégie de recherche et un critère d'arrêt. La plupart des approches proposées pour la sélection de variables non supervisée sont des approches filtres qui se basent sur la similarité ou la redondance des attributs. Nous proposons dans ce chapitre une approche intégrée de sélection de variables pendant un processus de classification automatique à deux niveaux : construction d'une carte auto-organisée et segmentation de cette carte. Les méthodes de classification à deux niveaux présentent deux intérêts majeurs : d'une part elles améliorent la robustesse à la présence de valeurs aberrantes et d'autres part elles permettent d'évaluer la qualité de nombreuses partitions comportant différents nombres de groupes sans que les temps de calcul ne deviennent prohibitifs [VA00].

Nous commençons par construire une carte auto-organisée que l'on découpe par la méthode des K-moyennes combinée à l'indice de Davies-Bouldin comme cela est suggéré dans [VA00]. Nous utilisons ensuite un indicateur statistique appelé "valeur test" qui a été proposée par [Mor84] pour caractériser les différentes parties d'une partition. Les variables sont ensuite éliminées tour à tour selon l'ordre établi par cette mesure à condition que leur suppression ne dégrade pas la qualité de la partition au sens de l'indice de Davies-Bouldin. Si les suppressions conduisent à une perte de qualité de la partition, on choisit la dégradation minimale. Ce processus d'élimination arrière est répété tant qu'on observe pas

perte d'information significative au sens de la statistique de Wilks. L'algorithme 2 rappelle les grandes lignes de notre approche.

Algorithm 2 Procédure de sélection de variables

/* Initialisation */
$R \leftarrow F$

/* Procédure de recherche : élimination arrière */
tant que ($\neg$critère d'arrêt) **faire**
  Construction d'un modèle
  Evaluation de la pertinence individuelle $R_{individuelle}(j)$
  Tri des variables selon la pertinence individuelle croissante
  $trouve \leftarrow false$
  tant que $(\neg trouve)$ **faire**
    Evaluation de la pertinence collective $R_{collective}(j)$ de la variable la moins pertinente individuellement
    si $(R_{collective}(j) \leq \theta)$ **alors**
      $trouve \leftarrow true$
      $R \leftarrow R \setminus \{j\}$
    fin si
  fin tant que
  si $(\neg trouve)$ **alors**
    $j \leftarrow \arg\min_{k \in R} \{R_{collective}(k)\}$
    $R \leftarrow R \setminus \{j\}$
  fin si
fin tant que

6.2.2 Mesures d'évaluations proposées

6.2.2.1 Pertinence individuelle

Comme nous l'avons rappelé au chapitre 5, la notion de pertinence d'une variable n'est pas clairement définie dans le cadre de l'apprentissage non supervisé et nous avons proposé de considérer qu'une variable est d'autant plus pertinente qu'elle met en évidence une structure marquée de l'espace des observations qui soit en cohérence avec la structure globale émergente. Dans cet esprit, nous proposons ici d'utiliser un indicateur statistique, appelé valeur test [Mor84], qui est permet habituellement d'identifier les meilleurs descripteurs d'un groupe relativement à la population dont il est issu. La valeur test d'une variable pour un groupe est définie comme la différence entre la moyenne du groupe et la moyenne de la population exprimée en nombre d'écart-type du groupe. La valeur absolue cette mesure quantifie la pertinence du choix d'une variable comme descripteur d'une sous-population.

Nous définissons alors la pertinence individuelle d'une variable comme le maximum en valeur absolue de ses valeurs tests sur l'ensemble des groupes identifiés ; dans ces conditions, dès lors qu'elle permet de mettre en avant un groupe d'objets, une variable est considérée comme individuellement pertinente.

Etant donné un découpage en C groupes, la pertinence individuelle de la variable j s'exprime donc ainsi :

$$R_{individuelle}(j) = \max_{k=1,\ldots,C} \left\{ \left| \frac{\mu_{kj} - \mu_j}{\sigma_{kj}} \right| \right\} \tag{6.1}$$

où μ_{kj} et σ_{kj} sont respectivement la moyenne et l'écart-type de la variable j dans le groupe k, et où μ_j est la moyenne de la population totale.

6.2.2.2 Pertinence collective

La mesure de pertinence individuelle définie ci-dessus n'est pas suffisante car elle ne tient pas compte de la pertinence mutuelle au sein d'un ensemble de variable. Ainsi, elle pourrait nous conduire à éliminer une variable qui n'est pas pertinence en elle-même mais qui associée aux autres est très intéressante. Nous proposons d'utiliser une mesure de pertinence collective pour mesurer l'intérêt d'une variable lorsqu'elle est associée à une sous-ensemble de variable $R \subset F$.

Rappelons que nous disposons d'une partition optimale relativemement à notre critère d'évaluation : l'indice de Davies-Bouldin. Nous définissons la pertinence collective d'une variable comme la perte en qualité qu'engendre sa suppression ; notre mesure est alors définie comme la différence entre la valeur de l'indice de Davies-Bouldin avec et sans la variable j :

$$R_{collective}(j) = I_{DB}|_R - I_{DB}|_{R\setminus\{j\}} \tag{6.2}$$

où $I_{DB}|R$ et $I_{DB}|_{R\setminus\{j\}}$ sont respectivement les indices de Davies-Bouldin évalués en prenant en compte la variable j et sans la prendre en considération.

6.2.3 Stratégie de recherche

Bien que la stratégie de sélection avant soit généralement plus efficace d'un point de vue computationnel, nous avons adopté la stratégie d'élimination arrière car elle permet de prendre en compte la pertinence mutuelle des variables. Notre procédure de recherche est guidée par la mesure de pertinence individuelle qui permet d'établir un ordre d'intérêt relativemement à une partition. Et la mesure de pertinence collective sert à lever un véto si la suppression d'une variable engendre un dégradation de la qualité de la partition ; on considère dans ce cas qu'associée aux autres variables, elle est importante. Si toutes les variables du sous-ensemble sont pertinentes, la moins intéressante est éliminée.

6.2.4 Critère d'arrêt

T. Cibas utilise la statistique de Wilks pour évaluer si un sous-ensemble d'attributs apporte une information supplémentaire par rapport à un autre [Cib96]. Nous avons retenu cette approche pour arrêter le procédure d'élimination arrière lorsque la suppression d'un attribut entraîne une perte d'information significative.

Sous l'hypothèse que l'ensemble des attributs F suivent une loi normale

$$N(\mu^{(k)}, \Sigma) \ : \ k = 1, \ldots, C \tag{6.3}$$

où $\mu^{(k)}$ est la moyenne des attributs de F dans le groupe k et où Σ est la matrice de covariance. On peut décomposer cette matrice et ces vecteurs de la manière suivante :

$$\mu^{(k)} = \left(\mu_1^{(k)}, \mu_2^{(k)}\right), \tag{6.4}$$

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \tag{6.5}$$

où les indices 1 et 2 correspondent respectivement aux sous-ensemble d'attributs R et $F \setminus R$. L'hypothèse nulle qui exprime que l'ensemble $F \setminus R$ ne donne pas d'information supplémentaire par rapport à l'ensemble R s'écrit ainsi :

$$H_0 : \mu_2^{(k)} - \mu_2^{(h)} - \Sigma_{21}\Sigma_{11}^{-1}\left(\mu_1^{(k)} - \mu_1^{(h)}\right) = 0 \tag{6.6}$$

avec $k \neq h = 1, \ldots, C$. Le test de cette hypothèse repose sur la statistique de Wilks. Définissons les matrices de covariance inter-classe B - pour *between* - et intra-classe W - pour *within* - de la manière suivante :

$$\begin{aligned} B &= \sum_{k=1}^{C} N^{(k)} \left(\mu^{(k)} - \overline{\mu}\right)\left(\mu^{(k)} - \overline{\mu}\right)^T \\ W &= \sum_{k=1}^{C}\sum_{i=1}^{N^{(k)}} \left(x_i^{(k)} - \mu^{(k)}\right)\left(x_i^{(k)} - \mu^{(k)}\right)^T \end{aligned}$$

où $N^{(k)}$ est le nombre d'objets présents dans le groupe k et $\overline{\mu}$ est la moyenne globale des attributs de F. Les matrices B, W et leur somme $T = B + W$ peuvent se décomposer en bloc de la même manière que Σ :

$$\begin{aligned} B &= \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \\ W &= \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \\ T &= B + W = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} \end{aligned}$$

Ainsi, le déterminant des matrices W et T s'écrivent

$$\begin{aligned} |W| &= |W_{11}|\left|W_{22} - W_{21}W_{11}^{-1}W_{12}\right| \\ |T| &= |T_{11}|\left|T_{22} - T_{21}T_{11}^{-1}T_{12}\right| \end{aligned}$$

Ensuite, on note :

$$K = \frac{\left|W_{22} - W_{21}W_{11}^{-1}W_{12}\right|}{\left|T_{22} - T_{21}T_{11}^{-1}T_{12}\right|} \tag{6.7}$$

qui a $\frac{(N-C-r)}{(C-1)}$ degrés de liberté. En utilisant les notations définies ci-dessus, la statistique de Wilks pour n variables s'exprime :

$$\begin{aligned} \Lambda_F &= \frac{|W|}{|T|} \\ &= K.\frac{|W_{11}|}{|T_{11}|} \\ &= K.\Lambda_R \end{aligned}$$

ce qui indique que, pour de petite valeur de K, les groupes sont mieux séparés avec n variables qu'avec r. Ainsi, l'hypothèse nulle (6.6) est vraie si et seulement si les attributs de R permettent la même séparabilité des groupes que l'ensemble complet des attributs F. Pour finir, la statistique de Wilks Λ est

équivalente à celle de Fisher-Snedecor et :

$$F_s = \frac{(N - C - r)}{(C - 1)} \frac{1 - K}{K} \tag{6.8}$$

suit la loi de Fisher suivante $F\,(C - 1, N - C - r)$ où C est le nombre de groupes, N le nombre d'individus et r le nombre d'attributs conservés.

6.3 Evaluation

6.3.1 Données

L'université de Californie à Irvine (UCI) met à la disposition de la communauté d'apprentissage artificiel de nombreux jeux de données pour valider leurs approches [DNM98]. Nous en avons retenu quatre de taille et de complexité variables pour valider notre algorithme :

- **Wisconsin Diagnostic Breast Cancer (WDBC) :** Les données de cette base de données ont été recueillies à partir d'images numérisées d'un prélèvement par biopsie d'une masse éventuellement cancéreuse. Elles décrivent les caractéristiques de noyaux de cellule présents dans l'image. Les exemples sont répartis en deux classes selon qu'il s'agit de tumeurs malignes (212 exemples) ou bénines (357 exemples). On notera qu'il s'agit d'un problème relativemement simple : les classes sont linéairement séparable et l'état de l'art fait mention d'une précision supérieure à 97 % en classement.
- **Glass :** Cette base contient les caractéristiques de 214 échantillons de verres suivantes : indice de réfraction, oxyde de sodium, oxyde magnésium, oxyde d'aluminium, oxyde de silicium, oxyde de potassium, oxyde de calcium, oxyde de baryum et oxyde de fer. Les différentes instances se répartissent dans les 7 classes suivantes : 70 dans la classe 1 (verre traité utilisé en construction), 76 dans la classe 2 (verre traité utilisé dans les véhicules), 17 dans la classe 3 (verre non traité utilisé en construction), 0 dans la classe 4 (verre non traité utilisé dans les véhicules), 13 dans la classe 5 (bocaux), 9 dans la classe 6 (vaisselle) et 29 dans la classe 7 (tête d'ampoule). La classe 4 n'étant pas représentée, on peut considérer qu'il s'agit d'un problème à 6 classes.
- **Waveform :** Ce jeu de données artificielles comporte 5000 exemples répartis en trois classes obtenues par combinaison de deux des trois "vagues de base" et ajout d'un bruit gaussien de moyenne nulle et de variance 1 à chacune des 21 variables originales. Dans leur version bruitée, les vagues de Breiman comportent 19 dimensions supplémentaires qui suivent une loi normale de moyenne nulle et de variance 1.
- **Wine :** Cette base recense les résultats d'une analyse chimique de différents vins produits à dans une même région d'Italie à partir de différents cépages. La concentration de 13 constituants est indiquée pour chacun des 178 vins analysés qui se répartissent ainsi : 59 dans la classe 1, 71 dans la classe 2 et 48 dans la classe 3.

6.3.2 Résultats

Nous avons utilisé la version *batch* de l'algorithme de Kohonen et l'algorithme *global k-means* qui sont tous les deux déterministes pour nos expérimentations. Les résultats présentés dans le tableau 6.1 sont les moyennes et écart-type obtenus après cinq validations croisées ; l'ensemble des données a été séparé en dix parties dont neuf ont servi à l'apprentissage et la dernière a été utilisée pour le test. Les

				Apprentissage		Test	
		C_T [σ_{C_T}]	n_{FS} [$\sigma_{n_{FS}}$]	I_{Rand} [$\sigma_{I_{Rand}}$]	P_R [σ_{P_R}]	I_{Rand} [$\sigma_{I_{Rand}}$]	P_R [σ_{P_R}]
Glass	F	7.04 [0.73]	9.0 [–]	0.301 [0.012]	56.25 [2.56]	0.295 [0.068]	67.52 [9.01]
189 - 21	R	5.10 [1.83]	2.84 [1.46]	0.376 [0.082]	50.83 [6.54]	0.382 [0.121]	58.38 [10.40]
Wine	F	6.86 [0.81]	13.0 [–]	0.171 [0.022]	93.59 [1.97]	0.165 [0.064]	95.28 [5.11]
189 - 21	R	5.70 [2.34]	6.3 [2.1]	0.247 [0.060]	80.32 [12.02]	0.239 [0.096]	83.44 [13.78]
WDBC	F	9.72 [0.67]	30.0 [–]	0.414 [0.014]	93.83 [1.56]	0.417 [0.026]	94.16 [3.03]
242 - 27	R	2.72 [1.96]	12.4 [3.3]	0.182 [0.077]	91.53 [1.04]	0.184 [0.091]	91.60 [3.49]
Wave	F	6.18 [2.56]	40.0 [–]	0.304 [0.016]	68.64 [8.48]	0.309 [0.014]	66.17 [7.82]
500 - 4500	R	4.82 [1.55]	28.2 [9.56]	0.304 [0.020]	66.93 [6.62]	0.306 [0.018]	65.97 [6.68]

Table 6.1 – Les deux nombres situés sous le nom des jeux de données indiquent respectivement la taille des ensembles d'apprentissage et de test. L'ensemble de tous les attributs est noté F et l'ensemble des attributs sélectionnés par R.

critères utilisés pour l'évaluation sont des critères externes et font intervenir les étiquettes qui sont disponibles pour les jeux de données utilisés. L'indice de Rand a été présenté au paragraphe 3.2.2 et l'indice de pureté correspond à la moyenne de la part de la classe majoritaire au sein des groupes découverts.

6.4 Discussion

6.4.1 Segmentation de la carte

Pour segmenter la carte auto-organisée, nous avons utilisé l'algorithme des k-moyennes associé à l'indice de Davies Bouldin comme cela est proposé dans [VA00]. Plus précisement, pour éviter les problème d'instabilité dont souffre la méthode des k-moyennes, nous utilisé l'algorithme *global kmeans* qui en est une version déterministe. Malheureusement, un article récent montre que cette approche mène généralement à des résultats sous-optimaux [HNCM05] et il convient d'envisager d'autres méthodes de découpage de la carte. Les prototypes sont généralement beaucoup moins nombreux que les observations et une classification ascendante hiérarchique peut donc être utilisée pour la segmentation de la carte. On s'affranchit alors des problèmes d'instabilité en conservant une complexité raisonnable puisque la même hiérarchie est utilisée pour évaluer différents découpages. Néanmoins, qu'elles soient basées sur une étude de la continuité [Mur95] ou sur la matrice des distances unifiées [MU05, OM04, US90, Ult05], d'autres méthodes de segmentation spécifiquement développées pour les cartes auto-organisées mériterait d'être utilisées pour compléter l'évaluation de notre méthode.

6.4.2 Stratégie de recherche

La méthode de sélection de variables proposée utilise une procédure de type élimination arrière - *backward* - qui nous permet de prendre en considération les interactions entre variables. Néanmoins, ce type de procédure est coûteux et un attribut très intéressant pourraient être éliminé dès le début de la procédure de sélection sans que cette décision ne soit remise en cause par la suite. En outre, les mesures de pertinence proposées s'appuient sur un découpage de la carte auto-organisée que l'on cherche à renforcer sans que sa pertinence ne soit garantie ; en particulier, au début de la procédure il est possible les structures intéressante présentes dans les données ne soient difficiles à détecter.

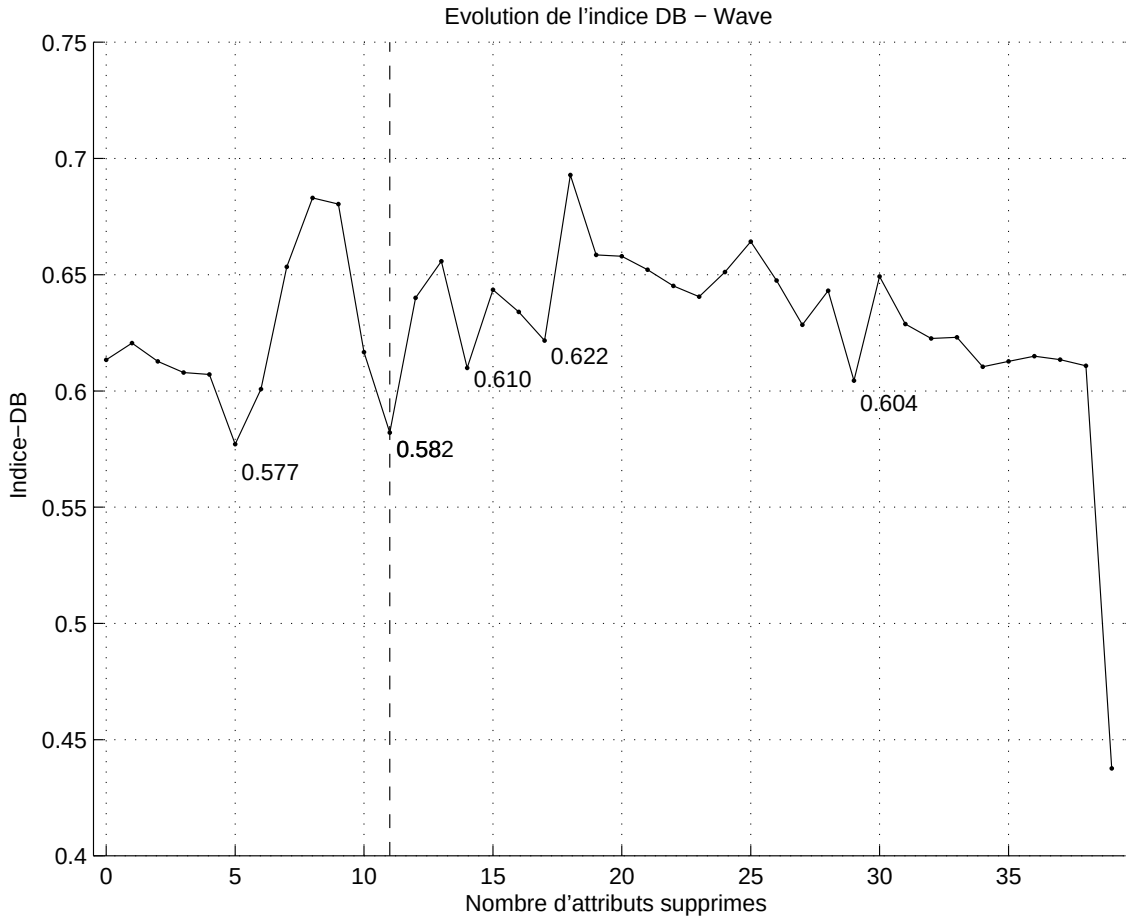

Figure 6.1 – Evolution de l'indice de Davies-Bouldin pendant la procédure d'élimination arrière : la ligne vertical en pointillée indique le modèle retenu par notre critère d'arrêt.

Ce point de l'algorithme peut être améliorer de différentes manière ; on pourrait, par exemple, utiliser des méthodes de recherche bi-directionnelles ou des méthodes de parcours aléatoire comme les algorithmes génétiques, mais nous pensons que la pondération de variables est également une alternative intéressante car elle permet l'apprentissage progressif d'une mesure de pertinence et une erreur d'appréciation au début de la procédure n'est alors plus irréversible.

6.4.3 Critère d'arrêt

Le critère d'arrêt que nous avons retenu est un facteur limitant majeur de l'approche présentée dans ce chapitre, car outre sa complexité importante liée au calcul de déterminant de matrice, il impose que la condition suivante doit être vérifiée :

$$N - C \geq n \tag{6.9}$$

pour garantir que la matrice de covariance intra-classe ne soit pas singulière. En d'autres termes, le nombre de variables ne peut dépassé le nombre d'individus moins le nombre de classes ; l'utilisation de notre approche sur des données spectrométriques ou sur des données génomiques qui comptent souvent d'avantage de variables qu'elles ne comportent d'individus n'est donc pas possible sans une modification préalable de ce point essentiel.

6.5 Conclusion

Une méthode de sélection de variables intégrée à un algorithme de classification a été présentée dans ce chapitre. Elle s'appuie sur la robustesse et l'efficacité des méthodes de classification à deux niveaux et combine une mesure de pertinence individuelle à un critère d'évaluation des attributs au sein d'un groupe. Notre approche permet d'une part de sélectionner le nombre de groupes en utilisant un critère de qualité de partition et d'autre part de renforcer progressivement la structuration des données en sélectionnant les attributs qui y contribue. Les limites de cette approche ont été mentionnées à la fin du chapitre.

CHAPITRE 7

Pondération et Sélection de variables

7.1 Motivations

Comme nous l'avons souligné à la fin du chapitre précédent, l'élimination d'une variable pertinente au début d'une procédure séquentielle de sélection de variables, sans que cette décision ne puisse être remise en cause ensuite, risque de conduire l'utilisateur à des résultats sans réel intérêt. Nous proposons dans ce chapitre une approche de sélection de variables qui s'appuie sur l'optimisation d'une pondération qui permet d'évaluer progressivement la pertinence des différentes dimensions.

Notre approche consiste à étendre la méthode de pondération des variables proposée par Huang [HNRL05] pour les algorithmes de type k-moyennes au cas des cartes auto-organisées. A cet effet, nous proposons d'introduire une contrainte de préservation de la topologie locale de l'espace d'entrée à l'aide d'une fonction de voisinage. La pondération obtenue permet d'ordonner les variables en fonction de leur pertinence et peut être utilisée comme critère d'évaluation dans une approche filtre de sélection de variables.

7.2 Approche Proposée

7.2.1 Algorithme w-kmeans

Commençons par introduire les notations utilisées dans ce chapitre :

- $U = (u_{ik})$ est une matrice binaire où $\begin{cases} 0, \text{si } x_i \notin \mathcal{C}_k \\ 1, \text{si } x_i \in \mathcal{C}_k \end{cases}$
- $W = (\omega_1, \omega_2, \ldots, \omega_n)$ est le vecteur qui regroupe le poids des différents attributs,
- $Z = \{z_k \in \mathbf{R}^n : k = 1, \ldots, C\}$ est l'ensemble des centres.

