Partitionnement des données pour les problèmes de classement difficiles: Combinaison des cartes topologiques mixtes et SVM

Mustapha Lebbah[*], Mohamed Ramzi Temanni[*,**], Christine Poitou-Bernert[**,***,****]
Karine Clement[**,***,****], Jean-Daniel Zucker[*,**]

[*] Université Paris 13, UFR de Santé,
Médecine et Biologie Humaine (SMBH) - Léonard de Vinci- LIM&BIO
74, rue Marcel Cachin 93017 Bobigny Cedex France
nom@limbio-paris13.org,
http://www.limbio-paris13.org
[**] Inserm, U755 Nutriomique, 75004 Paris, France;
[***] University Pierre and Marie Curie-Paris 6, Faculty of Medicine,
Les Cordeliers, 75004 Paris, France;
[****] AP-HP, Hôtel-Dieu Hospital, Nutrition department,
1 Place du parvis Notre-Dame, 75004 Paris, France
prénom.nom@htp.aphp.fr

Résumé. Dans ce papier, nous présentons un modèle pour aborder les problèmes de classement difficiles. Ces problèmes ont souvent la particularité d'avoir des taux d'erreurs en généralisations très élevés et ce quelles que soient les méthodes utilisées. Pour ce genre de problème, nous proposons d'utiliser un modèle de classement combinant le modèle de partitionnement des cartes topologiques mixtes et les machines à vecteur de supports (SVM). Le modèle non supervisé est dédié à la visualisation et au partitionnement des données composées de variables quantitatives et/ou qualitatives. Le deuxième modèle supervisé, est dédié au classement. Dans ce papier, nous présentons une combinaison de deux modèles qui permettent d'améliorer la visualisation des données et d'augmenter les performances en classement. Ce modèle consiste à entraîner des cartes auto-organisatrices pour construire une partition organisée des données, constituée de plusieurs sous-ensembles qui vont servir à reformuler le problème de classement initial en sous-problème de classement. Pour chaque sous-ensemble, on entraîne un classeur SVM spécifique. Pour la validation de notre modèle (CT-SVM), nous avons utilisé quatre jeux de données. La première base est un extrait d'une grande base médicale sur l'étude de l'obésité à l'Hôpital Hôtel-Dieu de Paris, et les trois dernières bases sont issues de la littérature. Les résultats obtenus montrent l'apport de ce modèle dans la visualisation et le classement de données complexes.

1 Introduction

En apprentissage artificiel, on distingue deux grands thèmes, l'apprentissage supervisé et l'apprentissage non supervisé ; la plupart des problèmes d'apprentissage sont traités par l'une des deux approches. Souvent pour les problèmes de classement, on a recours à de nombreuses méthodes qui sont évaluées sur leur capacité à prédire correctement la classe des observations qui n'ont pas participé à la phase d'apprentissage. Pour l'apprentissage non supervisé, les critères de qualité sont plus difficiles à définir ; ils s'articulent autour de l'interprétation des regroupements ou des partitions obtenues.

Parmi les problèmes d'apprentissage, il existe une catégorie de problèmes qui sont appelés dans la littérature : difficiles, complexes, hétérogènes. Ces problèmes ont souvent la particularité d'avoir des résultats non satisfaisants quelques soit la méthode standard utilisée (arbre de décision, SVM, MLP,...). En d'autres termes, ces méthodes obtiennent légèrement un meilleur résultat qu'un algorithme qui ferait un tirage aléatoire. Dans d'autres problèmes, appelés aussi problèmes difficiles, on dispose de données issues de sources différentes ou des données mixtes. Ces problèmes existent dans de nombreux domaines ; souvent ils sont mal posés, les bases d'apprentissage disposent de peu d'observations et/ou le nombre de variables est trop grand. Parmi ces domaines, on retrouve le domaine médical, spécialement les bases cliniques ou les bases biopuces,(Weston et al (2001); Tamayo et al (1999)). De nombreuses méthodes sont développées pour ce genre de problème qui consiste à utiliser les techniques de sélection de variables (Golub et al (1999); Xing et al (2001); Peng et al (2005)), la reformulation de problème en utilisant le classement heuristique, Clancey (1985), et le boosting (Philip et al (2003)). D'autres méthodes consistent à combiner ou fusionner les classifieurs Egmont-Petersen et al (1999); Liu et al (2001).