L'algorithme *w-kmeans* proposé par Huang [HNRL05] optimise la fonction de coût suivante :

$$P(U, Z, W) = \sum_{i=1}^{N} \sum_{j=1}^{n} \sum_{k=1}^{C} u_{ik}\, \omega_j^{\beta}\, (x_{ij} - z_{kj})^2 \tag{7.1}$$

où β est un paramètre de la pondération appelé coefficient ou exposant de discrimination. Rappelons que lorsque β tend vers 1 par valeur supérieure, les ω_j^{β} tendent vers 0 ou vers 1 et on retrouve le cas de la sélection de variables.

La minimisation de $P(U, Z, W)$ est possible en itérant la minimisation des trois sous problèmes suivants [HNRL05] :

1. Minimiser $P\left(U, \hat{Z}, \hat{W}\right)$ en fixant $Z = \hat{Z}$ et $W = \hat{W}$: chaque objet x_i est affecté au centre z_j dont il est le plus proche au sens de la distance euclidienne pondérée par $\hat{W}$.

$$u_{ik} = \begin{cases} 1, \text{si } \; k = \underset{l=1,\ldots,K}{\arg\min} \sum_{j=1}^{n} \hat{\omega}_j^{\beta} \; (x_{ij} - \hat{z}_{lj})^2 \\ 0, \text{sinon} \end{cases} \tag{7.2}$$

2. Minimiser $P\left(\hat{U}, Z, \hat{W}\right)$ en fixant $U = \hat{U}$ et $W = \hat{W}$: chaque centre est remplacé par le barycentre de l'ensemble des objets qui lui sont affectés.

$$z_k = \frac{1}{\sum_{i=1}^{N} \hat{u}_{ik}} \times \sum_{i=1}^{N} \hat{u}_{ik} \; x_i \tag{7.3}$$

3. Minimiser $P\left(\hat{U}, \hat{Z}, W\right)$ en fixant $U = \hat{U}$ et $Z = \hat{Z}$: Huang montre qu'on minimise ce problème de la manière suivante

   Lorsque $\beta \neq 1$, la fonction de coût $P\left(\hat{U}, \hat{Z}, W\right)$ est minimisée si et seulement si

$$\omega_j = \begin{cases} 0, & si \; D_j = 0 \\ \left[\sum_{t=1}^{m} \left(\frac{D_j}{D_t}\right)^{\frac{1}{\beta-1}}\right]^{-1} & si \; D_j \neq 0 \end{cases} \tag{7.4}$$

$$\text{avec } \; D_j = \sum_{l=1}^{C} \sum_{i=1}^{n} \hat{u}_{il} \; d(x_{ij}, \hat{z}_{lj}) \tag{7.5}$$

   Lorsque $\beta = 1$, la fonction de coût $P\left(\hat{U}, \hat{Z}, W\right)$ est minimisée si et seulement si

$$\begin{cases} \hat{\omega}_{j'} = 1, & si \; (\forall j) \left(D_{j'} \leq D_j\right) \\ \hat{\omega}_j = 0 & sinon \end{cases} \tag{7.6}$$

7.2.2 Extension aux cartes auto-organisatrices

L'approche proposée par Huang peut être étendue aux cartes auto-organisées en introduisant une contrainte de préservation de la topologie locale de l'espace des données ; la fonction de coût modifiée fait intervenir une fonction de voisinage et s'exprime ainsi :

$$P\left(U, Z, W\right) = \sum_{k=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{n} u_{ik} \; \omega_j^{\beta} \; \sum_{l=1}^{C} h_{kl} \; d(x_{ij} - z_{lj}) \tag{7.7}$$

où h_{kl} est la fonction de voisinage entre les prototypes. Le théorème proposé ainsi que sa démonstration restent valables en modifiant seulement la définition de D_j de la façon suivante :

$$D_j = \sum_{k=1}^{C} \sum_{i=1}^{n} \hat{u}_{ik} \sum_{l=1}^{C} h_{kl} \; d(x_{ij} - \hat{z}_{lj}) \tag{7.8}$$

7.2.3 Utilisation pour la sélection de variables

Huang suggère d'utiliser la pondération obtenue par l'algorithme w-kmeans comme critère d'évaluation dans une procédure de sélection de variables. Dans cet esprit, nous proposons de définir une approche filtre basée sur une stratégie d'élimination arrière guidée par la pondération calculée par optimisation de notre fonction de coût étendue (7.7). La procédure s'arrête lorsque la suppression de la variable modifie de manière significative la topologie locale de l'espace des données. L'évaluation de la pertubation induite par la suppression d'une variable peut se faire en testant l'hypothèse nulle suivante :

"On n'observe pas de différence significative entre les distances d'une observation à son référent après la suppression de la variable."

à l'aide d'un test de Wilcoxon. Ce test statistique est un test non paramétrique : il ne fait pas d'hypothèse sur la distribution des valeurs des deux échantillons que l'on souhaite comparer. Il repose sur le principe suivant : si on rassemble deux échantillons tirés d'une même population et qu'on ordonne les individus, alors ils s'intercalent de façon régulière. On calcule alors la statistique de Wilcoxon W_x qui est définie comme la somme des rangs des individus du premier échantillon. Les valeurs de cette statistique sont tabulées pour de petits échantillons et on dispose d'un théorème qui dit que la distribution de

$$\frac{W_x - n_x\,(n_x + n_y + 1)/2}{\sqrt{n_x\,n_y\,(n_x + n_y + 1)/12}} \tag{7.9}$$

où n_x et n_y désignent respectivement la taille du premier et du second échantillon, converge vers une loi normale $N(0, 1)$ lorsque les échantillons ont des tailles suffisantes.

7.3 Evaluation

7.3.1 Données

L'université de Californie à Irvine (UCI) met à la disposition de la communauté d'apprentissage artificiel de nombreux jeux de données pour valider leurs approches [DNM98]. Nous en avons retenu quatre de taille et de complexité variables pour valider notre algorithme :

- **Iris :** Ce jeu de données, à l'origine proposé par Fisher, est l'un des plus connus dans le domaine de la reconnaissance de formes. Il contient 3 classes de 50 instances qui correspondent chacune à une espèce d'iris : setosa, versicolor et virginica. L'une des classes est linéairement séparable des autres qui se chevauchent. Chaque fleur est décrite par les dimensions de ses pétales et sépales.
- **Glass :** Cette base contient les caractéristiques de 214 échantillons de verres suivantes : indice de réfraction, oxyde de sodium, oxyde magnésium, oxyde d'aluminium, oxyde de silicium, oxyde de potassium, oxyde de calcium, oxyde de baryum et oxyde de fer. Les différentes instances se répartissent dans les 7 classes suivantes : 70 dans la classe 1 (verre traité utilisé en construction), 76 dans la classe 2 (verre traité utilisé dans les véhicules), 17 dans la classe 3 (verre non traité utilisé en construction), 0 dans la classe 4 (verre non traité utilisé dans les véhicules), 13 dans la classe 5 (bocaux), 9 dans la classe 6 (vaisselle) et 29 dans la classe 7 (tête d'ampoule). La classe 4 n'étant pas représentée, on peut considérer qu'il s'agit d'un problème à 6 classes.
- **Waveform :** Ce jeu de données artificielles comporte 5000 exemples répartis en trois classes obtenues par combinaison de deux des trois "vagues de base" et ajout d'un bruit gaussien de moyenne nulle et de variance 1 à chacune des 21 variables originales. Dans leur version bruitée, les vagues de Breiman comportent 19 dimensions supplémentaires qui suivent une loi normale de moyenne nulle et de variance 1.

- **Wine :** Cette base recense les résultats d'une analyse chimique de différents vins produits à dans une même région d'Italie à partir de différents cépages. La concentration de 13 constituants est indiquée pour chacun des 178 vins analysés qui se répartissent ainsi : 59 dans la classe 1, 71 dans la classe 2 et 48 dans la classe 3.

Les jeux de données décrits ci-dessus contiennent de 150 à 5000 instances décrites par 4 à 40 variables. Nous souhaitions également montrer que notre algorithme est adapté aux données de dimension supérieure ; à cet effet, nous avons utilisé un jeu de données parmi ceux proposés lors de la compétition NIPS 2003 sur la sélection de variables pour la discrimination :

- **Madelon :** Cette base de données artificielles comporte 2000 instances réparties en deux classes équiprobables et qui sont décrites par 500 variables dont seulement 20 sont pertinentes. Les 480 attributs restants ont des distributions similaires mais n'apportent aucune information quant à la classe des exemples.

7.3.2 Résultats

Pour évaluer l'algorithme présenté au début de ce chapitre, nous avons réalisé dix simulations pour chacun des cinq jeux de données et pour des valeurs du paramètre β variant de 0 à 10 à l'exception de la valeur 1. Nous présentons les résultats obtenus ci-dessous.

7.3.2.1 Stabilité de l'algorithme de pondération

Le tableau 7.1 montre la faible dispersion des valeurs de la fonction objectif après la convergence de l'algorithme.

Données	SOM	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$	$\beta = 6$	$\beta = 7$	$\beta = 8$	$\beta = 9$	$\beta = 10$
Iris	5.82	3.41	2.84	2.33	2.43	2.50	2.12	4.37	2.53	4.96
Glass	4.00	1.59	3.72	1.33	3.68	3.16	4.33	3.50	4.12	4.57
Waveform	0.22	0.12	0.09	0.06	0.10	0.07	0.07	0.09	0.10	0.13
Wine	1.22	3.14	1.52	1.76	1.25	1.47	1.64	2.45	1.79	1.35
Madelon	0.56	1.23	0.53	0.81	0.81	0.36	0.63	0.98	0.40	0.59

Table 7.1 – Indice de dispersion $\sigma/\overline{x}$ ($\times 100$) de la fonction objectif pour 10 exécutions de l'algorithme ω^β-SOM

Les tableaux 7.2, 7.3 et 7.4 montrent respectivement les indices de dispersion des poids de chaque attributs après convergence de l'algorithme pour les jeux de données *Iris*, *Glass* et *Wine*. On note globalement une plus grande stabilité des pondérations calculées que celle rapporté dans [HNRL05] pour l'algorithme *w-kmeans*.

7.3.2.2 Pertinence et stabilité du sous-ensemble de variables sélectionnées

Sur les "vagues de Breiman", notre approche sélectionne de manière systématique les variables 4 à 18 et la sélection des variables 3 et 19 dépend uniquement de la valeur du paramètre β. Ainsi, la méthode proposée permet d'une part d'éliminer le bruit gaussien additionnel et d'autre part d'identifier les variables qui sont habituellement reconnues comme pertinentes par des techniques supervisées comme *Optimal Cell Damage (OCD)* ou *Heuristic for Variable Selection (HVS)* [Ben01]. Le bruit gaussien est

β	x_1	x_2	x_3	x_4
2	8.06	7.02	1.44	2.95
3	3.79	3.71	0.30	1.79
4	2.34	1.70	0.17	1.86
5	1.53	1.33	0.10	0.82
6	1.31	1.34	0.29	0.57
7	0.87	0.68	0.24	0.57
8	1.31	1.25	0.26	0.39
9	0.74	0.76	0.23	0.31
10	0.79	0.83	0.27	0.30

Table 7.2 – Indice de dispersion $\sigma/\overline{x}$ ($\times 100$) des poids des attributs pour la base IRIS au cours de 10 exécutions.

β	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
2	14.51	10.40	18.25	10.00	4.55	18.51	16.81	11.51	36.37
3	2.35	5.40	8.25	4.01	5.54	14.02	6.48	5.87	9.58
4	1.04	2.24	2.38	1.20	2.17	6.81	2.82	2.59	3.52
5	0.70	1.89	4.88	1.53	1.67	5.29	3.80	2.09	6.05
6	0.79	1.18	2.04	1.20	1.31	3.98	2.15	2.04	2.93
7	0.33	0.77	2.28	0.58	1.31	3.25	1.92	1.53	1.73
8	0.97	1.35	2.12	0.82	1.34	3.08	2.00	1.58	2.14
9	0.65	0.95	1.81	0.48	0.40	2.23	1.05	0.88	1.79
10	0.73	0.72	1.85	0.56	1.12	2.15	1.32	1.73	0.98

Table 7.3 – Indice de dispersion $\sigma/\overline{x}$ ($\times 100$) des poids des attributs pour la base GLASS au cours de 10 exécutions.

β	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}
2	9.35	3.24	6.04	3.24	4.50	4.59	8.19	4.35	2.79	18.39	2.20	2.93	7.51
3	0.67	1.77	6.03	4.40	4.66	2.23	2.18	2.64	3.85	10.66	1.24	1.23	2.11
4	1.16	1.34	4.63	3.44	3.12	1.45	1.67	1.56	2.87	6.59	0.81	1.31	1.05
5	0.98	1.04	2.63	2.01	1.91	1.07	1.06	1.36	1.98	3.76	0.65	0.68	0.59
6	1.23	0.75	2.73	1.78	1.58	0.64	0.97	0.74	0.80	3.46	0.52	0.70	0.69
7	0.71	0.67	2.26	1.54	0.99	0.68	1.07	0.80	1.25	3.17	0.45	0.60	0.68
8	0.35	0.47	1.78	1.43	0.80	0.37	0.56	0.91	1.42	2.64	0.36	0.36	0.21
9	0.79	0.45	1.97	1.29	0.97	0.43	0.83	0.76	0.52	2.49	0.28	0.53	0.56
10	0.45	0.32	1.59	0.98	0.83	0.44	0.55	0.54	0.58	2.01	0.36	0.53	0.25

Table 7.4 – Indice de dispersion $\sigma/\overline{x}$ ($\times 100$) des poids des attributs pour la base WINE au cours de 10 exécutions.

généralement considéré comme “facile à détecter”, nous avons donc répété nos simulations en remplaçant les dimensions 22 à 40 par des permutations des variables 1 à 21. Le même comportement de la méthode a été observé dans ces conditions moins favorables.

Sur la base *madelon* dont seulement 4% des dimensions sont pertinentes, notre approche sélectionne 12 variables de manière systématique et jusqu'à 5 variables supplémentaires ; toutes correspondent effectivement à des dimensions intéressantes.

7.4 Discussion

7.4.1 Pondération

Dans ce chapitre, nous avons proposé une méthode de pondération pour les cartes auto-organisées ; cette méthode s'est révélée efficace pour détecter le bruit et identifier les variables pertinentes. Néanmoins, nous avons utilisé une pondération globale qui suppose que les attributs pertinents sont les mêmes pour tous les groupes d'individus présents au sein de l'ensemble d'apprentissage. Cette hypothèse nous semble forte et nous pensons qu'il serait intéressant d'étudier l'extension de l'algorithme proposé au cas d'une pondération locale. Ce type d'approche permettrait en outre de faciliter la caractérisation des groupes identifiés et ainsi de faciliter la compréhension des données.

7.4.2 Critère d'arrêt

Le test statistique (Wilcoxon) utilisé dans notre algorithme de classification non supervisée pour faire la sélection est basé sur l'analyse des rangs des distances des données par rapport aux prototypes de la carte sans tenir compte ni de la notion de groupes ni de leur répartition. En outre, ce test s'est révélé inadapté sur la base de données *Iris* pour laquelle aucun attribut n'a été éliminé. Il serait opportun d'utiliser un autre type de test (par exemple le test de Fisher) en analysant plutôt la variance intra et inter clusters. Dans ce dernier cas, le test permettra de sélectionner les variables pertinentes minimisant par exemple le rapport variance intra/variance inter clusters.

7.4.3 Approche intégrée

Au cours de ce chapitre, nous avons proposé une approche filtre de sélection de variables et il semblerait intéressant de modifier la méthode proposée pour en faire une approche intégrée. Nous avons mentionné au début du chapitre qu'on peut montrer que lorsque $\beta \rightarrow 1_+$, les poids ω_j^β tendent à devenir binaires. Ainsi, nous envisageons de modifier le paramètre β pendant l'apprentissage, au même titre que la taille du voisinage qui diminue progressivement.

7.5 Conclusion

Une extension de l'algorithme *w-kmeans* proposé par Huang [HNRL05] a été présentée ; elle permet notament d'apprendre progressivement une pondération qui peut être utilisée dans une procédure de sélection de variable. L'intérêt majeur de commencer par un calcul de pondération est de permettre l'émergence progressive d'une partition de l'ensemble des observations et d'éviter que la suppression par erreur d'une variable pertinente en début d'apprentissage n'empêche la découverte d'une structure intéressante. Les résultats obtenus par cette méthode sont très encourageant et nous pensons qu'il serait opportun de l'étendre au cas d'une pondération locale qui permettrait en outre de faciliter l'interprétation

des groupes mis en évidence. Ensuite, le critère d'arrêt que nous retenu a été mis en défaut sur la base des *Iris* et il conviendrait d'évaluer les performances de notre approche en le remplaçant par un T-test. Enfin, la méthode proposée reste coûteuse d'un point de vue computationnel et le développement d'une approche intégrée dans laquelle la valeur du paramètre β diminuerait progressivement mérite d'être étudier, car il permettrait certainement d'améliorer ce point critique.

PARTIE III

Applications

CHAPITRE 8

Applications aux traitements de données comportementales

8.1 Application aux Marketing

8.1.1 Problématique

Dans un contexte économique toujours plus concurentiel, une entreprise qui souhaite perdurer et se développer doit savoir adapter sa stratégie aux évolutions de son marché. Pour y parvenir, ses décideurs ont à leur disposition différents outils dont l'analyse des résultats de sondage auprès de consommateurs qui est abordé dans la suite de cet article. Les objectifs d'une enquête peuvent se résumer en trois questions :

- "Qui ?" : Connaître ses clients et ses prospects est essentiel pour déterminer les canaux de communications à utiliser par exemple.
- "Quoi ?" : Identifier les produits qui les intéressent et leurs attentes permet d'adapter sa gamme pour toujours mieux les satisfaire ?
- "Pourquoi ?" : Cette question est sans nul doute à la fois la plus intéressante et la plus difficile. Il s'agit en effet de comprendre le comportement de nos clients et prospects afin de prendre les décisions les mieux adaptées.

Dans ce qui suit, nous allons montrer comment les méthodes connexionnistes peuvent être mises en œuvre pour mettre en évidence la structure d'un marché et permettre aux décideurs de se concentrer sur la dernière question.

8.1.2 Collecte des données

La qualité des données collectées auprès des consommateurs conditionne la validité et la pertinence des conclusions d'une analyse de marché. Il est ainsi essentiel d'apporter le plus grand soin aux différentes étapes du recueil de données.

8.1.2.1 Rédaction du questionnaire

La rédaction du questionnaire est un compromis difficile entre différentes exigences contradictoires. D'un coté, le spectre des questions retenues doit être suffisament large pour couvrir l'ensemble des caractérisques du marché étudié. De l'autre, la longueur du questionnaire doit rester raisonnable pour limiter les coûts de collecte des réponses. Signalons par ailleurs que l'on distingue généralement deux

types de questions en fonction des réponses autorisées ; ainsi, on parle de questions fermées si la liste des réponses possibles est fixée et de question ouvertes lorsque la personne interrogée est libre de formuler sa réponse comme elle le souhaite. Dans le cadre de cet article, nous ne traiterons pas de ce dernier type de questionnaire dont l'étude est l'objet d'un champs de recherche à part entière en statistique.

8.1.2.2 Définition de la population cible

Généralement réalisée conjointement avec la rédaction du questionnaire, la définition de la population cible nécessite un certain niveau d'expertise pour assurer la validité des résultats des analyses menées à partir de l'échantillon de la population retenu.

8.1.2.3 Collecte des réponses

La collecte des données peut être réalisée de différentes manières : par des enquêteurs, par courrier, par téléphone ou encore par voie électronique. Chaque canal de communications apporte ses biais propres qui doivent être pris en considération lors de la phase d'analyse des données. Ainsi, un enquêteur peut apparaître plus ou moins sympathique à une personne interrogée et celà peut introduire un biais dans les données receuillies.

Il est important de noter que malgré l'ensemble des précautions prises pendant la phase de collecte des réponses, celles-ci demeurent entâchées de nombreux biais liés par exemple au contexte de receuil des données ou à l'actualité. Imaginons un instant que l'on réalise un sondage sur la consommation de viande et qu'une nouvelle infection touchant certains animaux d'élevage soit annoncée au journal télévisé en plein milieu de la période de receuil des réponses. Il y a fort à parier que les réponses des consommateurs interrogés avant et après cette annonce diffèrent de façon significative. Par ailleurs, le sens d'une même réponse varie d'un individu à l'autre. Ainsi, la prise en compte de ces différentes considérations lors du codage des données recueuillies est un élément de succès déterminant d'une étude de marché.

8.1.3 Codage des réponses

Comme nous l'avons souligné dans le paragraphe précédent, le traitement des questions ouvertes sort du cadre de cet article et nous nous intéressons dans ce qui suit qu'au codage des réponses aux questions fermées. Les questions posées lors d'une enquêtes attendent soit des réponses quantitatives, soit des réponses qualitatives. Dans ce dernier cas, si les différentes modalités peuvent être ordonnées on parle de valeur ordinale et le cas échéant on parle de valeur nominale.

8.1.3.1 Codage des valeurs nominales

Pour un certain nombre de questions fermées, les différentes modalités de réponses ne peuvent pas être facilement ordonnées. A titre illustratif, considérons les différents statuts maritals possibles d'une personne suivants : “Célibataire”, “Divorcé”, “Marié”, “Veuf” ou “Vie maritale”. Il est difficile d'ordonner ces différentes modalités et dans ce cas de figure, on utilise souvent un codage binaire disjonctif ; comme l'illustre le tableau 8.1, une variable logique est affectée à chaque modalités et est fixée à 1 si la modalité correspond à la réponse de l'individu et à 0 sinon. Le passage du cadre de la logique classique à celui de la logique floue offre davantage de souplesse et permet notament de modéliser les incertitudes

concernant les réponses des personnes interrogées. Ainsi, plutôt que d'initialiser la variable correspondant à une modalité différente de celle choisie par le répondant, elle est initialisée avec une valeur de l'intervalle $[0; 1]$ correspondant aux incertitudes liées à sa réponse.

Modalité	Codage disjonctif
"Célibataire"	$\langle 1; 0; 0; 0\rangle$
"Marié"	$\langle 0; 1; 0; 0\rangle$
"Veuf"	$\langle 0; 0; 1; 0\rangle$
"Vie maritale"	$\langle 0; 0; 0; 1\rangle$

Table 8.1 – Exemple de codage binaire disjonctif d'une valeur nominale.

8.1.3.2 Codage des valeurs ordinales

Le codage proposé ci-dessus pourrait être appliqué à des valeurs ordinales mais au prix d'une perte d'information importante. A titre d'exemple, considérons la liste suivante : "Certainement", "Probablement", "Peut-être", "Probablement pas" ou "Certainement pas". En faisant abstraction de l'ordre, les réponses "Certainement" et "Probablement" sont considérées comme aussi différentes que "Certainement" et "Certainement pas". Cet exemple simple montre l'inéquation du codage binaire disjonctif dans le cas des valeurs ordinales ; néanmoins, le passage à la logique floue permet d'obtenir un premier codage satisfaisant si on dispose de l'expertise nécessaire pour fixer les différents coefficients ; un exemple en est donné par le tableau 8.2.

Modalité	Codage disjonctif flou
"Certainement"	$\langle 1,0\ ;\ 0,8\ ;\ 0,4\ ;\ 0,2\ ;\ 0,0\rangle$
"Probablement"	$\langle 0,8\ ;\ 1,0\ ;\ 0,8\ ;\ 0,4\ ;\ 0,2\rangle$
"Peut-être"	$\langle 0,4\ ;\ 0,8\ ;\ 1,0\ ;\ 0,8\ ;\ 0,4\rangle$
"Probablement pas"	$\langle 0,2\ ;\ 0,4\ ;\ 0,8\ ;\ 1,0\ ;\ 0,8\rangle$
"Certainement pas"	$\langle 0,0\ ;\ 0,2\ ;\ 0,4\ ;\ 0,8\ ;\ 1,0\rangle$

Table 8.2 – Exemple de codage disjonctif flou d'une valeur ordinale.

Le codage des réponses à l'aide d'une variable numérique qu'illustre le tableau 8.3 est sans doute le plus simple que l'on puisse imaginé. Néanmoins, il fait l'hypothèse forte d'une différence constante entre deux modalités successives. On pourra bien entendu adapter la différence entre les modalités en faisant appel à un expert du domaine.

Enfin, en modifiant légèrement la sémantique des variables du codage disjonctif associées aux différentes modalités on obtient le codage binaire dit "additif". Pour illustrer notre propos, si on considère la troisième et la quatrième variable logique dans le tableau 8.4, elles ont respectivement les sémantiques suivantes "Au moins peut-être" et "Au moins probablement".

8.1.4 Exemple d'étude

Nous présentons ici un exemple d'étude portant sur les intentions d'achats et les attentes exprimées par un panel original d'un millier de consommateurs. Le questionnaire utilisé lors de la collecte des

Modalité	Codage numérique
“Certainement”	5
“Probablement”	4
“Peut-être”	3
“Probablement pas”	2
“Certainement pas”	1

Table 8.3 – Exemple de codage numérique d’une valeur ordinale.

Modalité	Codage additif
“Certainement”	$\langle 1;1;1;1;1\rangle$
“Probablement”	$\langle 1;1;1;1;0\rangle$
“Peut-être”	$\langle 1;1;1;0;0\rangle$
“Probablement pas”	$\langle 1;1;0;0;0\rangle$
“Certainement pas”	$\langle 1;0;0;0;0\rangle$

Table 8.4 – Exemple de codage binaire additif d’une valeur ordinale.

données portait sur une centaine de produits et autant d’attentes ou besoins. Afin de garantir la confidentialité des données stratégiques utilisées dans le cadre de cet exemple, les réponses d’une partie des consommateurs ont été retirées et le nom des produits ainsi que l’intitulé des attentes ont été modifiés

8.1.4.1 Choix du codage des réponses

Pour cette étude, nous avons retenu un codage numérique des réponses aux questions portant sur les intentions d’achats ou sur les attentes. Pour les réponses aux questions signalétiques, un codage binaire additif a été employé chaque fois que les modalités étaient ordonnées et un codage binaire disjonctif dans les autres cas.

8.1.4.2 Pré-traitement des données

Le tableau de données obtenu après le codage des réponses a ensuite été pré-traité avant de commencer son analyse. Les réponses aux questions portant sur les intentions d’achat ou sur les attentes sont très liées à l’échelle individuelle de notation utilisée par chacune des personnes interrogées. Pour limiter le biais résultant, les réponses ont été centrées par individus ; ce qui revient à s’intéresser aux préférences des consommateurs plutôt qu’aux réponses brutes.

8.1.4.3 Segmentation des consommateurs

L’apprentissage d’une carte topologique a été réalisé en utilisant la version séquentielle de l’algorithme de Kohonen. Celle-ci a ensuite été segmentée à l’aide de l’algorithme de k-moyennes pour un nombre de centres variant de 2 à $\sqrt{M}$, où M est le nombre de neurones sur la carte. Les indices de Davies-Bouldin indiqués sur la figure 8.1 pour les versions hors-ligne et en-ligne de l’algorithme correspondent aux meilleures valeurs obtenues après dix exécutions. La version *fast-global-kmeans* est déterministe et n’a donc été éxécutée qu’une seule fois. La meilleure segmentation, obtenue avec la méthode

fast-global-kmeans pour douze centres, est présentée à la figure 8.2. Chaque segment de consommateurs

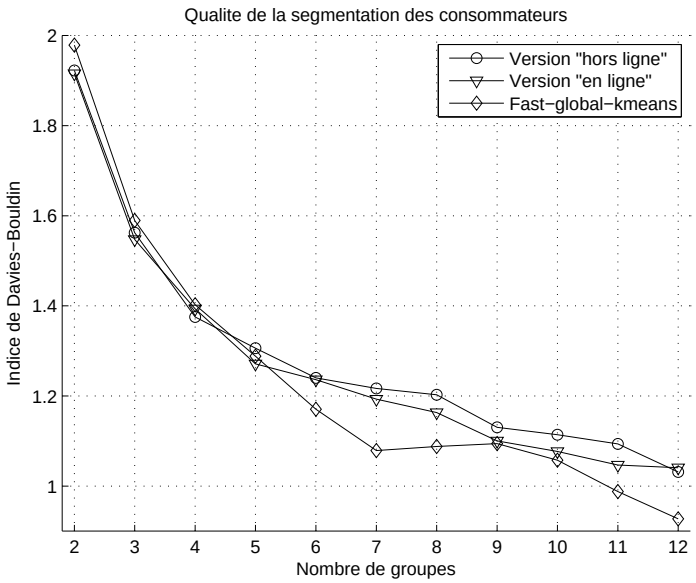

Figure 8.1 – Qualité des segmentations des consommateurs en fonction du nombre de groupes et de l'algorithme utilisé.

identifié peut ensuite être caractérisé par un sous-ensemble de variables dont les valeurs sont caractéristiques du groupe considéré. On peut utiliser la *valeur test* qui est un indicateur statistique proposé par A. Morineau [Mor84] à cette fin ; elle sera présentée au paragraphe 8.2.4.5.

8.1.4.4 Segmentation des attentes, produits et informations signalétiques

Des profils de produits, d'attentes et de variables signalétiques ont ensuite été extraits à partir de la carte des consommateurs. En effet, les prototypes des unités représentent des consommateurs moyens et le vecteur des valeurs d'une variable (produit, attente ou signalétique) sur l'ensemble des unités en donne un profil représentatif [VA99]. Les représentations ainsi obtenues ont été utilisées pour construire une carte des produits, attentes et variables signalétiques dont les projections sont représentées à la figure 8.3.

La carte obtenue a ensuite été segmentée en suivant le protocole décrit précédement ; le meilleur découpage est cette fois obtenu pour huit segments à l'aide de la version en-ligne de l'algorithme des k-moyennes (cf. figure 8.4). L'interprétation des segments mis en évidence (cf. figure 8.5) est très intuitive lorsque l'on dispose des noms de variables originales, mais le caractère stratégique de ce type de données ne nous permet pas de la détailler davantage ici.

8.1.5 Conclusion

Nous avons présenté ici une méthode systématique d'analyse de données issues d'enquêtes auprès de consommateurs où le questionnaire ne comporte que des questions fermées. Lorsqu'elle est utilisée pendant la phase exploratoire de l'analyse, cette approche permet de dégager rapidement les premiers éléments de compréhension d'un marché et de répondre aux deux premières questions fondamentales :

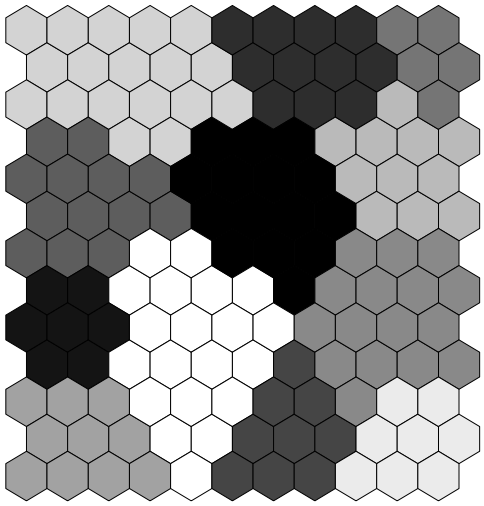

Figure 8.2 – Segmentation de la carte des consommateurs en douze segments.

"Qui ?" et "Quoi ?". Cette première perception du marché permet ensuite d'aborder le "Pourquoi ?" et d'ainsi appréhender les comportements de nos consommateurs. Ce n'est qu'alors, que des modifications de la stratégie marketing pourront être envisagées sereinement.

8.2 Application à l'Ethologie

8.2.1 Problématique

La vie sociale dans les groupes structurés implique une régulation constante des relations entre membres du groupe. La signalisation par l'individu de son appartenance à des sous-groupes (genre, statut, etc.) est un moyen de réguler les interactions. Par ailleurs, dans diverses situations sociales, la signalisation du genre est de loin la plus importante. Plusieurs parties du corps sont utilisées dans le signalement du genre et des études antérieures [Ber91, BBH$^+$93, BY98] ont montré que le visage était une région importante. Plusieurs auteurs ont mis en évidence la rôle des mouvements faciaux dans la catégorisation du genre [HJ01, TK02].

Curieusement, peu d'études ont été consacrées à la production des mouvements faciaux et à leur organisation temporelle [ADMR05, GTPF03, TGPF05]. De plus, l'utilisation de systèmes d'enregistrements nécessitant la pose de marqueurs sur le visage tend à rendre la situation peu naturelle. Notre étude s'intéresse à une situation expérimentale faisant intervenir de jeunes adultes confrontés à une tâche cognitive ne nécessitant pas d'interaction sociale directe. Les sujets ont cependant été accueillis par une expérimentatrice avant l'enregistrement vidéo de leur comportement depuis une pièce voisine. Cette situation n'est pas sociale bien que son contexte le soit. Nos objectifs étaient

1. de constituer une base de données permettant de nouvelles comparaisons entre hommes et femmes,
2. d'encoder les mouvement faciaux à l'aide d'un méthode objective,
3. de détecter et de caractériser l'organisation temporelle des mouvements faciaux,

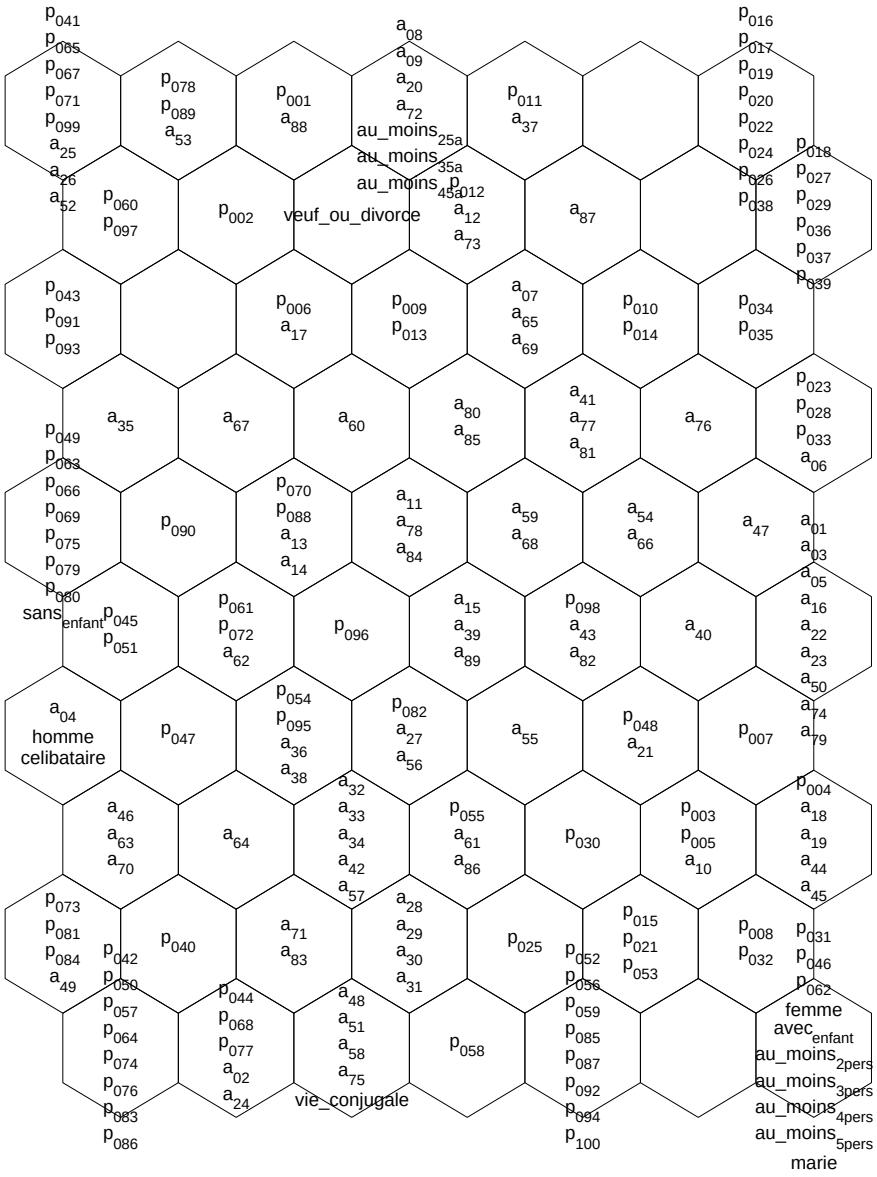

Figure 8.3 – Projection des produits, attentes et des variables signalétiques.

4. d'utiliser les même données pour mettre en oeuvre une approche connexionniste,
5. de confronter les premiers résultats des deux approches.

Nous commençons par décrire le recueil des données avant de présenter les approches éthologique et connexionniste mises en oeuvre. Nous poursuivons par une discussion générale des résultats obtenus avant d'indiquer les futurs travaux envisagés.

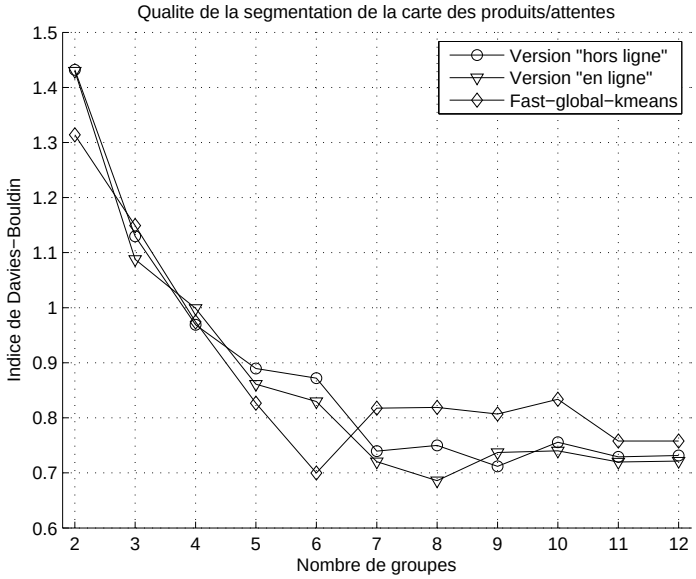

Figure 8.4 – Qualité des segmentation de la carte des produits et des attentes en fonction du nombre de groupes.

8.2.2 Constitution de la base de données

L'objectif est d'obtenir des enregistrements vidéo d'hommes et de femmes dans une situation standardisée permettant l'expression de mouvements faciaux divers : mouvements labiaux liés aux réponses verbales, réactions émotionnelles, etc. Les sujets recrutés acceptent d'être filmés mais sont naïfs quand à la thématique réelle de l'expérience à laquelle ils participent. La situation expérimentale est une tâche cognitive réalisée dans un contexte social indirect (accueil et consignes, présence de la caméra et d'un expérimentateur de sexe féminin dans la salle voisine).

Les sujets (11 femmes et 9 hommes, entre 19 et 25 ans) sont des étudiants volontaires de l'Université Paris 13 de Villetaneuse. Ils sont recrutés pour participer à une courte expérience portant sur la perception visuelle d'images qu'ils doivent juger soit normales (non ambiguës), soit anormales (ambiguës). Ils ne sont pas rémunérés.

Les sujets sont accueillis par les expérimentateurs, puis sont introduits et laissés seuls dans une salle, avec pour consigne de suivre les instructions données via un écran d'ordinateur. Le sujet s'assied sur une chaise, face à l'ordinateur portable posé sur une table. Il dispose d'une souris qui lui permet de gérer le déroulement de l'expérience (pas de temps limité). Une caméra vidéo numérique permet d'enregistrer le sujet (visage et épaules) pendant toute l'expérience. Cette caméra est placée dans une salle contiguë, derrière une vitre, et située au-dessus du niveau de l'écran d'ordinateur. Seul l'objectif du caméscope est visible par le sujet. Les réponses verbales des sujets sont enregistrées à l'aide d'un magnétophone.

La tâche des sujets consiste à visionner des images et verbaliser à voix haute une réponse quant à leur caractère anormal/ambigu ou normal/non ambigu. La dernière diapositive les informe que l'expérience est finie. La passation dure de 1 min 30 s à 4 min 30 s, selon les sujets (2 min 45 s en moyenne). En raison de caractéristiques particulières des sujets filmés (mouvements importants du corps ou de la tête, port de lunettes ou de barbe, etc.), ne sont finalement conservés pour la suite de l'étude que 10 sujets (5 femmes et 5 hommes).

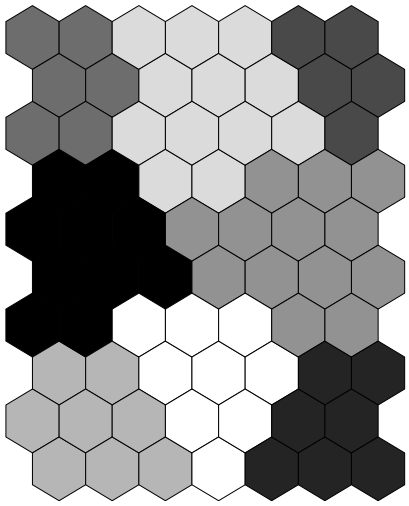

Figure 8.5 – Segmentation de la carte des produits et des attentes en huit segments.

On choisit de travailler sur des séquences de courte durée en prévision du travail de relevé ultérieur, particulièrement long. L'analyse prévue nécessitant la répétition de mouvements faciaux par un même individu, on sélectionne 3 séquences de 3 secondes chacune par sujet. Dans un souci de standardisation, les séquences sont centrées sur une réponse facile à objectiver du sujet, à savoir une réponse verbale. On échantillonne une seconde avant le début d'énonciation de la réponse, et deux secondes après. On dispose alors de 3 séquences par sujet dont le contexte est respectivement semblable d'un sujet à l'autre. Enfin, les extraits sont segmentés en images, à raison de 13 images par seconde, ce qui produit 39 images par séquence, chronologiquement indicées (analyse du mouvement à la précision de 0,08s). A ce stade, les données consistent en 10 jeux, un jeu par sujet, de 3 séries chronologiques de 39 images (soit un total de 1170 images).

8.2.2.1 Recueil des données sur des séries chronologiques d'images

Sur chaque image, le recueil des données se fait par pointage. Le principe de cette étape est d'obtenir les coordonnées successives, c'est-à-dire au fil des 39 images, d'un nombre déterminé de points du visage. L'évolution des coordonnées des points au cours du temps fournit une information quant à leur déplacement.

Nous définissons 36 points du visage, impliqués dans les mouvements faciaux [GTPF03, TGPF05] et facilement désignables. Ces 36 points sont situés au niveau des sourcils, des yeux, du nez, de la bouche et du menton. La figure 1 présente leur disposition, ainsi que leur désignation par des numéros.

On réalise 3 sessions de pointage de 3 s. par sujet (total : 117 images). Le pointage est réalisé image par image. Par exemple, on commence par pointer à l'écran, à l'aide du curseur, le point 1, de l'image 1 à l'image 39.

Par mesure de précaution et pour garantir une plus grande précision des mesures, on effectue de deux à quatre répétitions par relevé, par point et par image. Les quatre valeurs des deux coordonnées (x, y) ainsi obtenues sont moyennées, et c'est cette moyenne qui est prise en compte par la suite. A l'issue du

pointage, on connaît pour une séquence individuelle, les coordonnées prises par chacun des points sur chacune des 39 images. Ces coordonnées constituent les données brutes.

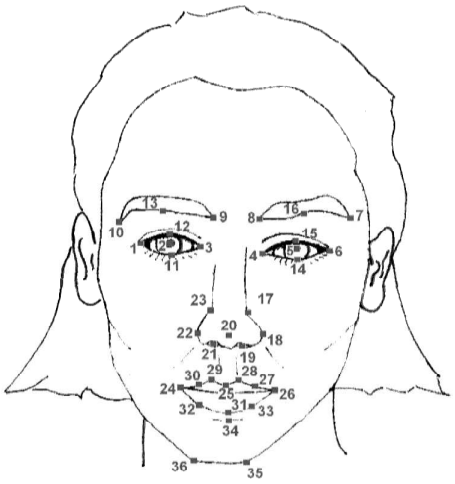

Figure 8.6 – Localisation et numérotation des points sur le visage.

8.2.2.2 Codage des données en coordonnées faciales

Les coordonnées issues du pointage, relatives à l'image, sont ensuite transformées en coordonnées faciales. On définit pour cela un nouveau repère à partir de trois points fixes du visage : le premier axe de ce repère passe par les coins internes des yeux (points 3 et 4), le second axe lui est perpendiculaire et passe par le bout du nez (point 20).

On dispose maintenant pour une séquence donnée de 3 s, des coordonnées faciales des 36 points au long des 39 images successives.

8.2.3 Approche éthologique

Pour un point du visage donné, un mouvement saillant correspond à un changement significatif de sa distance à l'origine du repère facial. Nous avons considéré qu'une différence de plus ou moins un écart type par rapport à la distance moyenne pendant la période de 3 secondes retenue était significative. Une étude comparative du nombre de mouvements saillants chez les hommes et chez les femmes est mise en oeuvre. Nous découvrons ensuite comment varient la distance à l'origine des 36 points du visage au cours de la période de 3 trois secondes chacun des genre à l'aide du logiciel THEME 5.0 développé par Magnusson (http ://www.noldus.com). Ce logiciel nous permet de détecter les motifs temporels des mouvements faciaux qui sont définis comme une répétition en temps réel de structure comportementale organisée [Mag00]. Seuls quelques résultats sont présentés ci-dessous.

Nous observons un nombre moyen de mouvements saillants produits pendant une période de 3 secondes plus important chez les hommes que chez les femmes (n=86 contre 69, $p<.05$, test de permutation exacte). Nous observons également un nombre de T-patterns plus élevé chez les hommes que chez les femmes (sur une base de 100, les hommes en produisent 66 contre 46 pour les femmes, $p=.05$). Les T-patterns impliquent en moyenne 4 points du visage chez les hommes contre 3 chez les femmes ($p=.07$).

Des différences qualitatives, liées au genre, dans la composition des T-patterns sont également mises en évidence : les hommes produisent des motifs simples impliquant l'extrémité temporale du sourcil gauche et la narine gauche, alors que les femmes produisent des motifs simples impliquant les parties interne et médiane du sourcil droit.

Nos résultats préliminaires indiquent que les mouvements faciaux des hommes et des femmes diffèrent quantitativement, et qu'au moins pour certain d'entres eux, ils diffèrent également qualitativement lors de la réalisation d'une tâche cognitive dans un contexte social.