Dans ce papier nous présentons un modèle combinant deux modèles d'apprentissages pour aborder les problèmes de classement difficiles ou "complexes". La question qui se pose pour ce genre de base complexe, *faut-il aborder le problème de classement d'une manière globale ou trouver un moyen de le diviser en sous-problèmes ?*. Notre modèle consiste à partitionner les données pour sélectionner le classifieur adéquat pour chaque observation,(Liu et al (2001); Rybnik et al (2003)).

Quelques méthodes spécifiques combinant l'apprentissage non supervisé et supervisé, aussi bien dans le domaine de la classification hiérarchique et les arbres de décision que pour la recherche de partitions ont été développées. Dans le domaine de la classification hiérarchique, Hyun-Chul et al (2003); Benabdeslem (2006) proposent des modèles combinant la classification hiérarchique en entraînant un SVM binaire dans chaque nœud du dendogramme de la classification hiérarchique. Lebrun et al (2004); Sungmoon et al (2004); Shaoning et al (2005) proposent d'utiliser une autre méthode de partitionnement qui permet de construire une partition de sous ensembles ordonnés sous forme d'arbre binaire afin d'apprendre un SVM binaire au niveau de chaque nœud de l'arbre.

Dans Wu et al (2004), les auteurs proposent d'utiliser les cartes topologiques de Kohonen (Kohonen (1995)) pour filtrer les données. Les observations non étiquetées héritent de l'étiquette de la classe du vote majoritaire de son sous-ensemble. A la fin de cette phase, un seul SVM

est appris sur l'ensemble d'apprentissage initial réétiqueté, sans prendre en compte la partition de données. Nous trouvons aussi l'utilisation d'autre méthodes de partitionnement comme le k-means lorsqu ?un sous-ensemble de la partition est constitué à 100% d'une seule classe alors toutes les observations de ce sous-ensemble sont remplacées par leur référent (représentant), calculé par le K-means (K-moyennes), dans la base d'apprentissage dédiée au SVM. Ce procédé permet de réduire significativement la taille de la base et par conséquent le temps de calcul sur de grandes bases de données. D'autres méthodes sont aussi inspirées des méthodes de partitionnement et de classement comme la définition de cartes topologiques dans l'espace de redescription (Sungmoon et al (2004)) ou l'utilisation des vecteurs supports pour définir une partition (Ben-Hur et al (2001)).

Toutes ces méthodes ont un point commun, celui de montrer que la visualisation et le prétraitement des données sont des étapes importantes dans la phase exploratoire de l'analyse de données. Cette phase permet d'inclure les connaissances de l'expert du domaine avant la phase de classement. La difficulté en classement augmente d'autant plus qu'il s'agit de données complexes, par exemple données mixtes (quantitatives et qualitatives) ou des données biomédicales, pour lesquelles il existe moins de méthodes standards.

Notre approche consiste à diviser le problème global de classement en sous-problème de classement guidé par la structure et l'organisation des données de la base dans l'espace des données. Ce modèle est basé sur le partitionnement des données, avec une méthode non supervisée, en une partition constituée de plusieurs sous-ensembles organisés, en tenant en compte de la typologie des données, qui vont servir à définir un classifieur pour chacun en utilisant les SVMs, Vapnik (1995). La tâche de partitionnement des données de notre modèle est réalisée à l'aide des cartes auto-organisatrices (Kohonen (1995); Lebbah et al (2005)).

Les cartes topologiques sont utilisées dans notre modèle parce qu'elles permettent à la fois d'être utilisé comme outils de visualisation et de partitionnement non supervisé de différents types de données (quantitatives et qualitatives). Elles permettent de projeter les données sur des espaces discrets qui sont généralement de dimensions deux. Le modèle de base, proposé par Kohonen (Kohonen (1995)), est uniquement dédié aux données numériques. Des extensions et des reformulations du modèle de Kohonen ont été proposées dans la littérature, bishop et al (1998); Lebbah et al (2000, 2005). Une généralisation des cartes topologiques sera présentée dans ce papier.