8.2.4 Approche proposée

La base de données constituée comporte un ensemble d'observations étiquetées, deux alternatives sont donc envisageables en vue de son exploitation : l'approche supervisée et l'approche non-supervisée. Notre objectif étant de dégager une structure intrinsèque de nos données qui soit liée au genre, nous optons donc pour une approche non supervisée. La classification automatique, ou clustering, consiste à identifier des groupes d'observations similaires que nous appellerons clusters par la suite. Nous nous focalisons sur une approche de type fouille de données et nous retenons les cartes auto-organisées pour mener à bien notre analyse. Ces dernières nous fournissent un moyen pratique pour visualiser nos données sur un espace de faible dimension. Notons également que l'étiquetage de la carte obtenue nous permettra de vérifier visuellement l'émergence d'une structure liée au genre.

Nous rappelons d'abord brièvement le principe des cartes auto-organisées et une méthode découpage, avant de présenter une mesure statistique pour la caractérisation des clusters. Nous terminerons enfin par la présentation de nos résultats expérimentaux qui seront brièvement discutés.

8.2.4.1 Les cartes auto-organisées

Les cartes auto-organisées (Self-Organizing Maps ou SOM), souvent appelées cartes topologiques ou carte de Kohonen, ont été introduites au début des années 80 comme une méthode de classification automatique et de visualisation de données multidimensionnelles. Elles implémentent une forme particulière de réseaux de neurones, dits réseaux de neurones à compétition, où le succès d'un neurone de sortie (neurone de la couche de compétition) à reconnaître une entrée, conduit à inhiber les autres neurones, donc à renforcer le neurone vainqueur. Par conséquent, le neurone vainqueur pour un exemple tend à se spécialiser dans la reconnaissance de cet exemple. On note que dans ces modèles l'apprentissage est non-supervisé car ni les classes ni leur nombre n'est donné à priori. Ce type de réseau est organisé en une couche à deux dimensions (figure 8.7). Chaque neurone k est connecté à un nombre n d'entrées à travers n connexions de poids respectifs ω_k. Les connexions latérales qui assurent la compétition entre les neurones sont de poids fixes et excitatrices dans un voisinage proche.

Ces cartes s'organisent par rapport aux exemples présentés en respectant les contraintes topologiques de l'espace d'entrée. Il y a mise en correspondance de l'espace d'entrée avec l'espace du réseau. Les zones voisines de l'espace d'entrée sont voisines sur la carte auto-organisée.

Les informations reçues par le réseau neuronal déterminent un arrangement spatial optimal des neurones. Lorsque la dimension de l'espace d'entrée est inférieure ou égale à 3, il est possible de représenter visuellement la position des vecteurs poids et les relations de voisinage direct entre deux cellules. Cette présentation permet de faire une appréciation visuelle de la carte. Elle fournit une information qualitative de la carte et le choix de son architecture.

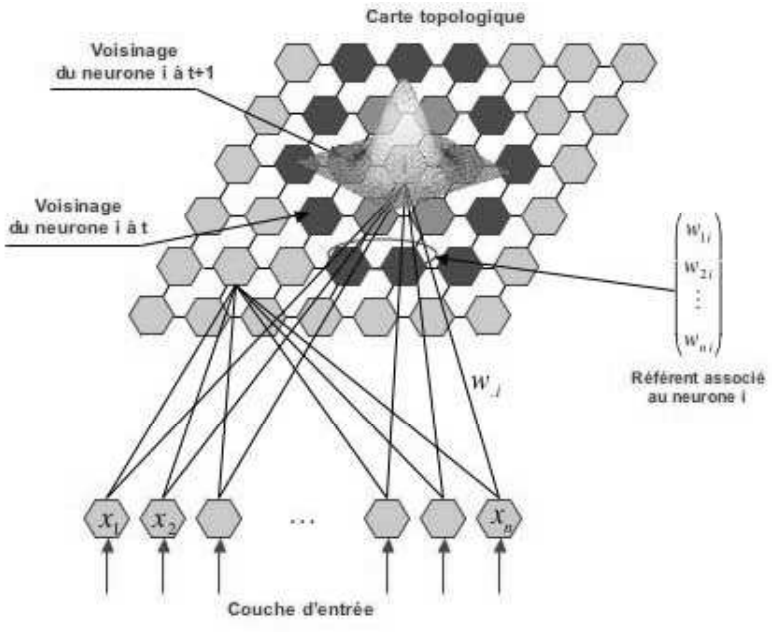

Figure 8.7 – Architecture du réseau pour l'algorithme des cartes topologiques.

8.2.4.2 Algorithme d'apprentissage

L'apprentissage connexionniste se présente souvent comme la minimisation d'une fonction de risque. Dans notre cas, il sera réalisé par la minimisation de la distance, entre exemples d'entrées et prototypes (référents) de la carte, pondérée par une fonction de voisinage h_{ij}. On pourra employer pour cela un algorithme de descente de gradient. Le critère à minimiser dans ce cas est défini par :

$$R_{SOM} = \sum_{i=1}^{N} \sum_{j=1}^{M} h_{b(i)j} \times \|x_i - \omega_j\|^2 \tag{8.1}$$

où N, M et h représentent respectivement le nombre d'exemples d'apprentissage, le nombre de neurones de la carte et la fonction de voisinage, enfin $b(i)$ est le neurone dont le référent est le plus proche de la forme d'entrée x_i . La fonction de voisinage h peut être de la forme suivante :

$$h_{rs} = \frac{1}{\lambda(t)} \exp\left(\frac{d^2(r,s)}{\lambda^2(t)}\right) \tag{8.2}$$

où $d(r,s)$ est la distance sur la carte entre les neurones r et s, et $\lambda(t)$ est la fonction température modélisant l'étendue du voisinage :

$$\lambda(t) = \lambda_i \left(\frac{\lambda_t}{\lambda_i}\right)^{\frac{t}{T_{max}}} \tag{8.3}$$

avec λ_i et λ_f sont respectivement la température initiale et la température finale (par exemple $\lambda_i = 2$ et $\lambda_f = 0,5$) et T_{max} le nombre maximum attribué au temps (nombre d'itérations x nombre d'exemples d'apprentissage), et la distance de Manhattan d_1 est définie, entre deux neurones de la carte r et s de coordonnées respectives (k,m) et (i,j) par :

$$d_1(r,s) = |i-k| + |j-m| \tag{8.4}$$

La version stochastique de l'algorithme d'apprentissage de ce modèle se déroule essentiellement en trois phases :

- la phase d'initialisation où des valeurs aléatoires sont affectées aux poids des connexions (référents ou prototypes) de chaque neurone de la carte ;
- la phase de compétition pendant laquelle, pour toute forme d'entrée x_i, un neurone $b(i)$, de voisinage $V_{b(i)}$, est sélectionné comme gagnant. Ce neurone est celui dont le vecteur de poids est le plus proche au sens de la distance euclidienne de la forme présentée :

$$b(i) = \underset{1 \leq j \leq M}{\arg\min} \|\omega_j - x_i\|^2 \tag{8.5}$$

- la phase d'adaptation où les poids de chaque neurone de la carte sont mis à jour selon les règles d'adaptation suivantes : si $\omega_{.j} \in V_{b(i)}$ ajuster les poids selon la formule :

$$\omega_{.j} \leftarrow \omega_{.j} - \epsilon h_{b(i)j}(\omega_{.j} - x_i) \tag{8.6}$$

Ce processus d'adaptation est répété jusqu'à stabilisation de l'auto-organisation.

8.2.4.3 Etiquetage de la carte

La phase d'apprentissage présentée précédemment est totalement non supervisée. Cependant, les données dont nous disposons sont étiquetées, nous pouvons utiliser cette information supplémentaire pour étiqueter les différents neurones de la carte obtenue en procédant par vote majoritaire. Ainsi, chaque neurone se voit attribuer l'étiquette majoritaire au sein de sa région de Voronoï. Il convient de noter que l'on peut améliorer la robustesse de l'étiquetage en utilisant un test du χ^2 [WW98] pour vérifier que la distribution des étiquettes parmi les observations de la région de Voronoï du neurone considéré diffère de manière significative de la distribution au sein de l'échantillon complet.

8.2.4.4 Découpage automatique

Une carte auto-organisée peut être vue comme une méthode de classification automatique dont résulte une partition de l'espace des observations qui comporte autant de partie qu'il y a de neurones. Il est souvent souhaitable de diminuer le nombre de clusters pour en faciliter l'analyse. Plusieurs méthodes de découpage automatique ont ainsi été proposée [VA00]. Nous avons retenu la méthode des k-moyennes associée à l'indice de Davies-Bouldin [DB79] pour découper notre carte.

La méthode des k-moyennes est une autre méthode de classification. Son principe consiste à choisir arbitrairement une partition. Ensuite, les exemples sont examinés un à un. Si un exemple devient plus proche du centre d'une classe autre que la sienne, il est déplacé vers cette nouvelle classe. On recalcule, ensuite, les centres des nouvelles classes et on réaffecte les exemples aux partitions, et ainsi de suite jusqu'à obtenir une partition stable.

Le critère à minimiser dans ce cas est défini par :

$$R_{K-moyennes} = \frac{1}{C} \sum_{k=1}^{C} \sum_{x \in \mathcal{C}_k} \|x - \omega_k\|^2 \tag{8.7}$$

où C , $\mathcal{C}_k$ et ω_k représentent respectivement le nombre de clusters, le cluster k et son centre.

L'algorithme initial nécessite de fixer à priori le nombre C de clusters souhaités. Néanmoins, [VA00] ont proposé de déterminer automatiquement une valeur de C en retenant la partition qui minimise l'indice de Davies-Bouldin [DB79] défini par :

$$I_{DB} = \sum_{k=1}^{K} \max_{l \neq k} \left\{ \frac{S_c(\mathcal{C}_k) + S_c(\mathcal{C}_l)}{D_{ce}(\mathcal{C}_k, \mathcal{C}_l} \right\} \tag{8.8}$$

où $S_c(\mathcal{C}_i)$ est la distance moyenne entre un objet du groupe $\mathcal{C}_i$ et son centre, et où $D_{ce}(\mathcal{C}_i, \mathcal{C}_j)$ est la distance qui sépare les centres des groupes $\mathcal{C}_i$ et $\mathcal{C}_j$:

$$\begin{aligned} S_c(\mathcal{C}_i) &= \frac{1}{|\mathcal{C}_k|} \sum_{x \in \mathcal{C}_k} \|x - \omega_k\| \\ D_{ce}(\mathcal{C}_i, \mathcal{C}_j) &= \|\omega_i - \omega_j\| \end{aligned}$$

La méthode des k-moyennes associée à l'indice de Davies-Bouldin recherche une partition de l'espace des observations dont les différentes parties sont compactes et bien séparées.

8.2.4.5 Indicateur statistique pour la caractérisation des clusters

Les différents clusters identifiés par le découpage de la carte peuvent être caractérisés à l'aide de l'indicateur statistique introduit par Morineau [Mor84] : la valeur test. Cet indicateur utilise le fait qu'une variable aléatoire qui suit la même loi au sein d'un cluster et au sein de l'échantillon dans son ensemble est sans intérêt pour caractériser ce cluster ; plus l'hypothèse d'un tirage aléatoire semble douteuse, plus pertinente sera la variable pour caractériser le cluster. La valeur test d'une variable pour le cluster $\mathcal{C}_k$ est définie ainsi :

$$t = \frac{(\mu_k - \mu)}{\sigma_k} \tag{8.9}$$

où μ, μ_k et σ_k sont respectivement la moyenne au sein de l'ensemble des observations, la moyenne et l'écart type au sein du cluster $\mathcal{C}_k$.

Sous l'hypothèse d'un tirage aléatoire sans remise des observations composant le cluster $\mathcal{C}_k$, les valeurs de la moyenne et de la variance d'une variable au sein du cluster devrait être sensiblement les même que les valeurs observées pour l'échantillon dans son ensemble. D'après le théorème central limite, la valeur test définie ci-dessus suit donc approximativement une loi de Laplace-Gauss centrée et réduite. Elle permet d'évaluer la distance entre la moyenne du cluster et la moyenne de l'échantillon en nombre d'écart type d'une loi normale.

Il convient de préciser que cette interprétation n'est valable que pour des variables illustratives. Pour les variables actives, la valeur absolue d'une valeur test ne peut être vue que comme une mesure de similarité entre une variable et un cluster.

8.2.4.6 Expérimentations

Codage des données : Notre étude s'intéresse avant tout au mouvements faciaux, nous calculons donc les distances entre deux positions successives de chacun des points. Les observations disponibles sous forme de séquences de longueurs de déplacement, sont en nombre réduit. Cela nous conduit à utiliser une fenêtre glissante pour d'une part augmenter le nombre d'observations et d'autre part, améliorer la robustesse aux décalages temporels des mouvements. Cependant, ce pré-traitement impose d'utiliser un paramètre supplémentaire : la largeur de la fenêtre W. L'étude de la dynamique des mouvements faciaux

impose que l'on intéresse aux déplacements simultanés de l'ensemble des points du visages retenus pour l'étude. Ainsi, nous avons utilisé la matrice de covariance dynamique de chacune de nos sous-séquences comme entrées du réseau. La matrice de covariance dynamique d'une séquence $S = (x_i \in \mathbf{R}^n)_{i=1,\ldots,W}$ est définie dans [ZB04b, ZB04a] de la manière suivante :

$$\Sigma_d = \frac{1}{W}\left(x_1\, x_1^T + \sum_{i=2}^{W}(x_i - \overline{x_i})\,(x_i - \overline{x_i})^T\right) \tag{8.10}$$

où la moyenne des vecteurs précédents $\overline{x_i}$ est donnée par

$$\overline{x_i} = \frac{1}{i}\sum_{j=1}^{i} x_j \tag{8.11}$$

Choix de la largeur de la fenêtre glissante : Sachant que notre objectif est d'identifier une structure intrinsèque de nos données qui soit relative au genre. Ainsi, il est pertinent de choisir une largeur de la fenêtre glissante qui permette de bien séparer les femmes des hommes. Nous évaluons donc les performances d'un classificateur basé sur une carte auto-organisée étiquetée. Nous procédons donc par validation croisée ; les données recodées de neuf des dix sujets de l'étude sont utilisées pour construire et étiqueter une carte. A titre de test, les données relatives au dernier sujet sont projetées sur la carte et l'étiquette la plus souvent rencontrée est attribuée à chacune de ses 3 séquences. L'opération est réalisée 5 fois pour chacune des valeurs possible de W. Les résultats des classificateurs dont les taux de reconnaissance de chacun des deux genres sont supérieurs à 50% sont présentés à la figure 8.9 à l'aide d'un graphe de ROC [Faw03]. Les graphes de ROC permettent de visualiser et de comparer les performance de différents classificateurs ; le meilleur d'entre eux est celui dont les performances se trouvent le plus proche du coin supérieur gauche. Dans notre cas, il s'agit du classificateur construit à partir des données recodées en utilisant une fenêtre glissante de largeur 33. Nous retenons donc cette valeur pour la suite de l'analyse.

Construction et découpage d'une carte auto-organisée : Nous construisons donc une carte auto-organisée avec les données recodées en utilisant une fenêtre glissante de largeur 33. Un découpage automatique est réalisé.

Les valeurs de l'indice de Davies-Bouldin sont données par la figure 8.8. Nous retenons le découpage en 3 classes qui minimise l'indice de Davies-Bouldin. La segmentation de la carte est donnée à gauche de la figure 8.10. La répartition des données correspondant aux 2 genres est également indiquée à droite de cette dernière figure ; le nombre de données recodées est indiqué entre parenthèses derrière le numéro de la classe. Les classes 1 et 2 correspondent respectivement aux hommes et aux femmes. Les 3 clusters obtenus par le découpage de notre carte sont étiquetés suivant la méthode présentée précédemment. Les clusters situés en haut et en bas de la carte correspondent respectivement aux femmes et aux hommes. Le cluster représenté en noir reste sans étiquette car il n'est ni clairement féminin, ni clairement masculin, il reste donc sans étiquette. La carte obtenue nous permet de conclure à l'existence d'un structure intrinsèque de nos données liée au genre.

Caractérisation des clusters obtenus : Seul les clusters "sexués" nous intéresse, nous ne caractérisons donc pas le cluster laissé sans étiquette. Dans la mesure où il nous semble plus naturel d'interpréter nos résultats à partir de séquences de déplacements qu'avec les coefficients d'une matrice de covariance,

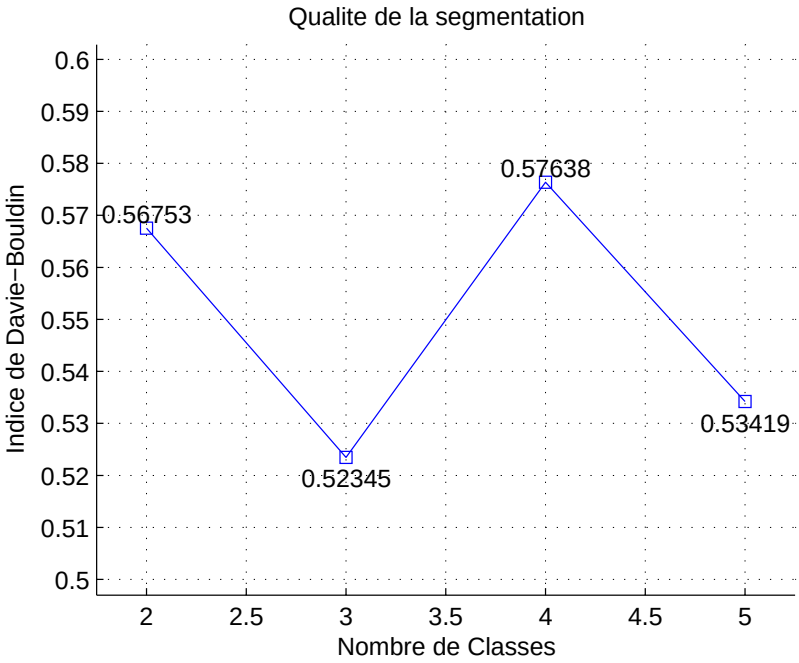

Figure 8.8 – Indice de Davies-Bouldin.

nous caractérisons les deux clusters retenus en utilisant les valeurs tests associées aux longueurs des déplacements (qui peuvent être considérées comme des variables illustratives).

Nous obtenons ainsi une valeur test par point et par déplacement qui se trouve dans la fenêtre glissante. Nous choisissons de ne représenter que les valeurs test qui sorte de l'intervalle de confiance à 95% de la moyenne de l'ensemble des valeurs test qui est donné par

$$I \equiv \mu \pm 1,96\sigma \tag{8.12}$$

où μ, σ et N sont respectivement la moyenne, l'écart type et la taille de l'échantillon. Les valeurs supérieures et inférieures à la moyenne sont représentées respectivement en haut et en bas de la figure 8.11. Les femmes et les hommes sont respectivement représentés à droite et à gauche de cette même figure. Un examen rapide de ces graphique met en évidence une plus forte structuration des mouvement chez les hommes que chez les femmes.

Point	Mouvements	Immobilité
8	0.14	0.23
17	0.14	0.19
21	0.14	0.21
33	0.14	0.28

Table 8.5 – Points caractéristiques des Hommes.

Nous souhaitons maintenant identifier les points qui permettent de différencier les deux genres. Pour cela, nous calculons pour chaque cluster et chacun des 36 points du visage retenus pour l'étude la moyenne des valeurs tests significatives. Nous retenons comme points caractéristiques d'un genre ceux

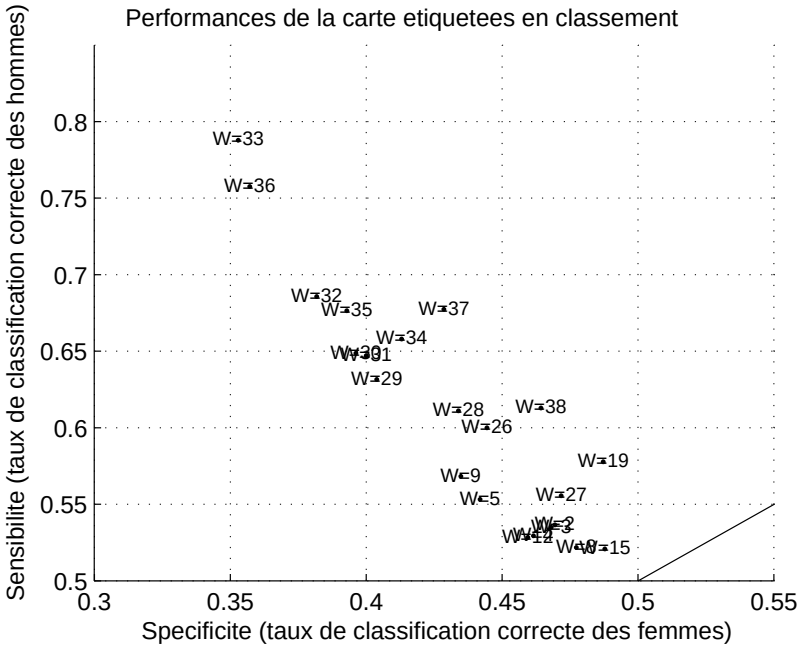

Figure 8.9 – Performances des classificateurs.

Point	Mouvements	Immobilité
1	0.06	0.10
3	0.07	0.08
6	0.07	0.09
13	0.09	0.09
21	0.08	0.10

Table 8.6 – Points caractéristiques des Femmes.

dont la valeur test sort de l'intervalle de confiance à 95% des deux cotés. Les tableaux 8.5 et 8.6 montrent les points retenus. Notons que le point 21 est présent dans les deux tableaux, nous ne le conserverons donc pas.

8.2.5 Conclusion et perspectives

Sur notre échantillon, les approches comportementale et connexionniste conduisent à des conclusions semblables : dans une tâche réalisée hors contexte social immédiat certains mouvements faciaux permettent de discriminer les hommes des femmes. Ainsi, les mouvements masculins sont localisés au niveau du sourcil, de la narine et de la lèvre inférieure gauche, alors que chez la femme ils se situent principalement au niveau du sourcil et de l'oeil droit. Cependant, on ne peut exclure que les différences quantitatives observées soient liées à la nature de la tâche et/ou de la situation. En effet les sujets savaient que les films seraient analysés par une observatrice, ce qui pourrait expliquer le biais quantitatif en faveur des hommes dans la production de mouvements faciaux.

Au plan méthodologique, la confrontation des résultats des deux approches conduit à un réexamen des périodes d'immobilité par les méthodes comportementales. Du point de vue éthologique l' existence

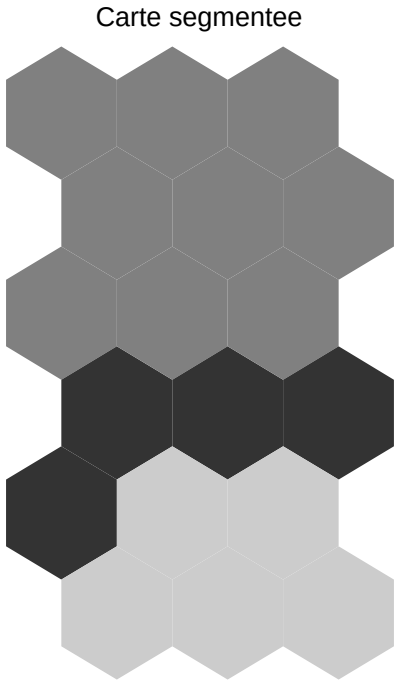

Figure 8.10 – Carte finale.

d'une latéralisation des mouvements faciaux liée au genre devra être confirmée sur un échantillon plus important et dans différentes situations. Il serait alors intéressant de comparer différents groupes culturels de façon à déterminer si les différences liées au genre sont communes à différents groupes.