Les machines à vecteurs de support ont été développées dans les années 90 par Vapnik (1995). Ces méthodes ont été utilisées dans notre modèle parce qu'elles s'avèrent particulièrement efficaces, car elles peuvent traiter des problèmes mettant en jeu un grand nombre de variables ou un petit nombre d'observations (individus), et qu'elles assurent une solution unique (pas de problèmes de minimum local comme pour les réseaux de neurones). L'algorithme sous sa forme initiale revient à chercher une frontière de décision linéaire entre deux classes, mais ce modèle peut considérablement être enrichi en se projetant dans un autre espace permettant ainsi d'augmenter la séparabilité des données. Ce cas d'utilisation des machines à vecteurs de support est le plus utilisé, car la plupart des problèmes réels sont non linéairement séparables. Dans ce cas, on se sert plutôt de l'astuce du noyau (kernel trick), par exemple : noyau linéaire,

noyau polynomial, noyau Gaussien (Radial). Ces fonctions sont des fonctions non linéaires, elles jouent un rôle similaire au rôle du produit scalaire dans les problèmes d'optimisation, et elles sont vues comme une mesure de similarité.

Pour la compréhension de notre modèle, nous présentons dans la section 1.1 les différentes notations utilisées. Dans la section 2 nous présentons le modèle des cartes topologiques pour l'analyse des données mixtes qui est une généralisation des cartes topologiques classiques de Kohonen, ainsi que l'amélioration apportée à ce modèle, par rapport à la version présentée dans Lebbah et al (2005), pour qu'il prenne en compte la particularité des données mixtes. Pour simplifier la présentation du papier, le modèle SVM ne sera pas présenté. Dans la section 3, nous présentons le modèle que nous proposions pour le classement. Dans la section 4, une validation du modèle sur des données issues de la littérature ainsi que des données médicales réelles. Cette validation permet de démontrer que notre modèle peut êtres utilisé pour augmenter les performances en classement sur certaines bases de données.

1.1 Notations-Définitions

Ce paragraphe introduit les notations de base utilisées. L'ensemble $\mathcal{D}$ représente l'espace des observations ; les observations sont supposées quantitatives ou qualitatives et en dimension multiple ; on suppose que chaque observation est de dimension d. On suppose, par la suite que l'on dispose d'observations correspondant à N individus représentés par l'ensemble des couples $\mathcal{A} = \{(\mathbf{z}_i, y_i); i = 1..N\}$ où l'observation est $\mathbf{z}_i$ et y_i l'étiquette de sa classe. Cette étiquette sera utilisée dans l'apprentissage supervisé (SVM).

La méthode de partitionnement cherche à déterminer une partition de $\mathcal{D}$ en N_{cell} sous-ensembles qui sera notée $\mathcal{P} = \{P_1, ..., P_{N_{cell}}\}$. A chaque sous-ensemble P_c, on associe un vecteur référent $\mathbf{w}_c \in \mathcal{D}$ qui sera le représentant ou le "résumé" de l'ensemble des observations de P_c. Par la suite nous notons $\mathcal{W} = \{\mathbf{w}_c; c = 1..N_{cell}\}$ l'ensemble des vecteurs référents. La partition $\mathcal{P}$ de $\mathcal{D}$ peut être défini d'une manière équivalente avec la fonction d'affectation ϕ qui est une application de $\mathcal{D}$ dans l'ensemble fini des indices $\mathcal{I} = \{1, 2, ..., N_{cell}\}$.

Dans le cas où il y a eu regroupement des sous-ensembles, nous avons défini une application surjective χ de $\mathcal{I}$ dans l'ensemble des indices $\mathcal{J} = \{1, 2, ..., S\}$ où $1 \leq S \leq N_{cell}$. Si on utilise ces définitions, le sous-ensemble P_c est alors représenté par $P_c = \{\mathbf{z} \in \mathcal{D}/\phi(\mathbf{z}) = c, \chi(c) \in \mathcal{J}\}$, (si $\chi(c) = 1$ alors $\mathcal{P} = P_c = \mathcal{A}$). On notera pas la suite l'ensemble des indices $\mathcal{I}_p$ des sous-ensemblepurses tel que $\mathcal{I}_p = \{c/\forall \mathbf{z} \in P_c, \chi(\phi(\mathbf{z})) = c, vote(P_c) = y_c\}$. y_c est l'étiquette du vote majoritaire à 100% du sous-ensemble P_c en utilisant la fonction $vote$.

2 Cartes Topologique Mixtes

On suppose que l'on dispose dans le cas des cartes topologiques de la base d'apprentissage $\mathcal{A}$ sans les étiquettes $\mathcal{A} = \{\mathbf{z}_i; i = 1..N\}$. Les observations $\mathbf{z}_i$ sont composées de deux parties : la partie numérique $\mathbf{z}_i^r = (z_i^{1r}, z_i^{2r}, ..., z_i^{nr})$ ($\mathbf{z}_i^r \in \mathcal{R}^n$), et la partie binaire $\mathbf{z}_i^b = (z_i^{1b}, z_i^{2b}, ..., z_i^{mb})$ ($\mathbf{z}_i^b \in \beta^m = \{0, 1\}^m$). Avec ces notations une observation $\mathbf{z_i} = (\mathbf{z_i^r}, \mathbf{z_i^b})$ est

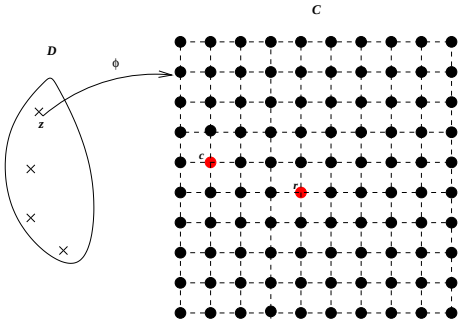


FIG. 1 – *Carte topologique de dimension* 10×10*, (*$\delta(c, r) = 4$*). ϕ est la fonction d'affectation de l'espace des données $\mathcal{D}$ dans l'espace de la carte $\mathcal{C}$.*

de dimension $d = n + m$ (numérique et binaire).

Comme tout modèle de cartes topologiques, nous supposons que l'on dispose d'une carte discrète $\mathcal{C}$ ayant N_{cell} cellules structurées par un graphe non orienté. Cette structure de graphe permet de définir une distance, $\delta(r, c)$ entre deux cellules r et c de $\mathcal{C}$, comme étant la longueur de la plus courte chaîne permettant de relier les cellules r et c, (voir figure 1). Le système de voisinage est défini grâce à la fonction noyau $\mathcal{K}$ ($\mathcal{K} \geq 0$ et $\lim_{|x|\to\infty} \mathcal{K}(x) = 0$). L'influence mutuelle entre deux cellules c et r est définie par la fonction $\mathcal{K}(\delta(c, r))$.

A chaque cellule c de la carte, est associée un vecteur référent $\mathbf{w_c} = (\mathbf{w_c^r}, \mathbf{w_c^b})$ de dimension d où $\mathbf{w}_c^r \in R^n$ et $\mathbf{w}_c^b \in \beta^m$. Par la suite nous notons $\mathcal{W}$ l'ensemble des vecteurs référents constitués par les parties numériques et par la partie binaire.

Dans la section suivante nous présentons un modèle original de cartes topologiques dédiées aux données mixtes avec la prise en compte des deux espaces réel et binaire en définissant un hyper-paramètre pour contrôler les variables quantitatives et qualitatives codées en binaires. L'algorithme d'apprentissage associé est dérivé de l'algorithme Batch de Kohonen dédié aux données numériques (Kohonen (1995)) et de l'algorithme BinBatch dédié aux données binaires (Lebbah et al (2000)). Dans cet algorithme, l'indice de similarité et l'estimation des vecteurs référents sont spécifiques pour chaque partie de la base : c'est la distance euclidienne avec le vecteur moyen pour la partie numérique et la distance de Hamming et le centre médian pour la partie binaire.

2.1 Minimisation de la fonction de coût

Comme dans le cas des cartes topologiques (Kohonen (1995)) nous proposons de minimiser la fonction de coût suivante :

$$\mathcal{E}(\phi, \mathcal{W}) = \sum_{\mathbf{z}_i \in App} \sum_{r \in C} \mathcal{K}(\delta(\phi(\mathbf{z}_i), r)) ||\mathbf{z}_i - \mathbf{w}_r||^2 \quad (1)$$

Où ϕ affecte chaque observation $\mathbf{z}$ à une cellule unique de la carte $\mathcal{C}$.