D'autre part, l'approche connexionniste que nous avons utilisée s'appuie sur le modèle des cartes auto-organisées proposé au début des année 80. Ce dernier ne traite pas spécifiquement la dimension temporelle. Depuis plusieurs modèles tenant compte des spécificités des données temporelles ont été introduits [Str04, ZB04b, ZB04a]. Leur utilisation pourrait permettre une analyse plus fine de nos données.

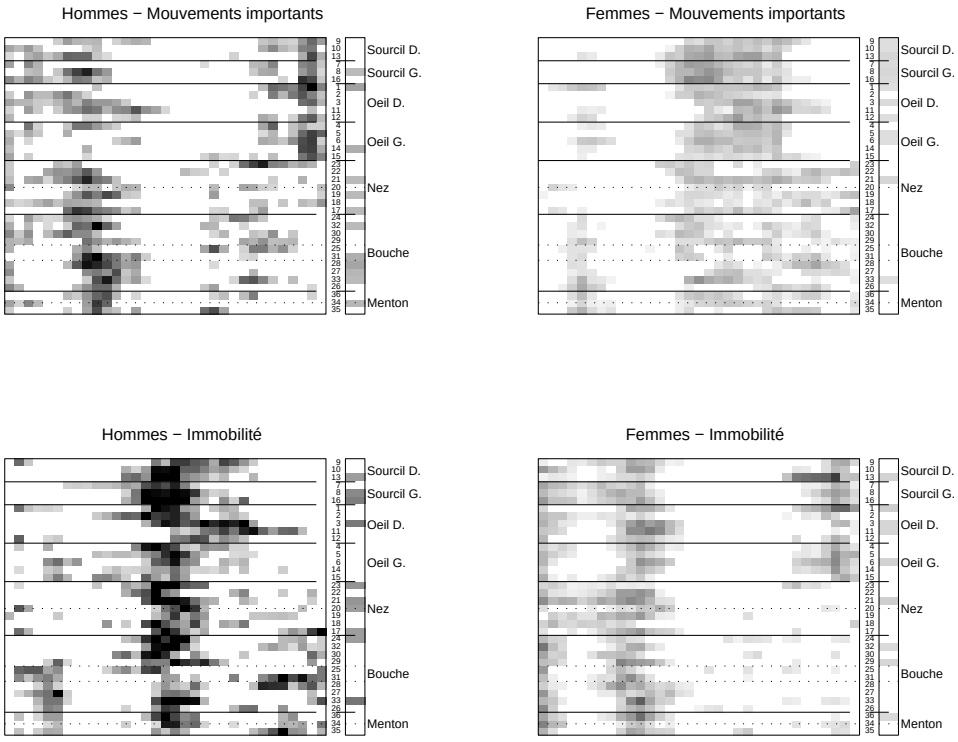

Figure 8.11 – Valeurs tests significatives.

PARTIE IV

Conclusion et perspectives

CHAPITRE 9

Conclusion et perspectives

Nous avons commencé en proposant l'algorithme μ-SOM qui s'appuie sur une classification simultanée des individus et des variables pour pondérer les variables en diminuant l'influence des dimensions redondantes. Bien que satisfaisant, les résultats obtenus nous amènent à envisager différentes améliorations :

- La mesure de similarité entre variables est un point essentiel de cette approche et mérite toute notre attention : nous envisageons de remplacer la distance euclidienne qui s'est révélée inadaptée par la distance entre les partitions qu'induisent ; nous envisageons plus précisement d'utiliser la variation d'information proposée par Marina Meilă [Mei06].
- Ensuite, nous avions retenu la version *batch* de l'algorithme de Kohonen pour l'optimisation de nos deux fonctions de coût, il conviendrait d'utiliser le formalisme lagrangien d'optimisation des systèmes modulaires introduit par Léon Bottou [BG91, Bot91] pour améliorer l'optimalité de nos solutions.

Une approche intégrée de sélection de variables et du nombre de groupes a ensuite été présentée ; son utilisation est limitée au cas le nombre de dimensions n est inférieure au nombre d'individus moins le nombre de groupes. Cette limitation est liée au critère d'arrêt retenu et il conviendrait de l'adapter pour pouvoir traiter également des données en grande dimension pour lesquelles il y a a peu d'individus ; cela est typiquement le cas en bio-informatique ou en spectrométrie.

Enfin, nous avons présenté l'algorithme ω^β-SOM qui étend l'algorithme w-kmeans proposé par [HNRL05] au cas des cartes topologiques. Cet algorithme a montré sa capacité a identifier correctement les dimensions pertinentes et sa grande stabilité. Nous avons également introduit une approche filtre de sélection de variable qui s'appuie sur la pondération obtenue ; le choix du test de Wilcoxon qu'elle utilise est peut-être à reconsidérer et nous envisageons de le remplacer par une test de Fisher. Un deuxième point mérite notre intérêt, tous les groupes mis en exergue s'appuient sur un même sous ensemble d'attributs mais rien ne nous assure que tous les regroupements pertinents s'appuient sur le même ensemble de descripteurs. Ainsi, il n'est pas exclu que, sans être totalement pertinentes, les partitions découvertes comportent des regroupements d'objets intéressants. Un moyen de pallier à ce problème est d'étendre nos travaux sur la sélection de variable au cas de la classification contextuelle [Can06] - *subspace clustering* - et d'utiliser des prondérations locales plutôt que globales [Bla06].

Dans le cadre de cette thèse, nous avons pu identifier différents problèmes liés à la réduction de dimension dans le cadre de l'apprentissage non supervisé. Dans ce contexte, la problématique de l'évaluation est un enjeu majeur, car contrairement aux problèmes de prédiction ou de regression il est difficile de définir ce qui est pertinent et ce qui ne l'est pas puisqu'on ne dispose pas de référence. Les différentes approches proposées pour la sélection de variables s'appuient sur la définition suivante de la pertinence d'un sous-ensemble de variables :

> *Dans le contexte de la classification automatique, un sous-ensemble de variables est pertinent dès lors qu'il participe à l'émergence d'une structuration en groupes d'un ensemble d'objets.*

Mais cette définition ne constitue qu'une amorce de la résolution du problème ; en outre, il nous reste à exhiber ce qui constitue une structuration en groupes ou une partition pertinente. La littérature est abondante en critères de qualité de partition mais il n'existe hélas pas de consensus autour d'un critère particulier ; ainsi, il est difficile de choisir un critère adéquat. Une étude approfondie des différents critères au travers la structure en treillis proposée par Marina Meilă nous semblerait pertinente ; cela nous conduirait soit à confirmer que la topologie induite est adaptée à l'évaluation de partition, soit au contraire à constater qu'il est nécessaire d'en établir une autre.

Ensuite, comme nous l'avons rappelé au chapitre 2, il peut exister plusieurs partitions intéressantes d'un même ensemble d'objets et dans ce cas il n'est pas possible de déterminer automatiquement celles qui sont intéressantes pour l'utilisateur sans avoir recours à des informations supplémentaires. Celle-ci peuvent être fournies au système d'apprentissage sous différentes formes et le premier mode d'acquisition qui vient à l'esprit consiste sans doute à interagir avec l'utilisateur ; on parle alors d'apprentissage actif. Lorsque des connaissances à priori sont disponibles et qu'on décide de les intègrer directement au processus d'apprentissage on parle d'apprentissage semi-supervisé [Bas05, Wag02].

Une approche proposée récemment par Jain et Law consiste à utiliser différents algorithmes de classification et à utiliser les partitions obtenues pour définir une nouvelle mesure de similarité qui traduit la propension des objets à se regrouper. Nous pensons que ce type d'approches est une alternative intéressante aux formes d'apprentissage que nous venons d'évoquer. En outre, elle pourrait permettre de traiter les données complexes en utilisant différentes représentation et en combinant les différentes partitions obtenues pour former un consensus.

Pour finir, l'un des enjeux actuels de l'apprentissage artificiel est la capacité à traiter de grandes masses de données. Une première approche consiste à définir des algorithmes incrémentaux qui permettent d'éviter un ré-apprentissage complet lorsque de nouvelles données sont disponibles ; il est alors nécessaire d'introduire les propriétés de stabilité et de plasticité des modèles construits. Ce travail a été réalisé par Farida Zehraoui [ZB04b, ZB04a] et il semble pertinent d'étendre notre pondération au modèle qu'elle a proposé.

Bibliographie

[ADMR05] L. Anolli, S. JR Duncan, M.S. Magnusson, and G. Riva, editors. *The hidden structure of interaction : from neurons to culture patterns*, volume 7 of *Emerging Communication : Studies on New Technologies and Practices in Communication*. IOS Press, Amsterdam, The Netherlands, April 2005.

[Amb96] Christophe Ambroise. *Approche probabiliste en classification automatique et contraintes de voisinage*. PhD thesis, UTC, Compiègne, 1996.

[Azz05] Hanene Azzag. *Classification hiérarchique par des fourmis artificielles : applications à la fouille de données et de textes pour le web*. PhD thesis, Université François Rabelais, Tours, December 2005.

[Bas05] Sugato Basu. *Semi-supervised Clustering : Probabilistics Models, Algorithms and Experiments*. PhD thesis, University of Texas, Austin - USA, August 2005.

[Bat94] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4) :537–550, 1994.

[BB95] Y. Bennani and F. Bossaert. A neural network based variable selector. In C. H. Dagli, M. Akay, C. L. Chen, B. R. Fernandez, and J. Ghosh, editors, *ANNIE'95*, 1995.

[BBD00] P. S. Bradley, K. P. Bennett, and A. Demiriz. Constrained k-means clustering. Technical Report MSR-TR-2000-65, Microsoft Research, May 2000.

[BBH+93] V. Bruce, AM. Burton, E. Hanna, P. Healey, O. Mason, A. Coombes, R. Fright, and A. Linney. Sex discrimination : how do we tell the difference between male and female faces ? *Perception*, 22(2) :131–152, 1993.

[BDL+04] Y. Bengio, O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16(10) :2197–2219, 2004.

[Ben01] Younès Bennani. *Systèmes d'apprentissage connexionnistes : sélection de variables*, volume 15(3-4) of *Revue d'Intelligence Artiticielle*. Hermes Science Publications, Paris, France, 2001.

[Ben06] Younès Bennani. *Apprentissage Connexionniste*. Editions Hermès Science, 2006.

[Ber91] Diane S. Berry. Child and adult sensitivity to gender information in patterns of facial motion. *Ecological Psychology*, 3(4) :349–366, 1991.

[Ber02] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[BG91] Léon Bottou and Patrick Gallinari. A framework for the cooperation of learning algorithms. In D. Touretzky and R. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 3, Denver, 1991. Morgan Kaufmann.

[BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *COLT '92 : Proceedings of the fifth annual workshop on Computational learning theory*, page 144–152, New York, NY, USA, 1992. ACM Press.

[Bla06] Alexandre Blansché. *Classification non supervisée avec pondération d'attributs par des méthodes évolutionnaires*. PhD thesis, Université Louis Pasteur - Strasbourg I, September 2006.

[BLP05] Fernando Bação, Victor Lobo, and Marco Painho. Geo-som and its integration with geographic information systems. In Marie Cottrell, editor, *WSOM*, pages 505–512, 2005.

[Bot91] Léon Bottou. *Une Approche théorique de l'Apprentissage Connexionniste : Applications à la Reconnaissance de la Parole*. PhD thesis, Université de Paris XI, Orsay, France, 1991.

[BY98] Vicki Bruce and Andrew Young. *In the Eye of the Beholder : The Science of Face Perception*. Oxford University Press, USA, December 1998.

[Can06] Laurent Candillier. *Contextualisation, visualisation et évaluation en apprentissage non supervisé*. PhD thesis, Université Charles de Gaulle - Lille 3, Lille, France, 2006.

[CB02] Dusan Cakmakov and Younes Bennani. *Feature Selection for Pattern Recognition*. Informa Press, Ed., 2002.

[CFGR94] T. Cibas, F. Fogelman, P. Gallinari, and S. Raudys. Variable selection with optimal cell damage. In *Proceeding of the ICANN'94*, volume 1, pages 727–730, 1994.

[CGG+95] M. Cottrell, B. Girard, Y. Girard, M. Mangeas, and C. Muller. Neural modeling for time sseries : A statistical stepwise method for weight elimination. *IEEE Transactions on Neural Networks*, 6(6), 1995.

[Cib96] Tautvydas Cibas. *Contrôle de la complexité dans les réseaux de neurones : régularisation et sélection de caractéristiques*. PhD thesis, University of Paris XI Orsay, Paris, France, December 1996.

[CIL03] Marie Cottrell, Smail Ibbou, and Patrick Letrémy. Traitement des données manquantes au moyen de l'algorithme de kohonen. In *Actes de la dixième conférence ACSEG, Nantes*, 2003.

[DB79] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, 1(2) :224–227, 1979.

[DHG01] Richard O. Duda, Peter E. Hart, and Stork David G. *Pattern Classification, Second Edition*. John Wiley and Sons, Inc., 2001.

[DLC03] Bi-Ru Dai, Cheng-Ru Lin, and Ming-Syan Chen. On the techniques for data clustering with numerical constraints. In Daniel Barbará and Chandrika Kamath, editors, *SDM*. SIAM, 2003.

[DNM98] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[DPJ+96] B. Dorizzi, G. Pellieux, F. Jacquet, T. Czernikov, and A. Munoz. Variable selection using generalized rbf networks : Application to forecast french t-bonds. 1996.

[Faw03] T. Fawcett. Roc graphs : Notes and practical considerations for data mining researchers. Technical Report HPL-2003-4, HP Labs, 2003.

[Fis36] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 :179–188, 1936.

[FLC02] J.-C. Fort, P. Letremy, and M. Cottrell. Advantages and drawbacks of the batch kohonen algorithm. In *10th European Symposium on Artificial Neural Networks, ESANN'2002*, Bruges, Belgium, April 2002.

[FM83] E. B. Fowlkes and C. L. Mallows. A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association*, 78(383) :553–569, September 1983.

[Fun01] Glenn Fung. A comprehensive overview of basic clustering algorithms, May 2001.

[GGNZar] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh. *Feature Extraction, Foundations and Applications, Editors*. Series Studies in Fuzziness and Soft Computing, Physica-Verlag. Springer, 2006, to appear.

[GTPF03] P. Giovanoli, C-H. J. Tzou, M. Ploner, and M. Frey. Three-dimensional video analysis of facial movements in healthy volunteers. *British Journal of Plastic Surgery*, 56(7) :644–652, October 2003.

[HA85] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1) :193–218, December 1985.

[HBV01] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3) :107–145, 2001.

[HJ01] H. Hill and A. Johnston. Categorizing sex and identity from the biological motion of faces. *Current Biology*, 11(11) :880–885, August 2001.

[HNCM05] Pierre Hansen, Eric Ngai, Bernard K. Cheung, and Nenad Mladenovic. Analysis of global k-means, an incremental heuristic for minimum sum-of-squares clustering. *Journal of Classification*, 22(2), September 2005.

[HNRL05] Joshua Zhexue Huang, Michael K. Ng, Hongqiang Rong, and Zichen Li. Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5) :657–668, 2005.

[HS93] B. Hassibi and D.G. Stork. Second order derivatives for networks pruning : Optimal brain surgeon. In *Advances in Neural Information Processing Systems 5*, pages 164–171. Morgan Kaufmann Publishers, 1993.

[JD88] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[JKV01] Bertrand Jouve, Pascale Kuntz, and François Velin. Extraction de structures macroscopiques dans des grands graphes par une approche spectrale. *Extraction des Connaissances et Apprentissage*, 1(4), 2001.

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering : a review. *ACM Computing Surveys*, 31(3) :264–323, 1999.

[Kay97] Daniel Kayser. *La représentation des connaissances*. Hermès, 1997.

[Koh01] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, New York, third extended edition edition, 1995,1997,2001.

[LCDS90] Y. Le Cun, J.S. Denker, and S.A Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems 2*, pages 598–605. Morgan Kaufmann Publishers, 1990.

[LG] P. Leray and P. Gallinari. De l'utilisation d'obd pour la sélection de variables dans les perceptrons multicouches. *Systèmes d'apprentissage connexionnistes : sélection de variables, Numéro spécial de la Revue d'Intelligence Artificielle*, 15(3-4) :373.

[Li06] Tao Li. A Unified View on Clustering Binary Data. *Machine Learning*, 62(3) :199–215, March 2006.

[LLB04] Fernando Lourenço, Victor Lobo, and Fernando Bação. Binary-based similarity measures for categorical data and their application in self-organizing maps, April 2004.

[LM98] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, 1998.

[LVV03] A. Likas, N. Vlassis, and J. Verbeek. The Global k-means Clustering Algorithm. *Pattern Recognition*, 36(2) :451–461, 2003.

[Mac94] D.J.C. MacKay. *Bayesian methods for backpropagation networks*, chapter 6. Springer-Verlag, New York, USA, 1994.

[Mag00] MS. Magnusson. Discovering hidden time patterns in behaviour : T-patterns and their detection. *Behavior research methods, instruments and computers : a journal of the Psychonomic Society, Inc.*, 32(1) :93–110, Febrary 2000.

[MB88] Geoffrey J. McLahlan and Kaye E. Bashord. *Mixture Models : Inference and Applications to Clustering*. Marcel Dekker, Inc., New York, 1988.

[Mei03] Marina Meilă. Comparing clusterings by the variation of information. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *COLT*, volume 2777 of *Lecture Notes in Computer Science*, pages 173–187. Springer, 2003.

[Mei05] Marina Meilă. Comparing clusterings : an axiomatic view. In Luc De Raedt and Stefan Wrobel, editors, *ICML*, pages 577–584. ACM, 2005.

[Mei06] Marina Meilă. Comparing clusterings - an information based distance. in print, 2006.

[ML01] Vladimir Makarenkov and Pierre Legendre. Optimal Variable Weighting for Ultrametric and Additive Trees and K-means Partitioning : Methods and Software. *Journal of Classification*, 18(2) :245–271, February 2001.

[MMP02] P. Mitra, C.A. Murthy, and S.K. Pal. Unsupervised Feature Selection Using Feature Similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4), 2002.

[Moo94] J. Moody. Prediction risk and architecture selection for neural networks. In V. Cherkassky, J.H. Friedmann, and H. Wechsler, editors, *From Statistics to Neural Networks - Theory and Pattern Recognition Application*, 1994.

[Mor84] André Morineau. Note sur la caractérisation statistique d'une classe et les valeurs-tests. Bulletin technique 2, Centre international de statistique et d'informatique appliquées, Saint-Mandé, France, 1984.

[MU05] F. Moutarde and A. Ultsch. U*F clustering : a new performant "cluster-mining" method based on segmentation of Self-Organizing Maps. In *Proceedings of the 5th Workshop On Self-Organizing Maps (WSOM'05)*, pages 25–32, Paris 1 Panthéon-Sorbonne University, France, September 2005.

[Mur95] F. Murtagh. Interpreting the Kohonen self-organizing feature map using contiguity-constrained clustering. *Pattern Recognition Letters*, 16(4) :399–408, April 1995.

[Nea94] R.M. Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, Canada, 1994.

[OM04] D. Opolon and F. Moutarde. Fast semi-automatic segmentation algorithm for Self-Organizing Maps. In *Proceedings of ESANN'2004 , European Symposium on Artificial Neural Networks, Bruges (Belgium)*, pages 507–512, 2004.

[PÖ4] Georg Pölzlbauer. Survey and comparison of quality measures for self-organizing maps. In Ján Paralič, Georg Pölzlbauer, and Andreas Rauber, editors, *Proceedings of the Fifth Workshop on Data Analysis (WDA'04)*, pages 67–82, Sliezsky dom, Vysoké Tatry, Slovakia, June 24–27 2004. Elfa Academic Press.

[PHL96] M.W. Pedersen, L.K. Hansen, and J. Larsen. Pruning with generalization based weight saliencies : γobd, γobs. In *Advances in Neural Information Processing Systems 8*. Morgan Kaufmann Publishers, 1996.

[Ros96] F. Rossi. Attribute suppression with multi-layer perceptron. In *Proceedings of IEEEI-MACS'96, Lille, France.*, 1996.

[Rou85] Maurice Roux. *Algorithmes de classification*. Masson, Paris, 1985.

[RRK90] D. W. Ruck, S. K. Rogers, and M. Kabrisky. Feature selection using a multilayer perceptron. *International Journal on Neural Network Computing*, 2(2) :40–48, 1990.

[RS00] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Local Linear Embedding. *Science*, 290 :2323–2326, December 2000.

[RZ99] A-P. N. Refenes and A.D. Zapranis. Neural model identification, variable selection and model adequacy. *Journal of Forecasting*, 18(5) :299–332, Sep 1999.

[Str04] Marc Strickert. *Self-Organizing Neural Networks for Sequence Processing*. PhD thesis, University of Osnabrück, Germany, June 2004.

[TdSL00] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290 :2319–2323, December 2000.

[TGPF05] C.H.J. Tzou, P. Giovanoli, M. Ploner, and M. Frey. Are there ethnic differences of facial movements between europeans ans asians ? *British Journal of Plastic Surgery*, 58(2) :183–195, March 2005.

[TK02] IM. Thornton and Z. Kourtzi. A matching advantage for dynamic human faces. *Perception*, 31(1) :113–132, 2002.

[TNZ96] V. Tresp, R. Neuneier, and H. G. Zimmermann. Early brain damage. In M. Mozer, M. Jordan, and Th. Petsche, editors, *Advances in Neural Information Processing Systems (NIPS 1996)*, pages 669–675. MIT Press, 1996.

[Ult05] A. Ultsch. Clustering with SOM : U*C. In *Proceedings of the 5th Workshop On Self-Organizing Maps (WSOM'05)*, pages 75–82, Paris 1 Panthéon-Sorbonne University, France, September 2005.

[US90] A. Ultsch and H.P. Siemon. Kohonen's self organizing feature maps for exploratory data analysis. In *Proceedings of the International Neural Networks Conferences (INNC'90)*, pages 305–308. Kluwer Academic Press, Paris, 1990.

[VA99] Juha Vesanto and Jussi Ahola. Hunting for Correlations in Data Using the Self-Organizing Map. In H. Bothe, E. Oja, E. Massad, and C. Haefke, editors, *Proceeding of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA '99)*, pages 279–285. ICSC Academic Press, 1999.

[VA00] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3) :586–600, 2000.

[VSH03] Juha Vesanto, Mika Sulkava, and Jaakko Hollmén. On the decomposition of the self-organizing map distortion measure. In *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03)*, pages 11–16, Kitakyushu, Japan, September 2003.

[Wag02] Kiri Lou Wagstaff. *Intelligent Clustering With Instance-Level Constraints*. PhD thesis, Cornell University, August 2002.

[Wal83] David L. Wallace. A Method for Comparing Two Hierarchical Clusterings : Comment. *Journal of the American Statistical Association*, 78(383) :569–576, September 1983.

[WW98] Thomas-H. Wonnacott and Ronald-J. Wonnacott. *Statistique, Economie - Gestion - Sciences - Médecine*. Economica, Paris, 1998.

[XW05] Rui Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3) :645–678, 2005.

[YB97] M. Yacoub and Y. Bennani. Hvs : A heuristics for variables selection in multilayer neural network classifiers. In C. H. Dagli, M. Akay, C. L. Chen, B. R. Fernandez, and J. Ghosh, editors, *ANNIE'97*, volume 7, pages 527–532, St. Louis, Missouri, USA, 1997. ASME Press.

[You04] Genane Youness. *Contributions à une méthodologie de comparaison de partitions*. PhD thesis, Université Paris 6, July 2004.

[Zan05] Jean-Marc Zanimetti. *Statisque spatiale, méthodes et applications géomatiques*. Hermès Sciences Publications, Lavoisier, Paris, 2005.

[ZB04a] Farida Zehraoui and Younès Bennani. M-SOM-ART : Growing Self Organizing Map for Sequences Clustering and Classification. In Ramon López de Mántaras and Lorenza Saitta, editors, *ECAI*, pages 564–570. IOS Press, 2004.

[ZB04b] Farida Zehraoui and Younès Bennani. M-SOM : Matricial Self Organizing Map for sequences clustering and classification. In *Proceeding of the International Joint Conference on Neural Network, IJCNN'04*, Budapest, Hungary, July 2004.

PARTIE V

Annexes

μ-SOM : Weighting features during clustering

Sébastien Guérif, Younès Bennani
LIPN - CNRS - University of Paris 13
Villetaneuse. France
{sebastien.guerif, younes.bennani}@lipn.univ-paris13.fr

Éric Janvier
Numsight Consulting France
Boulogne Billancourt. France
e.janvier@numsight.com

Abstract - *Real life datasets used in marketing studies contain a lot of redundant features which may prevent data-mining techniques such as self-organizing maps from discovering relevant clusters. An extension of the batch Kohonen's algorithm is proposed in this paper to avoid the large amount of work which is required by data preprocessing if redundancy isn't treated explicitly by the training method. The proposed approach integrates a weighting of variables built on a simultaneous clustering of both observations and variables and avoids the side effects of redundancy. An application to market segmentation is then briefly described to validate the learning algorithm introduced; identified clusters of products and motivations are used to simplify the analysis of the consumer segmentation by giving the user a first rough description of the different groups.*

Key words - Data-mining, Market segmentation, Redundant features, Self-Organizing Map, Weighting

1 Introduction

In real life application, data-mining techniques are applied to datasets which contain numerous redundant features. On the one hand, strong correlations between variables may be useful to deal with missing values [2] or to detect outliers. On the other hand, clustering algorithms built on Euclidean distance may be prevented from discovering correct clusters if data are not preprocessed. Intuitively, redundancy gives more importance to some information which are represented by many features and may occult others that are less present. In the worst case, some irrelevant informations would be expressed by many dimensions and some relevant knowledge by very few variables; this extreme situation may lead to a less interesting clustering of the data. To address this problem, different ways are proposed, in which three categories can be distinguished: selection of variables, extraction of features or weighting of features [1].
Some methods for unsupervised selection of variables using similarity of features have been proposed in [7, 8]. It is well known that Euclidean distance can be approximated when few dimensions compared with the data dimension are missing, but then eliminating some fea-

tures makes it harder to treat correctly missing values. Principal component analysis (PCA) [6, 9] and factor analysis [13] address efficiently this problem by reducing the attribute space from a large number of variables to a smaller number of orthogonal factors which preserve the maximum of variance. However, they require an important effort from the user to interpret and understand the new representation of one's data. Moreover, these techniques are built on the correlation matrix computation which requires the whole data to be known, and the computation of its eigenvalues and associated eigenvector which may suffer from numerical instabilities. The Mahalanobis distance has been introduced to take care of correlations between dimensions but suffers from the same numerical instabilities as PCA or factor analysis, because it requires the computation of the correlation matrix inverse.
The proposed approach is built on a simultaneous clustering of both observations and variables using self-organizing maps [4] which are well known for their ability to make good representation of data in large dimension. A weighting mechanism which decrease the weight of redundant features has been integrated to the learning algorithm.
The remainder of this paper is organized as follows. Section 2 presents the new algorithm designed to reduce redundancy side effects during the construction of self-organizing maps. Section 3 discusses obtained results and application of our approach to market segmentation while section 4 concludes the paper.

2 μ-SOM: weighting features during clustering

2.1 Outlines and algorithm of μ-SOM

Two self-organizing maps are constructed simultaneously, the first one represents observations and the second one the features' profile. [11] suggests to first realize a clustering of observations and then a clustering of component planes to detect correlations between variables, it is the starting point of our approach. The first basic idea used here is that components planes are a good representation of features, robust to outliers. The second basic idea is that the total weight of variables could be shared between the different dimensions according to the distribution of their best matching units over the map.
The high-level algorithm 1 gives outlines of the μ-SOM learning. The map of observations $SOM^{(data)}$ is made up of $m^{(data)}$ units noted $U^{(data)} = \{1, \ldots, m^{(data)}\}$. Analogously, the map of features $SOM^{(attr)}$ comprises $m^{(attr)}$ units noted $U^{(attr)} = \{1, \ldots, m^{(attr)}\}$. The unit $i \in U^{(data)}$ (resp. $j \in U^{(attr)}$) has $\omega_i(t) \in \mathbb{R}^n$ (resp. $\omega_i(t) \in \mathbb{R}^{m^{(data)}}$) as profile at iteration t.
Some details of the μ-SOM learning algorithm have to be defined:

- The distance used to find the best matching unit of an observation at the t^{th} iteration is the following weighted Euclidean distance $d^{(data)}(x, y) = \sqrt{\sum_{i=1}^{n} \mu_i(t)(x_i - y_i)^2}$, where $\mu_i(t) \in \mathbb{R}_+$ are such that $\sum_{i=1}^{n} \mu_i(t) = 1$.
- The profile of features $i \in \{1, \ldots, n\}$ at iteration t is given by the corresponding component plane, that is $fp_i(t) = \left(\omega_{ji}^{(data)}(t)\right)_{j \in U^{(data)}}$, which are normalized to unit range.
- $\alpha : \{0, \ldots, T_{Max}\} \rightarrow [0, 1]$, where T_{Max} is the number of iterations, increases from 0 to 1 and is used to avoid oscillations of weights during the learning process. A linear function such $\alpha(t) = \frac{t}{(T_{Max}-1)}$ is appropriated.

Algorithm 1 μ-SOM learning

Initialize $\mu_i(0) = \frac{1}{n}$, for $i = 1, \ldots, n$
Initialize $\omega_i^{(data)}(0) \in \mathbb{R}^n$, for $i \in U^{(data)} = \{1, \ldots, m^{(data)}\}$
Rough training of $SOM^{(data)}$
Extract profile of attributes $fp_i(t)$ from $SOM^{(data)}$
Initialize $\omega_i^{(attr)}(0) \in \mathbb{R}^{m^{(data)}}$, for $i \in U^{(attr)} = \{1, \ldots, m^{(attr)}\}$
Rough training of $SOM^{(attr)}$
Compute new weights $\mu_i^{new}(0)$
Update weights $\mu_i(1) \leftarrow \alpha(0).\mu_i(0) + (1 - \alpha(0)).\mu_i^{new}(0)$
Initialize $t \leftarrow 1$
while $(t < T_{max})$ **do**
  Fine training epoch on $SOM^{(data)}$
  Extract profile of attributes from $SOM^{(data)}$
  Fine training epoch on $SOM^{(attr)}$
  Compute new weights $\mu_i^{new}(t)$
  Update weights $\mu_i(t) \leftarrow \alpha(t).\mu_i(t) + (1 - \alpha(t)).\mu_i^{new}(t)$
  $t \leftarrow t + 1$
end while

The map of observations is first roughly trained to organize neurons according to topological ordering. Then profiles of features are extracted and used to roughtly train the map of variables. Finally, fine tuning epoches of both maps are alternated and weighting is computed after each update of the map of features.

2.2 Details of the weighting mechanism

The basic idea of the integrated weighting mechanism is to share total weight between a set of features $F = \{1, \ldots, n\}$ according to their similarity. It proceeds as follows :

1. Each unit $i \in U^{(attr)}$ receives a potential weight to share between the different features that is computed using Geary local spatial auto-correlation index [3, 5]:

$$G_i(t) = \frac{\frac{1}{2.L_i(t)} \sum_{j \in U^{(attr)}} c_{ij}(t) . \|\omega_i(t) - \omega_j(t)\|^2}{\frac{1}{m^{(attr)}-1} \sum_{j \in U^{(attr)}} \|\omega_i(t) - \omega_j(t)\|^2}$$

   where $L_i(t) = \sum_{j \in U^{(attr)}} c_{ij}(t)$. $c_{ij}(t) \in \{1, 0\}$ indicates whether units i and j are neighbors or not. Typically, $c_{ij}(t) = \left(d^{(attr)}(i,j) < 1\right)$, where $d^{(attr)}(i,j)$ is the distance between units $i \in U^{(attr)}$ and $j \in U^{(attr)}$ on the map of features.

2. Then, each variables $i \in F$ asks each units $j \in U^{(attr)}$ in the neighborhood of its best matching units $\tilde{i}$ for a part of its potential weight : $part_i^{(j)}(t) = \exp\left(-\frac{1}{2}\left(\frac{d^{(attr)}(\tilde{i},j)}{\sigma(t)}\right)^2\right)$

3. Finally, the potential weight of each units is shared between features according to the requested part: $\mu_i^{new}(t) = \frac{1}{\sum_{j \in U^{(attr)}} G_j(t)} \sum_{j \in U^{(attr)}} G_j . \left(\frac{part_i^{(j)}(t)}{\sum_{k \in F} part_k^{(j)}(t)}\right)$

The Geary local spatial auto-correlation index has been chosen for its ability to measure the similarity of a unit and its neighbors compared to the global variance of unit's prototype. Indeed, areas of the map which represent highly similar features have a lower potential weight than areas with high distortion. It has been noticed that units on the border of the map are slightly penalized because they have less neighbors than the other, leading to a lower local variance is for units in the middle of the map.
It must be pointed out that the set of features F can be replaced by any of its subsets; actually the proposed approach is ready to deal with missing values.

2.3 Cluster analysis

When using self-organizing maps, more or less as many clusters as units on the map are obtained so it is impracticable to analyze each one separately. A clustering of unit prototypes permits to reduce the number of clusters. Hierarchical Ascending Classification (HAC) or k-means are often used to perform this task. We have chosen to apply the method proposed in [12] to cluster our maps. Several k-means clustering are computed for varying number of centers and then the Davies-Bouldin index is used to choose the best one.
Thus, a first rough description of identified clusters of observations can be made using features groups. In the same way, class of observations should be used to roughly describe clusters of attributes. we proposed to proceed as follow:

1. For each cluster i of observations, compute the mean $\overline{x_{ij}}$ of each dimension j.
2. Then, normalize to unit range each mean per dimension $pos_{ij} = \frac{\overline{x_{ij}} - \min_i\{\overline{x_{ij}}\}}{\max_i\{\overline{x_{ij}}\} - \min_i\{\overline{x_{ij}}\}}$.
3. For each cluster i, compute the mean $\overline{pos_i} = mean_{j \in F}(pos_{ij})$ and standard deviation $\sigma_{pos_i} = std_{j \in F}(pos_{ij})$ of the normalized means pos_{ij}.
4. For each cluster i, select all dimensions j such $pos_i j \geq \overline{pos_i} + \sigma_{pos_i}$
5. Representation ratios of each classes of features is a useful rough description of the cluster i.

Rough descriptions given by representation ratios are useful to give the user a first idea of relationships between observations and features clusters and facilitate a cross analysis of revealed groups.

3 Application and results

3.1 Results

Our approach has been evaluated using various dataset and obtained results on the *waveform* and the *isolet* datasets from the UCI Machine Learning Repository [10] are presented here. Cross validation has been used to compare the quality of maps obtained using μSOM to those built with the batch version of Kohonen's algorithm. Each dataset has been divides in five parts; four subsets has been used by the training algorithm and the last one to evaluate the quality of the map. Three indexes has been used to evaluate the quality of topological maps:

- mean quantification error (Qerr)

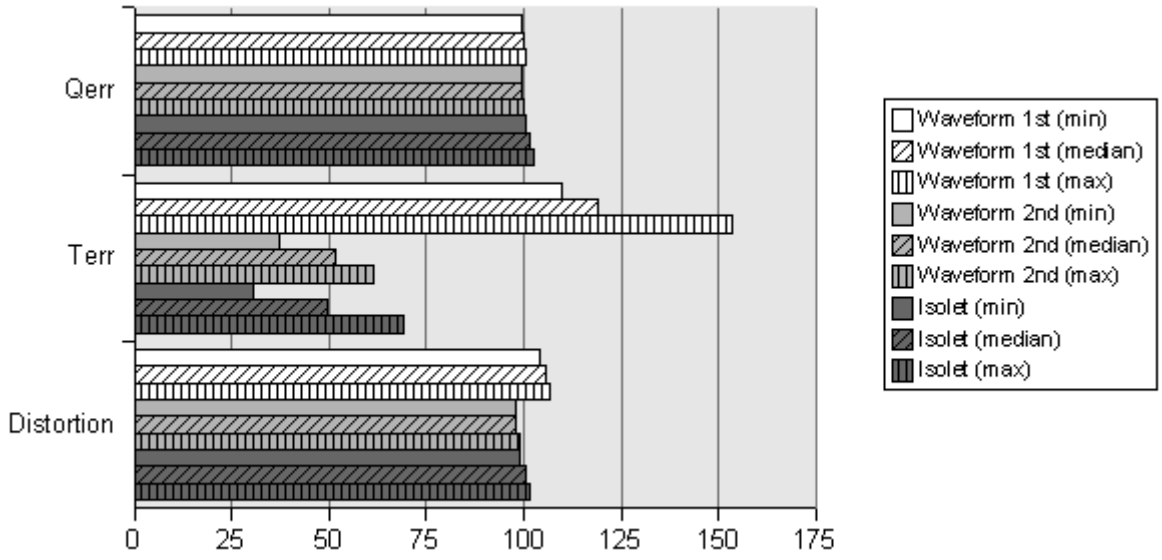

Figure 1: Relative quality of μSOM (index 100 for SOM)

- topological error rate (Terr)
- distortion measures

In our first experiments on the *waveform* dataset (waveform 1st), the number of neurons on the map of features was greater than the number of variables. The resulting map was unusable to identify correct correlations between features. Then, the number of units has been decreased (waveform 2nd). Observed differences on quantification error and distortion measure between topological maps obtained using μSOM and the standard algorithm are not significant. Nevertheless, it should be noticed that the topological error rate has been greatly improved on both dataset.

3.2 Application to marketing

The aim of market studies is to understand the behavior of consumers and identify groups which share the same interests. Data are generally collected by a sample survey of consumers and contains typically several hundred of observations described by several tens of variables. A segmentation of both observations and variables allows us to identify group of consumers, categories of products and relationship between them.
Our dataset contains some 230 answers from 1006 consumers. The application of μ-SOM algorithm and the clustering of the obtained map have permitted to identify 17 categories of products and 14 groups of consumers. The segmentation of products has been analyzed first and then rough descriptions of groups of consumers have been computed. They are very helpful in practice because they give a first idea of what a cluster contains and gives a pertinent axis of analysis.

The figure 2 presents both the distributions of consumers over the map and the different identified groups. Then the whole classes of features are presented figure 5 and a zoom on

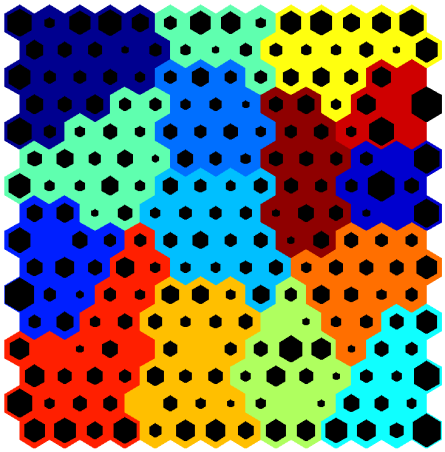

Figure 2: Distribution and classes of observations over the map

two diferent areas is proposed figure 3 and 4. Finally, figure 6 shows the distribution of features' weights.

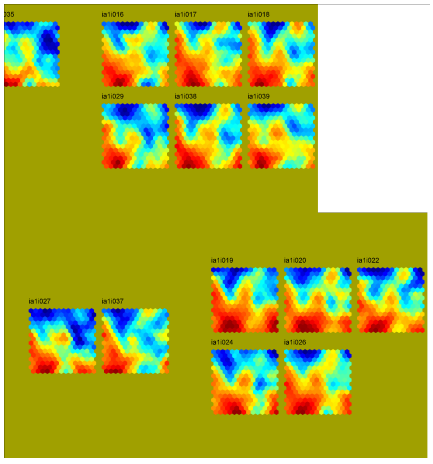

Figure 3: Upper right corner of the map of features.

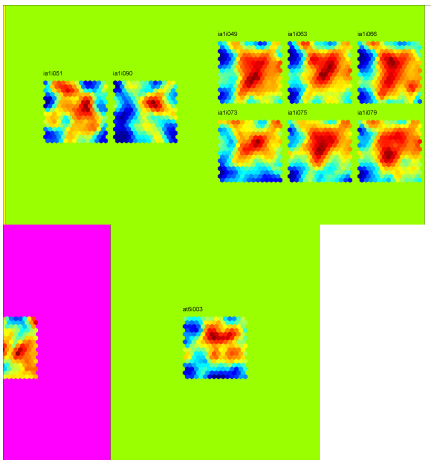

Figure 4: Middle right area of the map of features.

4 Conclusions and further research

A novel learning algorithm for Self-Organizing Map is presented in this paper. It leads to better quality maps than the batch version of the Kohonen's batch algorithm. Actually, it has been successfully applied on market studies datasets and appears to be useful for both avoiding a large amount of work needed to preprocess data and providing rough descriptions of clusters which could be used as starting point for the analysis. Experiments are under way to evaluate the ability of the proposed algorithm to deal with missing values and noisy data. Future work includes adaptation of this method to the on-line version of Kohonen's algorithm and improvement of the quality of the distance used with features profile.

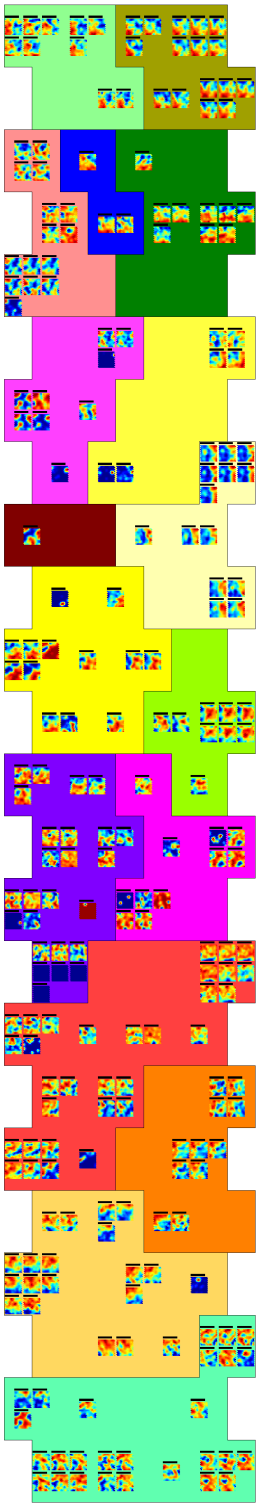

Figure 5: Distribution of features and categories. Component planes of th map of observations are represented at the position of their best matching units. This visualization is useful to analysis features correlations.

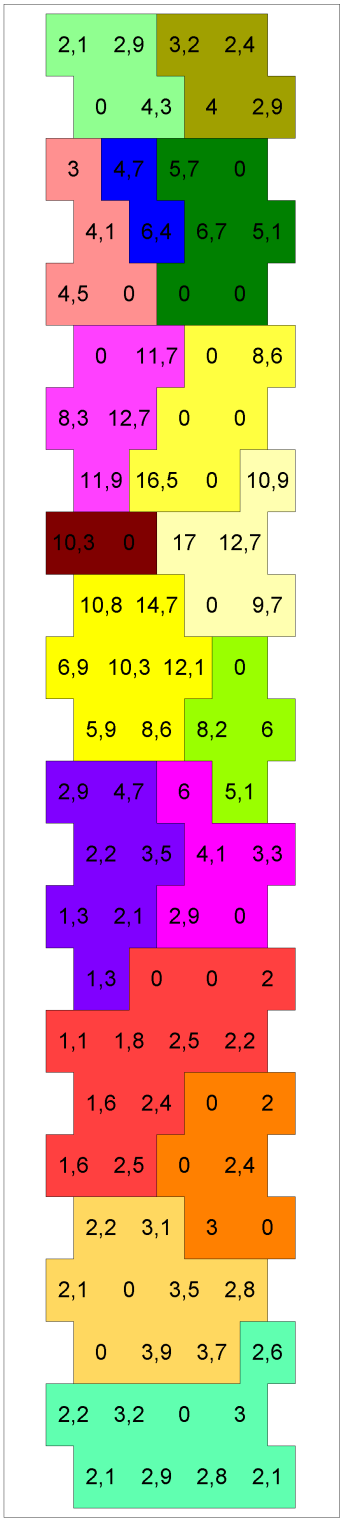

Figure 6: Distribution of weight ($\times 10^{-3}$) of features. Each features is given a weight according to it best matching unit.

Acknowledgement

We would like to thank Mark Kerslake from Numsight Consulting France for our discussion about the relevance of revealed classes of both products and consumers, his review and english correction.

References

[1] Y. Bennani (1999), Adaptive weighting of pattern features during learning, *International Joint Conference on Neural Networks, IJCNN'99*, **vol. 5**, p. 3008-3013.

[2] M. Cottrell, S. Ibbou et P. Letrémy (2003), Traitement des données manquantes au moyen de l'algorithme de Kohonen, *Actes de la dixième conférence ACSEG, Nantes*.

[3] R. C. Geary (1954), The contiguity ratio and statistical mapping, *The Incorporated Statistician*, p. 115-145.

[4] T. Kohonen (2001), *Self-Organizing Maps 3rd edition*, Heidelberg, Springer.

[5] L. Lebart (1969), Analyse statistique de la contiguité, *Publications de l'ISUP*, p. 81-112.

[6] L. Lebart, A. Morineau et M. Piron (2000), *Statistique exploratoire multidimensionnelle 3e édition*, Dunod.

[7] P. Mitra, C.A. Murthy and Sankar K. Pal (2002), Unsupervised Feature Selection Using Feature Similarity, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **vol. 24-3**, p. 301-312.

[8] Sankar K. Pal, Rajat K. De and J. Basak (2000), Unsupervised Feature Evaluation: A Neuro-Fuzzy Approach, *IEEE Transactions on Neural Networks*, **vol. 11-2**, p. 366-376.

[9] G. Saporta (1990), *Probabilités, analyse de données et statistiques*, Paris, Editions Technip.

[10] UCI Machine Learning Repository, *http://www.ics.uci.edu/ mlearn/MLRepository.html*.

[11] J. Vesanto and J. Ahola (1999), Hunting for Correlations in Data Using the Self-Organizing Map, *In Proceeding of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99)*, ICSC Academic Press, p. 279-285.

[12] J. Vesanto and E. Alhoniemi (2000), Clustering of the Self-Organizing Map, *In IEEE Transactions on Neural Networks*, **vol. 11-3** p. 586-600.

[13] N. Wu and J. Zhang (2005), Factor-analysis based anomaly detection and clustering, *Decision Support Systems*, **to appear**.