Dans cette expression $||\mathbf{z} - \mathbf{w}_r||^2$ représente le carré de la distance euclidienne. Etant donné que, pour les données binaires la distance euclidienne n'est rien d'autre que la distance de Hamming $\mathcal{H}$, la distance euclidienne peut être réécrite : $||\mathbf{z}-\mathbf{w}_r||^2 = ||\mathbf{z}^r-\mathbf{w}_r^r||^2+\mathcal{H}(\mathbf{z}^b, \mathbf{w}_r^b)$. Pour contrôler les deux parties des données (réelles et binaires), nous avons utilisé un hyper-paramètre F, qui respecte la propriété $0 \leq F \leq 1$ pour pondérer les variables réelles et qualitatives codées en binaires. Cette pondération permet de pallier le problème d'échelle entre les variables binaires et réelles normalisées entre 0 et 1. Ainsi, la distance est égale à :

$$||\mathbf{z} - \mathbf{w}_r||^2 = (1-F)||\mathbf{z}^r - \mathbf{w}_r^r||^2 + F\mathcal{H}(\mathbf{z}^b, \mathbf{w}_r^b).$$

Utilisant cette expression, la fonction de coût devient :

$$\begin{aligned} \mathcal{E}(\phi, \mathcal{W}) &= (1-F) \sum_{\mathbf{z}_i \in App} \sum_{r \in C} \mathcal{K}(\delta(\phi(\mathbf{z}_i), r)) \mathcal{D}_{euc}(\mathbf{z}_i^r, \mathbf{w}_r^r) \\ &\quad + F \sum_{\mathbf{z}_i \in App} \sum_{r \in C} \mathcal{K}(\delta(\phi(\mathbf{z}_i), r)) \mathcal{H}(\mathbf{z}_i^b, \mathbf{w}_r^b) \end{aligned} \tag{2}$$

La fonction de coût peut être encore réécrite :

$$\mathcal{E}(\phi, \mathcal{W}) = (1-F)\mathcal{E}_{som}(\phi, \mathcal{W}^r) + F\mathcal{E}_{bin}(\phi, \mathcal{W}^b) \tag{3}$$

Où

$$\mathcal{E}_{som}(\phi, \mathcal{W}) = \sum_{\mathbf{z}_i \in App} \sum_{r \in C} \mathcal{K}^T(\delta(\phi(\mathbf{z}_i), r)) ||\mathbf{z}_i^r - \mathbf{w}_r^r||^2 \tag{4}$$

est la fonction de coût classique utilisée par l'algorithme de Kohonen (la version batch), Kohonen (1995).

Et

$$\mathcal{E}_{bin}(\phi, \mathcal{W}) = \sum_{\mathbf{z}_i \in App} \sum_{r \in C} \mathcal{K}^T(\delta(\phi(\mathbf{z}_i), r)) \mathcal{H}(\mathbf{z}_i^b, \mathbf{w}_r^b) \tag{5}$$

est la fonction de coût classique utilisée par l'algorithme BinBatch, Lebbah et al (2000).

Dans le cas particulier où $F \in \{0, 1\}$ la fonction de coût (3) est réduite à la fonction de coût (4) ou (5) utilisées respectivement dans le cas numérique et binaire. Pour les autres valeurs de F les deux parties sont prises en compte avec leurs pondérations. Le choix du paramètre F est déterminé par expérimentation.

Pour un paramètre F fixé, La minimisation de la nouvelle fonction de coût globale (3) est réalisée à l'aide d'une procédure itérative en deux phases :

1. **Phase d'affectation** : mise à jour de la fonction d'affectation ϕ associée à l'ensemble $\mathcal{W}$ fixé. On affecte chaque observation $\mathbf{z}$ au référent défini à partir de l'expression suivante :

$$\forall \mathbf{z},\ \phi(\mathbf{z}) = \arg\min_c((1-F)||\mathbf{z}^r - \mathbf{w}_c^r||^2 + F\mathcal{H}(\mathbf{z}^b, \mathbf{w}_c^b)) \tag{6}$$

2. **Phase de d'optimisation** : La fonction d'affectation étant fixée à sa valeur courante, choisir le système de référents qui minimise la fonction $\mathcal{E}(\phi, \mathcal{W})$ dans l'espace $R^n \times \beta^m$. Ceci nous amène à minimiser la fonction $\mathcal{E}_{som}(\phi, \mathcal{W})$ (4) dans R^n et la fonction $\mathcal{E}_{bin}(\phi, \mathcal{W})$ (5) dans β^m. Ces deux minimisations permettent de définir les expressions nécessaires pour calculer l'ensemble des référents :

   – la partie numérique $\mathbf{w}_c^r$ du vecteur référent $\mathbf{w}_c$ est le vecteur moyen défini comme suit :

$$\mathbf{w}_c = \frac{\sum_{\mathbf{z}_i \in \mathcal{A}} \mathcal{K}(\delta(c, \phi(\mathbf{z}_i)))\mathbf{z}_i^r}{\sum_{\mathbf{z}_i \in \mathcal{A}} \mathcal{K}(\delta(c, \phi(\mathbf{z}_i)))},$$

   – la partie binaire $\mathbf{w}_c^b$ du vecteur référent $\mathbf{w}_c$ est le centre médian de la partie binaire des observations $\mathbf{z}_i \in \mathcal{A}$ pondérées par $\mathcal{K}(\delta(c, \phi(\mathbf{z}_i)))$. Chaque composante $w_c^b = (w_c^{b1}, ..., w_c^{bk}, ..., w_c^m)$ est calculée comme suit :

$$w_c^{kb} = \begin{cases} 0 & \text{si } \left[\sum_{\mathbf{z}_i \in \mathcal{A}} \mathcal{K}(\delta(c, \phi(\mathbf{z}_i)))(1 - \mathbf{z}_i^{bk})\right] \geq \\ & \left[\sum_{\mathbf{z}_i \in \mathcal{A}} \mathcal{K}(\delta(c, \phi(\mathbf{z}_i)))\mathbf{z}_i^{bk}\right] \\ 1 & \text{sinon} \end{cases},$$

La minimisation de la fonction de coût $\mathcal{E}(\phi, \mathcal{W})$ s'effectue par itération successive des deux phases jusqu'à stabilisation ou jusqu'à un nombre d'itérations définies à l'avance. A la fin de l'apprentissage, $\mathbf{w}_c$ partage le même codage que les observations initiales, ce qui permet une interprétation symbolique de la partie binaire du référent. La qualité de la partition résultat de la carte topologique ainsi que l'ordre topologique fourni par la grille, dépend fortement de la fonction voisinage $\mathcal{K}$. Dans la pratique, comme dans le cas des cartes topologiques classiques, nous utilisons une fonction noyau avec un paramètre T pour contrôler la taille du voisinage définie par : $\mathcal{K}^T(\delta(c, r)) = \exp(\frac{-0.5\delta(c,r)}{T})$. Ainsi, par analogie avec l'algorithme de Kohonen, les deux itérations précédentes sont répétées en faisant décroître le paramètre T entre deux valeurs T_{max} et T_{min}.

3 Méthode hybride CT-SVM :Cartes topologiques et SVM

Pour certains problèmes de classement, il est préférable de décomposer le problème global de classement en sous-problèmes pour améliorer les performances en classement, (Platt (1999); Gamma et al (2000); Kuncheva et al (2002); Lebrun et al (2004)). Par exemple, si l'on dispose d'une base de données où certaines observations sont linéairement séparables et les autres sont non linéairement séparables, alors il est possible de décomposer la base entière en deux sous-ensembles et d'entraîner un classifieur SVM pour chacun des sous-ensembles. Ce cas d'utilisation des machines à vecteurs de support dans le cas non linéaire est le plus intéressant car la plupart des problèmes réels sont non linéairement séparables. Il est évident que la détermination du nombre d'observations et par conséquent la taille de la partition utilisée pour l'apprentissage de chaque SVM est important d'un point de vue de la théorie de l'apprentissage, Vapnik (1995). Dans cette section, nous ne présentons pas un indice permettant

d'estimer la taille de la partition, mais nous allons présenter par la suite un modèle de classement qui permet d'augmenter les performances en classement en utilisant le partitionnement des observations.

Dans (Kuncheva , 2004, chapitre 6), l'auteur fournit une démonstration pour ce type de modèle. Si l'on considère que l'on dispose de S classifieurs notés Cla associés à différents sous-ensembles P_i et si on note par $p(Cla_i/P_i)$ la probabilité du classement correct avec le classifieur Cla_i dans le sous-ensemble P_i, alors la densité de probabilité du classement correct de notre système de partitionnement et de classement s'écrit :

$$p(correct) = \sum_{i=1}^{S} p(P_i)p(Cla_i/P_i)$$

Où $p(P_i)$ est la probabilité a priori que l'observation soit générée dans le sous-ensemble P_i. Pour maximiser ce mélange de probabilité, on choisit $p(Cla_i/P_i)$ tel que $p(Cla_i/P_i) \geq p(Cla_j/P_j), j = 1..S$.