Connectionist and Ethological Approaches for Discovering Salient Facial Movements Features in Human Gender Recognition

Sébastien Guérif, Younès Bennani
University of Paris 13, CNRS UMR 7030 - LIPN, F-93430 Villetaneuse
{*sebastien.guerif, younes.bennani*}*@lipn.univ-paris13.fr*

Claude Baudoin
University of Paris 13, CNRS UMR 7153 - LEEC, F-93430 Villetaneuse
claude.baudoin@leec.univ-paris13.fr

Abstract. *Individual Facial movements signal various social information to other persons, like the gender of the sender. We used an ethological and a connectionist approaches in order to detect these movements and their characteristics in men and in women. Behavioural results indicate both qualitative and quantitative differences between men and women. The connectionist approach involves similar and complementary conclusions. The ethological study has been focused on the main movement differences as well as did the connectionist one but this last approach showed important differences between men and women in motionless events. These pilot results leads to a re-examination of behavioural events and a checking of lateralization of movements correlated with the gender.*

Keywords. *Facial movements, gender recognition, unsupervised learning, clustering, self-organizing maps*

1. Introduction

Social life in human groups involves constant regulatory processes like social categorization of interacting partners. One type of social category among the most obvious is the gender. Various body parts are used in signaling gender and the face is an important area as it has been demonstrated in previous studies [2, 3, 4]. Several authors showed the role of facial movements in gender categorization [9, 13]. Curiously only few studies concern the production of facial movements and their temporal organization [1, 8, 14]. Moreover the use of complex experimental systems for facial recording induces unnatural situations. Our study concerns an experimental situation with young adult subjects confronted to a cognitive task without direct interacting partners but with a female experimenter welcoming them before testing and video recording their behaviour from a contiguous room. This situation was not social but the context was social. Our aims were (i) to constitute a database allowing further comparisons between men and women, (ii) to code facial movements using an objective method, (iii) to detect and to characterize the temporal organization of the facial movements, (iv) to use the same data base for studying salient facial features with a connectionist approach, (v) to compare emerging results from ethological and connectionist approaches.

The rest of this paper is organized as follows. After, a brief presentation of the protocol used to collect the data, we present both ethological and connectionist approaches. Then, experiments and results are presented and discussed. Finally, we conclude and we give some point that will we developed in further research.

2. Collection of data base

Our purpose was to obtain a video recording from women and men in a standardized situation that permits expression of various facial movements: labial movement related to verbal answer, emotional reaction, etc. The experimental situation was a cognitive task realized in an indirect social context (reception followed by task instructions, filmed by a video camera operated by a female experimenter in the next room).

A total of 20 students (11 women and 9 men) from the University of Paris XIII volunteered to participate in the study. All were naive to the true

purpose of the study and were not paid for their participation.

Subjects were received by an experimenter and then left alone in a room where they followed instructions given by a laptop screen. The task consisted in looking at pictures and saying whether it was ambiguous/normal or not. No time limit had been imposed and the experiments lasted between 1.5 and 4.5 minutes (mean duration was 2.75 minutes). Some subjects were set aside because of particular situations (important movements of the body or the head, wearing glasses or a beard, etc.) Only 5 subjects of each gender were selected for the remaining part of the study.

With a view to standardizing the database we chose 3 sequences of 3 seconds centred on an easy to locate verbal answer from the subjects. Thus 3 sequences with a similar context are available per subject. We defined 36 face points involved in the facial movements that were easy to identify [8,14]. Figure 1 indicates the positions of the face points considered.

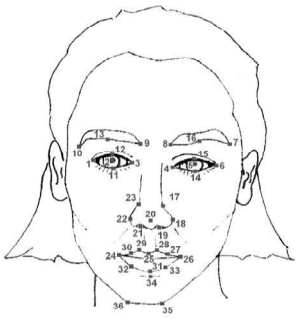

Figure 1: Position of the face points

The sequences were sampled at 13 images per second, and an operator recorded the 36 face points coordinates. This selection was repeated at least twice and the mean position was retained to reduce errors due to tiredness of the operator. The coordinates of the points were relative to the subjects face. Actually, the x axis is the line between points 3 and 4, and the y axis the orthogonal line crossing through point 20.

3. Ethological approach

For a given facial point (fig. 1), a salient movement was defined as the distance from origin which was higher than the mean distance calculated during a 3 seconds period (39 images) majored by the standard error. The number of movements was studied comparatively between men and women. Then we detected how distance variations from origin of the 36 facial points occurred during the 3 seconds periods for men and for women using the Magnussons THEME 5.0 software (http://www.noldus.com). This software allowed to detect T-patterns of facial movements, which are defined as repetitive real time organized behavioural structures [11]. Only some results are presented below.

We observed a higher mean number of movements produced per 3 seconds period by men comparatively to women (n = 86 vs. 69, $p < .05$, exact permutation test) as well as a tendency to present a higher number of T-patterns in men (on a basis of 100 movements, men produced 66 T-patterns vs. 46 in women, $p = .055$, exact permutation test). T-patterns involved on average 4 different facial points in men and 3 in women ($p = .079$). Moreover we discovered qualitative differences in the T-pattern composition linked to gender: men produced simple patterns involving temporal left eyebrow and left nostril, and women produced simple patterns involving internal and median parts of the right eyebrow.

Our pilot results indicated that man and woman facial movements were quantitatively and, at least for some of them, qualitatively different during a cognitive task performed in a social context.

4. Connectionist approach

Two different approaches are available to exploit our dataset: classification and clustering. The first one falls into supervised learning and builds a classifier. The latter approach detects groups of similar observations, called clusters. Our purpose is to determine whether the intrinsic structure of the data space is related to the gender of subjects or not. So, our interest has been focused on unsupervised learning approaches and Self-Organizing Maps (SOM) [10] were chosen to carry out our analysis. On one hand, SOM provides a convenient way to visualize the structure of our data [15]. On the other hand, the different clusters can be labeled according the gender of grouped observations and then be used as a classifier whose evaluation may give us some interesting information. First SOM are briefly introduced, then our methodology is explained and finally, the experimental results obtained using the Matlab somtool-

box [16] are given and discussed.

4.1. Connectionist model : Self-Organizing Maps

SOM was introduced by Pr. Teuvo Kohonen in the early 80's as a convenient clustering and visualization tool. High-dimensional data are projected on a low dimension discrete space, called the topological map, preserving the local topology of the initial space; thus, the observations which are close to each other are projected on a localized area. A map should be viewed as a set of neurons (or units), organized according to a grid that defines their neighbourhood relationships. Each neuron is associated to one point of the observations' space: its profile.

Self-Organizing Maps (SOM) implement a particular form of competitive artificial neural networks; when an observation is recognized, activation of an output cell competition layer leads to inhibit activation of other neurons and reinforce itself. It is said that it follows the so called Winner Takes All rule. Actually, neurons are specialized in the recognition of one kind of observations. The learning is unsupervised because neither the classes nor their number is fixed a priori.

A SOM consists in a two dimensional layer of neurons which are connected to n inputs according n exciting connections of respective weights w and to their neighbors with inhibiting links.

The training set is used to organize these maps under topological constraints of the input space. Thus, a mapping between the input space and the network space is constructed; closed observations in the input space would activate two closed units of the SOM.

An optimal spatial organization is determined by the SOM from the received information, and when the dimension of the input space is lower than three, both position of weights vectors and direct neighbourhood relations between cells can be represented visually.

4.2. Learning algorithm

Connectionist learning is often presented as a minimization of a risk function (cost function). In our case, it will be carried out by the minimization of the distance between the input samples and the map prototypes (referents), weighted by a neighbourhood function h_{ij} . To do that, we use a gradient algorithm for optimization. The criterion to be minimized is defined by:

$$R_{SOM} = \frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{M} h_{jNN(x^{(k)})} \left\| \omega_j - x^{(k)} \right\|^2 \qquad (1)$$

N represents the number of learning samples, M the number of neurons in the map, $NN(x^{(k)}$ is the neuron having the closest referent to the input form $x^{(k)}$, and h the neighbourhood function.

The weights of all the neurons are updated until stabilization according to the following adaptation rules: If $\omega_j \in V_{NN(x^{(k)})}$ then adjust the weights using:

$$\omega_j(t+1) = \omega_j(t) - \varepsilon(t) h_{jNN(x^{(k)})} \left(\omega_j - x^{(k)} \right) \qquad (2)$$

4.3. Labelling the map

Training of the self-organizing map is totally unsupervised; and actually, it does not make use of the data labels (namely female or male). Therefore, at the end of the training phase we only had a topological map based on the transformed coordinate data without any additional information. Nevertheless, it should be emphasized that the map defined a partition of the dataset which can be used to assign each neuron a label. Actually, each neuron is labelled using the most represented gender associated with that neuron. As such, the labelling is very sensitive to small changes in gender distribution. Therefore, to increase robustness of the labelling, a chi-square test was used to check whether the distribution of that part is significantly different from that of the whole dataset. Therewith, it should be emphasized that some neurons remained unlabeled.

4.4. SOM segmentation

We segment the SOM using the K-means algorithm. It is another clustering method. It consists in choosing arbitrarily a partition. Then, the samples are treated one by one. If one of them becomes closer to the center of another class, it is moved into this new class. We calculate the centers of new classes and we reallocate the samples to the partitions. We repeat this procedure until having a stable partition.

The criterion to be minimized in this case, is defined by:

$$R_{K-means} = \frac{1}{C}\sum_{k=1}^{C}\sum_{x \in Q_k} \|x - c_k\|^2 \qquad (3)$$

where C represents the number of clusters, Q_k is the cluster k, c_k is the center of the cluster Q_k or the referent.

The basic algorithm requires fixing K, the number of clusters wished. However, there is an algorithm to calculate the best value for K assuring an optimal clustering. It is based principally on the minimization of Davies-Bouldin index, defined as follows :

$$I_{DB} = \frac{1}{C}\sum_{k=1}^{C} max_{k\neq l}\left\{\frac{S_c(Q_k)+S_c(Q_l)}{d_{ce}(Q_k,Q_l)}\right\} \quad (4)$$

where $S_c(Q_k) = \frac{\sum_i \|x_i - c_k\|}{|Q_k|}$ is the intracluster dispersion of cluster Q_k and $d_{ce}(Q_k,Q_l) = \|c_k - c_l\|$ is the distance (centroid linkage) between the center of clusters k and l. This clustering procedure aims to find internally compact spherical clusters which are widely separated.

There are several methods to segment the SOMs [17]. Usually, they are based on the visual observations and the manual assignment of the map cells to the clusters. Several methods use the K-means algorithm with given ranges for K value. Our work is based on the approach of Davies-Bouldin index minimization [5].

4.5. Statistical measure for cluster characterization

In the sequel, the word cluster refers to a group of neurons that share the same label and which define a contiguous area on the map. The test-value, proposed in [12] was used to identify dimensions that were relevant for each cluster. Intuitively, it indicates how different a cluster is from the whole population according to the feature considered. Thus, the more different is the feature from the whole population the more relevant it is to describe that cluster. It is defined by

$$t_k = \frac{(\mu - \mu_k)}{\sigma_k} \quad (5)$$

where, μ is the mean of the whole dataset and, μ_k and σ_k are respectively the mean and standard deviation of the class k. Therefore, to interpret subsequences seemed to us more natural than to interpret the dynamic covariance matrices. So rather than directly use subsets of the covariance matrices, we used subsets of the corresponding subsequences. Thus we are able to quantify the relative importance of each point, at each step in time, for the different clusters.

5. Data pre-processing

Analysis was focused on facial motion; therefore, the gradients of the coordinate points were computed. Then, to eliminate the structural cue to individuals with a larger face who have a longer shift, the gradients were normalized. Thereafter, sequences of movements were resampled using a sliding window to improve robustness to the time lag of the selected video recording. Nevertheless, it introduced an additional parameter that had to be chosen carefully, namely the width of the temporal window. The observations then had too many dimensions to be used. So, the dynamic covariance matrix of each sub-sequence was computed according the following expression [18, 19]:

$$\Sigma_d = \frac{x_{(1)}x_{(1)}^T + \sum_{i=2}^{W}\left(x_{(i)} - \overline{x}_{(i)}\right)\left(x_{(i)} - \overline{x}_{(i)}\right)^T}{W} \quad (6)$$

with $\overline{x}_{(i)} = \frac{1}{i}\sum_{j=1}^{i} x_{(j)}$. Thus, the dimension of the data only depends only on the number of face points considered.

6. Experiments and results

Our objective was to verify whether facial movements are related to the subject gender or not. Thus, it appeared relevant to select the parameter value that involved the best separation between the two classes. A cross-validation was adopted to evaluate values from 1 to 38 and each evaluation was repeated 5 times. The SOM that were trained with the dynamic covariance matrices from nine of the ten subjects was labelled. Then, the labelling of the map was evaluated by comparing the label from the remaining data with their best matching unit label.

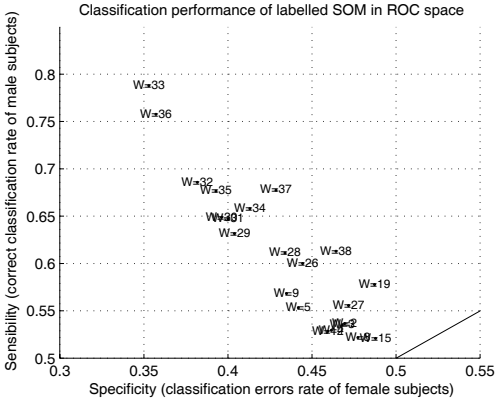

Figure 2: SOM based classifiers performance

Point	Important Move	Motionless
1	0.06	0.10
3	0.07	0.08
6	0.07	0.09
13	0.09	0.09
21	0.08	0.10

Table 1: Significant test values for Female

Receiver Operating Characteristics (ROC) graphs are a useful technique for visualizing, organizing and selecting classifiers based on their performance [7]. Thus, performances of SOM based classifiers are given in the ROC space. For convenience, only ones with more than 50% correct classification rate of both gender have been represented on figure 2. The nearest point W=33 from the upper left corner corresponds to a 33 time units sliding window. So this value has been retained for the remaining exploratory analysis of our data.

Figure 3 shows the distribution of subsequences gender over the final map and the segmentation obtained using the Davies-Bouldin index. On the left hand side, dark and light grey represents respectively female and male neurons, while black colour stands for unlabelled neurons. On the right hand side, 1 and 2 respectively stands for male and female, and the number between parenthesis indicates the number of hits.

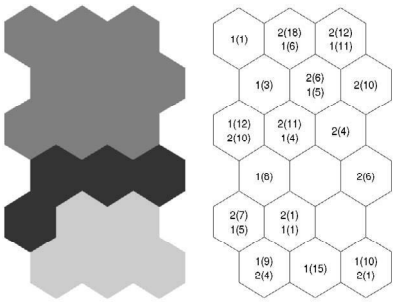

Figure 3: Final map

The tables 1 and 2, and the figures 4 and 5 show the significant test values for each points considered at each time step. Columns on the right, indicate the significance of the corresponding points for the whole subsequences. A visual inspection of the test values indicates that male produce more structured movement than female.

A deeper analysis of the test values emphasized that female cluster (respectively male cluster) is characterized by more structured movements of points 1, 3, 6, and 13 (respectively 8, 17 and 33). It should be highlighted that points 1, 3 and 13 are from the right part of the face whereas points 8, 17 and 33 are on the left part.

Point	Important Move	Motionless
8	0.14	0.23
17	0.14	0.19
21	0.14	0.21
33	0.14	0.28

Table 2: Significant test values for Male

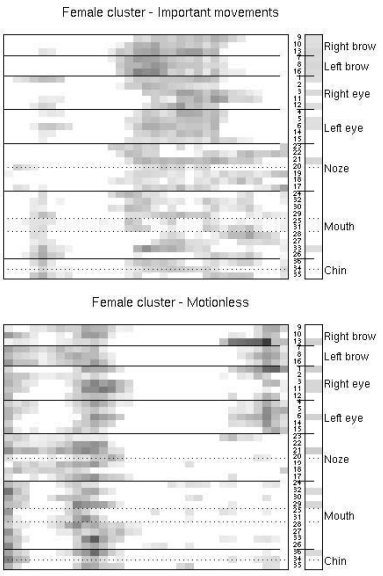

Figure 4: Test values for Female

7. General discussion

The two approaches presented above had involved similar conclusions. On one hand, male facial movements appear more structured than female ones. On the other hand, points the more implicated in movements seems to differ from one gender to an other. Our results led us to hypothesise that the lateralization of facial movements should be an important feature to discriminate ones gender. Anyway, experiments should be repeated with a larger sample of population and with subjects from more different culture to confirm our hypothesis.

8. Conclusion and further research

In this paper, we have presented results from a pilot study with both an ethological and a connectionist approaches which had involved similar and complementary conclusions. Moreover, we chose a quite simple connectionist model for this first study, nevertheless, more elaborated connectionist model have been developed to integrate the temporal dimension of our data [6, 18, 19] and should

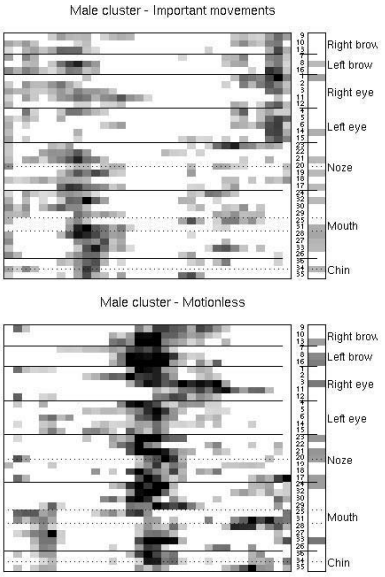

Figure 5: Test values for Male

be considered in future work.

9. References

[1] Anolli L, Duncan S, Magnusson M, Riva G, editors. The hidden structure of interaction: from neurons to culture patterns. IOS Press; 2005.

[2] Berry D. S. Child and adult sensitivity to gender information in patterns of facial motion. Ecological Psychology 1991; 3(4):349-366.

[3] Bruce V, Burton AM, Hanna E, and al. Sex discrimination : how do we tell the difference between male and female faces ? Perception 1993, 22(2):131-152.

[4] Bruce V, Young A. In the Eye of the Beholder: The Science of Face Perception. Oxford: Oxford University Press; 1998.

[5] Davies DL, Bouldin DW. A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 1979, 1(2):224–227.

[6] Euliano N. Temporal Self-Organization for Neural Networks. PhD Thesis, University of Florida, USA; 1998.

[7] Fawcett T. ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. HP Labs Tech Report HPL-2003-4; 2003.

[8] Giovanoli P, Tzou C-H J, Ploner M, Frey M. Three-dimensional video analysis of facial movements in healthy volunteers. British Journal of Plastic Surgery 2003, 56(7): 644-652.

[9] Hill H, Johnston A. Categorizing sex and identity from the biological motion of faces. Current Biology 2001, 11(11):880-885.

[10] Kohonen T. Self-Organizing Maps, Third Extended Edition. Berlin, Heidelberg, New York: Springer; 2001.

[11] Magnusson MS. Discovering hidden time patterns in behaviour: T-patterns and their detection. Behavior research methods, instruments and computers : a journal of the Psychonomic Society, Inc. 2000, 32(1):93-110.

[12] Morineau A. Note sur la caractérisation statistique d'une classe et les valeurs-tests. Bulletin technique n 2, p.20-27. Centre international de statistique et d'informatique appliquées, Saint-Mandé, France; 1984.

[13] Thornton IM, Kourtzi Z. A matching advantage for dynamic human faces. Perception 2002, 31(1):113-132.

[14] Tzou C-H J, Giovanoli P, Ploner M, Frey M. Are there ethnic differences of facial movements between Europeans ans Asians? British Journal of Plastic Surgery 2005, 58(2):186-195.

[15] Vesanto J. SOM-Based Data Visualization Methods. Intelligent Data Analysis 1999, 3(2):111-126.

[16] Vesanto J, Himberg J, Alhoniemi E, Parhankangas J. Self-Organizing Map in Matlab: the SOM Toolbox. In: Proceedings of the Matlab DSL Conference; Espoo, Finland; 1999. p. 35-40.

[17] Vesanto J, Alhoniemi E. Clustering of the Self-Organizing Map. IEEE Transactions on Neural Networks 2000, 11(3):586-600.

[18] Zehraoui F, Bennani Y. M-SOM: Matricial Self Organizing Map for sequences clustering and classification. In: Proceeding of the International Joint Conference on Neural Network, IJCNN'04; Budapest, Hungary. 2004.

[19] Zehraoui F, Bennani Y. M-SOM-ART: Growing Self Organizing Map for Sequences Clustering and Classification. In: Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004; 2004 Aug 22-27; Valencia, Spain; 2004. p. 564-570.

SELECTION OF CLUSTERS NUMBER AND FEATURES SUBSET DURING A TWO-LEVELS CLUSTERING TASK

Sébastien Guérif and Younès Bennani
Université Paris 13, LIPN - CNRS UMR 7030
F-93430 Villetaneuse, France
{sebastien.guerif,younes.bennani}@lipn.univ-paris13.fr

ABSTRACT
Simultaneous selection of the number of clusters and of a relevant subset of features is part of data mining challenges. A new approach is proposed to address this difficult issue. It takes benefits of both two-levels clustering approaches and wrapper features selection algorithms. On the one hands, the former enhances the robustness to outliers and to reduce the running time of the algorithm. On the other hands, wrapper features selection (FS) approaches are known to given better results than filter FS methods because the algorithm that uses the data is taken into account. First, a Self-Organizing Maps (SOM), trained using the original data sets, is clustered using k-means and the Davies-Bouldin index to determinate the best number of a clusters. Then, an individual pertinence measure guides the backward elimination procedure and the feature mutual pertinence is measure using a collective pertinence based on the quality of the clustering.

KEY WORDS
Clustering, feature selection, self-organizing maps, model selection

1 Introduction

During the last decade, it became obvious that adapted tools are needed to exploit more and more huge companies databases. Actually, databases contain important hidden knowledge and the matter of data mining is to emphasize it. The curse of dimensionality problem states that the number of needed examples for training grows exponentionnaly with the dimensionality of the data. That way, whereas Knowledge Discovery from Database (KDD) is only possible because of the data redundancy, too many redundant features stand in the way of the nuggets discovery. This issue can be addressed by one of the two main approaches, namely, features extraction or feature selection.
The former presents a major drawback, actually, an important effort from the user is required to interpret and understand the new representation his data. Among the techniques of this category, the most widely used are probably Principal Component Analysis (PCA) [1, 2] which suffers from numerical instabilities whenever the correlation of the data is ill-conditionned. Moreover, this methods assume that the most relevant dimensions are those with the largest variance which not always the case as it is showed by the figure 1. Other approaches that does not suffer from the same numerical instabilities has been proposed [3] although the features extracted are not as intuitive as the original features. Whereas, the problem of feature selection

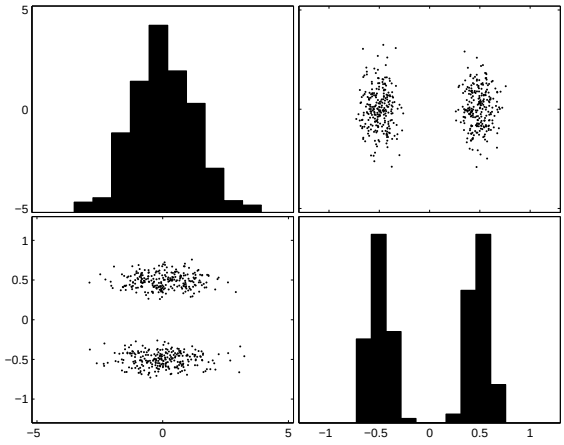

Figure 1. The feature variance is not always a relevant pertinence measure; actually, in this example, whereas $\sigma^2(X) = 1.03$ and $\sigma^2(Y) = 0.25$, the best separation is provided by the Y axis.

had been widely studied in the context of supervised learning, it gains researchers interest more recently in the context of unsupervised learning. In the context of supervised learning, feature selection is driven by the main purpose : achieve better accuracy on unseen data. Nevertheless, in the unsupervised learning framework, the issue is very different because neither the data labels nor their number are available. Therefore, the notion of feature relevance is not as obvious the latter context as in the former context. Anyway, selection of a relevant features subset remains a crucial stake for the data-mining techniques. In this paper, we propose an original method to find both the right number of clusters and the respective subset of features. Our approach is based on both the Davies-Bouldin index [4, 5] and the Test Values [6]. It is assumed that features that does not participate in the structure identified are irrelevant and should be thrusted away from the subset of features selected.

The rest of this paper is organized as follows. The two-levels clustering approache used is presented in section 2. Then, the feature selection method proposed is presented in section 3. Finally, some experimental results are given befor to conclude.

2 Method

2.1 Self-Organizing Maps

SOM was introduced by Pr. Teuvo Kohonen in the early 80's as a convenient clustering and visualization tool. High-dimensional data are projected on a low dimension discrete space, called the topological map, preserving the local topology of the initial space; thus, the observations which are close to each other are projected on a localized area. A map should be viewed as a set of neurons (or units), organized according to a grid that defines their neighbourhood relationships. Each neuron is associated to one point of the observations' space: its prototype.

Self-Organizing Maps (SOM) implement a particular form of competitive artificial neural networks; when an observation is recognized, activation of an output cell competition layer leads to inhibit activation of other neurons and reinforce itself. It is said that it follows the so called *Winner Takes All* rule. Actually, neurons are specialized in the recognition of one kind of observations. The learning is unsupervised because neither the classes nor their number is fixed a priori. A SOM consists in a two dimensional layer of neurons which are connected to the inputs with exciting connections and to their neighbors with inhibiting links.

The training set is used to organize these maps under topological constraints of the input space. Thus, a mapping between the input space and the network space is constructed; closed observations in the input space would activate two closed units of the SOM. An optimal spatial organization is determined by the SOM from the received information, and when the dimension of the input space is lower than three, both position of weights vectors and direct neighbourhood relations between cells can be represented visually.

2.2 Learning algorithms

For convenience, let us mention some notations : let N be the number of sample points in the data set Ω, n be the number of features in the original feature set F, r be the number of features in the reduced feature set F_R, M be the size of the map units set U and ω_j be the prototype of the j^{th} unit.

Connectionist learning is often presented as a minimization of a risk function (cost function). In our case, it will be carried out by the minimization of the distance between the input samples and the map prototypes (referents), weighted by a neighbourhood function h_{ij}. The criterion to be minimized is defined by:

$$R_{SOM} = \frac{1}{N} \sum_{x_i \in \Omega} \sum_{j \in U} h_{b_i j} . \left\| \omega_j - x_i \right\|^2 \quad (1)$$

where b_i is the *Best Matching Unit* (BMU) of the sample point $x_i \in \Omega$ and is defined as the unit with the closest prototype:

$$b_i = \arg\min_{j \in U} \{ \left\| \omega_j - x_i \right\|^2 \}$$

In our experiments, we use the gaussian neighborhood function h defined

$$h_{ij} = \exp\left(-\frac{d^2(i,j)}{2.\sigma^2(t)} \right)$$

where $d(i,j)$ is the distance between units i and j on the map and $\sigma(t)$ is a decreasing function that defines the size of the neighborhood considered at step t.

Two main approaches can be used to optimize the criterion mentionned above, namely the *on-line algorithm* and the *batch algorithm*. Whereas the latter suffers from several drawbacks [7], it provides faster convergence. So we choose the batch Kohonen's algorithm [8] because our approach necessitates several running of the learning of the learning algorithm. The weights of all the neurons are updated until stabilization according to the following adaptation rules:

$$\omega_j(t+1) = \frac{\sum_{i \in \Omega} h_{b_i j} x_i}{\sum_{i \in \Omega} h_{b_i j}} \quad (2)$$

2.3 SOM segmentation

Whereas both agglomerative and partitive clustering algorithm have been successfully applied to the segmentation of SOM [9], several specific approaches have been proposed to take into account the topological ordering of the unit maps. They rely on either the contiguity study [10] or the U-matrix (the matrix of distances between adjacent map units) [11, 12, 13]. We adopted the *kmeans* based approach proposed by J. Vesanto [9]. Although the number of clusters is needed to run the *kmeans* algorithm, it is not known in the unsupervised learning framework. So several values should be tried and the best one according to the Davies-Bouldin index [4] is selected. Assuming that C, $S_c(k)$ and $d_{ce}(k,l)$ respectively refers to the number of clusters, the mean quantization error in cluster k and the distance between the centers of clusters k and l, the Davies-Bouldin index is defined by

$$I_{DB} = \frac{1}{C} \sum_{k=1}^{C} \max_{l \neq k} \left\{ \frac{(S_c(k) + S_c(l))}{d_{ce}(k,l)} \right\}$$

It should be noticed that the kmeans algorithm is a special case of the SOM training algorithm when no neighborhood constraints are imposed to the center. In other words, the neighborhood function $h_{b_i j}$ is replaced by the chronecker symbol $\delta_{b_i j}$.

3 Feature Selection

Feature Selection necessitates three essential elements [14]:

- A pertinence measure
- A search procedure
- A stop criterion

3.1 Pertinence measure

Whereas in the supervised learning case, a pertinence measure can be easily defines using the performance of the model in the task it has been designed to, in the unsupervised learning framework, it is not possible anymore.
So we have to define new criteria. We propose to use two different feature evaluation criteria : an individual criteria, $R_{individual}(j)$, to guide the search procedure and a collective criteria, $R_{collective}(j)$, to take the mutual relevance of features.
We propose to select features that involve a good clustering; thus, the SOM is segmented using the method presented above and the test-values [6] of each feature according each cluster are computed. Therefore, the maximum of absolute test values along the the different clusters is used as an individual relevance measure. The first individual relevance criteria is defined by

$$R_{individual}(j) \quad = \quad \max_{k=1,\ldots,C} \left\{ \left| \frac{\mu_{kj} - \mu_j}{\sigma_{kj}} \right| \right\} \qquad (3)$$

where C, μ_j, μ_{kj} and σ_{kj} are respectively the number of clusters, the mean values of the feature j in the whole data set and in the cluster k, and the standard deviation of feature j in the cluster k.
Then, whenever the removing of a feature involves an increasing of the I_{DB}, we consider that it is relevant according the current clustering. Thus, we define the collective relevance of a feature as the increasing of the I_{DB} involved by its removing :

$$R_{collective}(j) \quad = \quad I_{DB} - I_{DB}|_{F_R \setminus \{j\}} \qquad (4)$$

where $I_{DB}|_{F_R \setminus \{j\}}$ is the Davies-Bouldin index evaluated without taking in account the feature j.
Whereas these criteria have been successfully apply to several data set from UCI [15], they present some drawbacks. On the one hand, they rely on the kmeans algorithm which is well known for its strong dependance with the initial centers. So, to insure the reliability of the result several running of the algorithm have to be done at each step of the feature selection procedure and for each possible number of clusters. On the other hand, when many features are noisy or irrelevant, they may prevent kmeans algorithm and Davies-Bouldin to identified the right clusters; therefore the feature selection procedure might fail. Two other criteria which avoid the additional computational cost due to the map segmentation and the possible weak of robustness of the above criteria are presented in the next paragraph.

3.2 Search procedure

To find an optimal solution requires either an exhaustive search or the monotonicity of the pertinence measure. On the ones hand, the former involves the pertinence evaluation of 2^n subsets where n is the number of features and it becomes infeasible since n is large. On the other hand, the latter is difficult to insure. We propose a Backward Elimination procedure that takes into account both the individual and the collective pertinence measures defined in the previous section. It begins with the whole features set and progressively eliminates the less interesting features. The individual measure guides the selection and the collective pertinence insures that the removing of the feature candidate do not alter the quality of the model. The threshold θ in the algorithm 1 is used to balance the relative importance of the two pertinence measures.

Algorithm 1 Feature Selection Procedure

```
F_R ← F
while (¬stopping criterion) do
  Build a model.
  Evaluate individual relevance R_individual(j)
  Sort features according ascending individual relevance ordering
  found ← false
  while (¬found) do
    Evaluate the collective criterion R_collective(j) of the less relevant feature according individual criterion
    if (R_collective(j) ≤ θ) then
      found ← true
      R ← R \ {j}
    end if
  end while
  if (¬found) then
    j ← arg min_{k∈R} {R_collective(k)}
    R ← R \ {j}
  end if
end while
```

3.3 Stop criterion

We use the statistic criterion proposed by T. Cibas [16] to evaluate whether a feature subset gives any additionnal information according another one. Therefore, the backward elimination procedure is stopped since the removing of the feature selected involves a loss of information.
Assuming that F, the set of features, and $F \setminus F_R$, the removed features subset, are distributed according a gaussian law

$$N(\mu^{(k)}, \Sigma) \ : \ k = 1, \ldots, C$$

where $\mu^{(k)}$, the mean of the features from F in the cluster

k, and Σ, the covariance matrices, are defined as follows

$$\mu^{(k)} = \left(\mu_1^{(k)}, \mu_2^{(k)}\right), \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

where 1 and 2 as index respectively stand for F_R and $F \setminus F_R$. Then, the null hypothesis which says that $F \setminus F_R$ does not give any additionnal information than F_R is expressed as follows :

$$H_0 : \mu_2^{(k)} - \mu_2^{(h)} - \Sigma_{21}\Sigma_{11}^{-1}\left(\mu_1^{(k)} - \mu_1^{(h)}\right) = 0 \qquad (5)$$

with $k \neq h = 1, \ldots, C$.
A test of this hypothesis is based on Wilks statistics. Let B and W be respectively the between and the within covariance matrices :

$$\begin{aligned} B &= \sum_{k=1}^{C} N^{(k)} \left(\mu^{(k)} - \overline{\mu}\right)\left(\mu^{(k)} - \overline{\mu}\right)^T \\ W &= \sum_{k=1}^{C}\sum_{i=1}^{N^{(k)}} \left(x_i^{(k)} - \mu^{(k)}\right)\left(x_i^{(k)} - \mu^{(k)}\right)^T \end{aligned}$$

where $N^{(k)}$ is the number of elements in the cluster k and $\overline{\mu}$ is the mean of the features from F for the whole sample. Then, the same block decomposition as for Σ can be applied to the matrices B, W and their sum T :

$$\begin{aligned} B &= \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \\ W &= \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \\ T &= B + W = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} \end{aligned}$$

Therefore, the determinants of the matrices W and T can be written

$$\begin{aligned} |W| &= |W_{11}|\left|W_{22} - W_{21}W_{11}^{-1}W_{12}\right| \\ |T| &= |T_{11}|\left|T_{22} - T_{21}T_{11}^{-1}T_{12}\right| \end{aligned}$$

Thus, we denote

$$K = \frac{\left|W_{22} - W_{21}W_{11}^{-1}W_{12}\right|}{\left|T_{22} - T_{21}T_{11}^{-1}T_{12}\right|}$$

which has $\frac{(N-C-r)}{(C-1)}$ degrees of freedom. With the above notations, the Wilks statistics for n variables are :

$$\begin{aligned} \Lambda_F &= \frac{|W|}{|T|} \\ &= K.\frac{|W_{11}|}{|T_{11}|} \\ &= K.\Lambda_{F_R} \end{aligned}$$

which shows that, with a small value of K, the clusters separability is larger with n than r features. Therefore, the null hypothesis (5) is true if and if only features from F_R involve the same separability as the whole features set F. Then, the Wilks statistic Λ is equivalent to the Fisher-Snedecor one :

$$F_s = \frac{(N-C-r)}{(C-1)}\frac{1-K}{K}$$

which is distributed according $F\left(C-1, N-C-r\right)$

4 Experiments and results

The method presented above has been apply to several commonly used UCI machine learning data sets [15]. Whereas the data labels havent't been used during the learning stage, they can be used for evaluation purpose; actually, the ability of our approach to identified the true clusters can be measured using the following criterion :

- the number of identified clusters refered by C_T
- the couple error which measures how far the discovered partition is from the *true* classes and is defined by $E_C = \frac{2}{N(N-1)}\sum_{(i,j)\in\{1,\ldots,N\}^2,\ i<j}\epsilon_{ij}$ where ϵ_{ij} is null when samples points i and j are either grouped or separated in both true and discovered partitions.
- the Purity of clusters in term of known classes $P_R = \frac{1}{N}\sum_{k=1}^{C_T} maxM_k$ where M is the confusion matrix.

In our experiments, we used the *batch* Kohonen's algorithm and the *fast global k-means* algorithm [17] which are both deterministic. For each of the data sets considered, we run five 10-folds validation and we summarized the results obtained in Table 1. Then, the figure 4 shows the evolution of the Davie-Bouldin index during the feature selection process. The last model index value can be considered as an outlier, therefore, the best model according to the Davies-Bouldin index is obtained when five features have been removed. Nevertheless, our stop criterion indicates that the model with eleven removed features should be retained.

5 Conclusion

A new approach to select both the number of clusters and the related features subset has been proposed in an unsupervised learning framework. Whereas the preliminary results are encourageous, the stop criterion proposed can not always be uses. For instance, it requires that $N - c \geq p$, where N, c and p are respectively the number of map units, the number of identified clusters and the total number of features, to insure that the within covariance matrix W is not singular. Research work are on the way to enhance the proposed method to data sets with more features than observations.

				Training set		Testing set	
		C_T [σ_{C_T}]	n_{FS} [$\sigma_{n_{FS}}$]	E_C [σ_{E_C}]	P_R [σ_{P_R}]	E_C [σ_{E_C}]	P_R [σ_{P_R}]
Glass	F	7.04 [0.73]	9.0 [–]	0.301 [0.012]	56.25 [2.56]	0.295 [0.068]	67.52 [9.01]
189 - 21	F_R	5.10 [1.83]	2.84 [1.46]	0.376 [0.082]	50.83 [6.54]	0.382 [0.121]	58.38 [10.40]
Wine	F	6.86 [0.81]	13.0 [–]	0.171 [0.022]	93.59 [1.97]	0.165 [0.064]	95.28 [5.11]
189 - 21	F_R	5.70 [2.34]	6.3 [2.1]	0.247 [0.060]	80.32 [12.02]	0.239 [0.096]	83.44 [13.78]
Cancer	F	9.72 [0.67]	30.0 [–]	0.414 [0.014]	93.83 [1.56]	0.417 [0.026]	94.16 [3.03]
242 - 27	F_R	2.72 [1.96]	12.4 [3.3]	0.182 [0.077]	91.53 [1.04]	0.184 [0.091]	91.60 [3.49]
Wave	F	6.18 [2.56]	40.0 [–]	0.304 [0.016]	68.64 [8.48]	0.309 [0.014]	66.17 [7.82]
500 - 4500	F_R	4.82 [1.55]	28.2 [9.56]	0.304 [0.020]	66.93 [6.62]	0.306 [0.018]	65.97 [6.68]

Table 1. The two numbers under the data set name indicates the size of the training and testing sets respectively. Then F and F_R stands for the whole features set and the reduced subset selected.

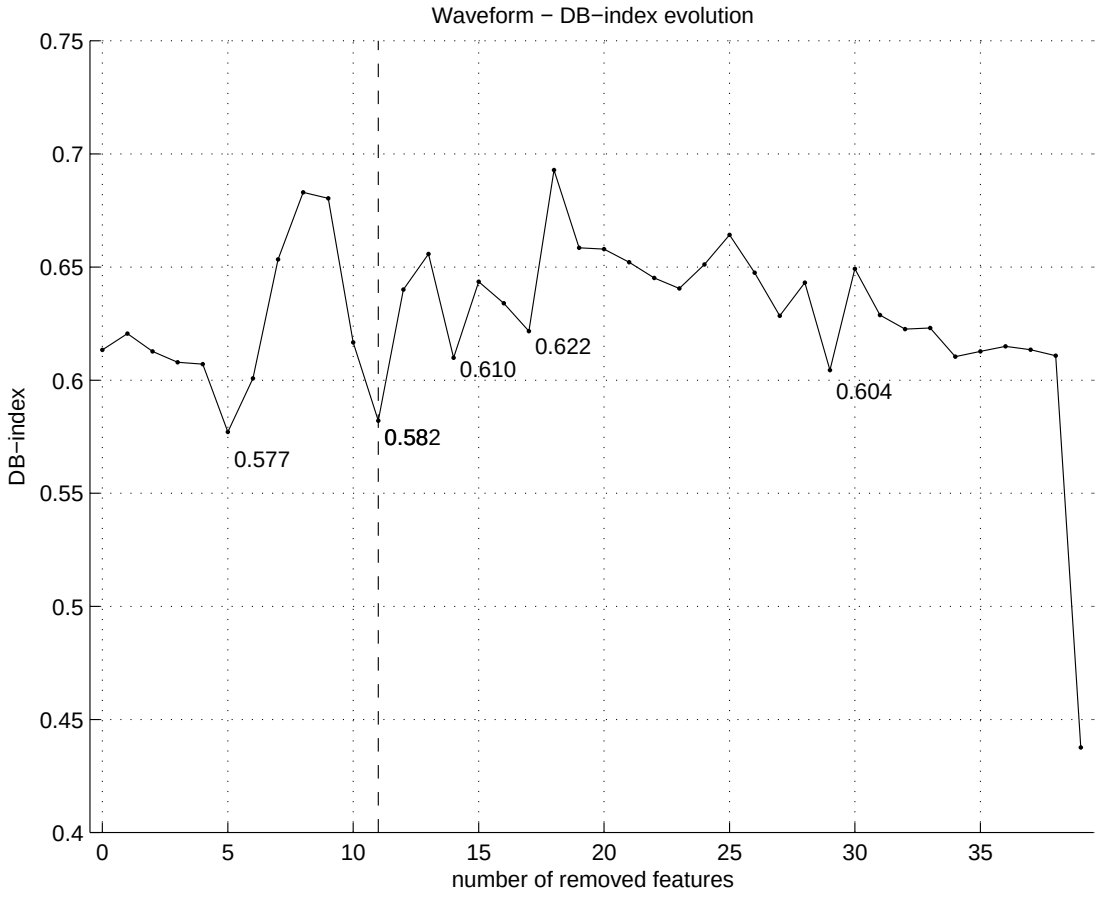

Figure 2. Evolution of the Davies-Bouldin index during the backward features elimination procedure : the vertical dash line indicates the model retained by our stop criterion and some of the best index values are indicated too.

References

[1] L. Lebart, A. Morineau, and M. Piron. *Statistique exploratoire multidimensionnelle*. Éditions Dunod, 1995.

[2] G. Saporta. *Probabilités, analyse des données et statistique*. Editions Technip, Paris, France, 1990.

[3] S. K. Pal, R. K. De, and J. Basak. Unsupervised feature evaluation: A neuro-fuzzy approach. *IEEE Transactions on Neural Networks*, 11(2):366–376, 2000.

[4] D. Davies and D. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 1(2):224–227, 1979.

[5] J. Vesanto and E. Alhoniemi. Clustering of the self-organizing map. *IEEE-NN*, 11(3):586–600, May 2000.

[6] A. Morineau. Note sur la caractérisation statistique d'une classe et les valeurs-tests, 1984.

[7] J-C. Fort, P. Letrémy, and M. Cottrell. Advantages and drawbacks of the batch kohonen algorithm. In M.Verleysen Ed., editor, *ESANN'2002 Proceedings, European Symposium on Artificial Neural Networks, Bruges (Belgium)*, pages 223–230, Bruxelles, Belgium, 2002. Editions D Facto.

[8] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[9] J. Vesanto and E. Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586–600, 2000.

[10] F. Murtagh. Interpreting the Kohonen self-organizing feature map using contiguity-constrained clustering. *Pattern Recognition Letters*, 16(4):399–408, April 1995.

[11] F. Moutarde and A. Ultsch. U*F clustering: a new performant cluster-mining method based on segmentation of Self-Organizing Maps. In *Proceedings of the 5th Workshop On Self-Organizing Maps (WSOM'05)*, pages 25–32, Paris 1 Panthéon-Sorbonne University, France, September 2005.

[12] D. Opolon and F. Moutarde. Fast semi-automatic segmentation algorithm for Self-Organizing Maps. In *Proceedings of ESANN'2004 , European Symposium on Artificial Neural Networks, Bruges (Belgium)*, pages 507–512, 2004.

[13] A. Ultsch. Clustering with SOM: U*C. In *Proceedings of the 5th Workshop On Self-Organizing Maps (WSOM'05)*, pages 75–82, Paris 1 Panthéon-Sorbonne University, France, September 2005.

[14] D. Cakmakov and Y. Bennani. *Feature Selection for Pattern Recognition*. Informa, Skopje, Macedonia, 2002.

[15] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[16] T. Cibas. *Contrôle de la complexité dans les réseaux de neurones : régularisation et sélection de caractéristiques*. PhD thesis, University of Paris XI Orsay, Paris, France, December 1996.

[17] J. J. Verbeek. *Mixture Models for Clustering and Dimension Reduction*. PhD thesis, University of Amsterdam, Amsterdam, The Netherlands, December 2004.

Réduction de dimension en Apprentissage Numérique Non Supervisé

Sébastien GUÉRIF

Résumé

La classification automatique - *clustering* - est une étape importante du processus d'extraction de connaissances à partir de données (ECD). Elle vise à découvrir la structure intrinsèque d'un ensemble d'objets en formant des regroupements - *clusters* - qui partagent des caractéristiques similaires. La complexité de cette tâche s'est fortement accrue ces deux dernières décennies lorsque les masses de données disponibles ont vu leur volume exploser. En effet, le nombre d'objets présents dans les bases de données a fortement augmenté mais également la taille de leur description. L'augmentation de la dimension des données a des conséquences non négligeables sur les traitements classiquement mis en oeuvre : outre l'augmentation naturelle des temps de traitements, les approches classiques s'avèrent parfois inadaptées en présence de bruit ou de redondance. Dans cette thèse, nous nous intéressons à la réduction de dimension dans le cadre de la classification non supervisée. Différentes approches de sélection ou de pondération de variables sont proposées pour traiter les problèmes liés à la présence d'attributs redondants ou d'attributs fortement bruités :

- Nous proposons d'abord l'algorithme μ-SOM qui limite l'effet de la présence d'attributs redondants en calculant une pondération des attributs à partir d'une classification simultanée des objets et des attributs.
- Nous présentons ensuite une approche intégrée – *embedded* – de sélection de variables pour la classification automatique qui permet de découvrir à la fois le nombre de groupes d'objets présents dans les données mais aussi un sous-ensemble d'attributs pertinents.
- Nous terminons en présentant l'algorithme ω^{β}-SOM qui introduit une pondération des attributs dans la fonction de coût des cartes auto-organisatrices - *Self Organizing Maps* - qui est ensuite optimisée itérativement en alternant trois étapes : optimisation des affectations, optimisation des prototypes et optimisation des poids. La pondération obtenue après convergence est ensuite utilisée pour proposer une approche filtre - *Filter* - de sélection de variables.

Nous concluons cette thèse en indiquant les limites des approches proposées et envisageant quelques axes à développer lors de la poursuite ces recherches.