Afin de simplifier le problème de classement, notre approche consiste à entraîner des SVMs ($Cla_i = SVM$) différents avec des sous-ensembles d'une partition $\mathcal{P}$ de la base $\mathcal{A}$. Ceci permet de redéfinir des espaces de redescription différents (ou les mêmes) pour chaque sous-ensemble $P_c \in \mathcal{P}$, L'objectif de notre modèle CT-SVM est d'améliorer la discrimination en entraînant un SVM pour chaque sous-ensemble $P_c \in \mathcal{P}$ qui a plus d'une classe (les sous-ensembles non purs §1.1). Pour les sous-ensembles, qui sont composés d'observation de la même classe, aucun un SVM ne sera entraîné. L'algorithme des cartes topologiques mixtes définit au paragraphe (§2) est utilisé pour définir une partition de la base d'apprentissage.

Afin de réduire la partition et par conséquent le nombre de SVMs entraînés, nous avons utilisé la classification hiérarchique (CAH), sur l'ensemble des référents $\mathcal{W}$ de la carte pour réduire la partition ainsi le nombre de sous-ensembles,Yacoub et al (2001); Ripley (1996). Cette phase de réduction de la partition, qui consiste à fusionner certains sous-ensembles, est optionnelle et, elle peut être déterminée en interaction avec les experts et après visualisation des cartes topologiques.

L'algorithme de note modèle CT-SVM est le suivant :

Pour un nombre de sous-ensembles S fixé faire :

- **Phase 1 :** Construction d'une partition $\mathcal{P} = \{P_1, ..., P_{N_{cell}}/$ en utilisant les cartes topologiques avec l'algorithme définit dans la section 2.1.
- **Phase 2 (optionnelle) :** Si $S < N_{cell}$ appliquer la classification hiérarchique (CAH) pour construire la nouvelle partition $\mathcal{P} = \{P_1, ..., P_S/1 \leq S \leq N_{cell}\}$
- **Phase 3 :** Détecter l'ensemble des indices $\mathcal{I}_p$ des sous-ensembles purs tel que $\mathcal{I}_p = \{c/\forall \mathbf{z} \in P_c, \chi(\phi(\mathbf{z})) = c, vote(P_c) = y_c\}$. y_c est l'étiquette du vote majoritaire à 100%

du sous-ensemble P_c.

– **Phase 4 :** Apprentissage du SVM pour chaque sous-ensemble P_i tel que $i \notin \mathcal{I}_p$.

Remarque :
Pour l'apprentissage des cartes topologiques mixtes, nous avons utilisé notre programme développé en C/C++. Nous avons aussi utilisé les programmes et l'heuristique développée par l'équipe de Kohonen, Vesanto et al (2000), pour estimer la dimension de la carte. Pour l'apprentissage du modèle SVM dans le cas du multi-classe, nous avons utilisé le modèle DAG-SVM (Directed Acyclic Graph SVM) développé par Platt (1999); Platt et al (2000); Cawley (2000).

Avec ce modèle CT-SVM, la topologie ou la forme des observations est présentée par les cartes topologiques. Lorsqu'on présente une nouvelle observation qui n'a pas participé à la phase d'apprentissage, elle sera projetée d'abord sur la carte topologique avec la fonction d'affectation associée ϕ (formule 6), puis on utilisera la fonction d'affectation χ (voir §1.1), pour sélectionner le sous-ensemble qui va déterminer le classifieur SVM associé. Cette méthode d'affectation de notre classement permet de comprendre le comportement d'une observation à travers son référent $\mathbf{w}_c$. Si on note par svm_r la fonction de classement du modèle SVM du sous-ensemble P_r alors la fonction d'affectation globale de notre système s'écrit comme suite :

$$y_i = \begin{cases} svm_{\chi(\phi(\mathbf{z}_i))} & \text{si } \chi(\phi(\mathbf{z}_i)) \notin \mathcal{I}_p \\ vote(P_{\chi(\phi(\mathbf{z}_i))}) & \text{sinon} \end{cases} , \tag{7}$$

où $\mathcal{I}_p$ est l'ensemble des indice des sous-ensembles pures. $\chi(c) = c$ si $\mathcal{P} = \{P_1, ..., P_c, ..., P_{N_{cell}}\}$ et $\chi(c) = 1$ si $\mathcal{P} = \mathcal{A}$ (voir §1.1).

4 Expérimentations

Dans la suite, nous avons illustré les performances obtenues par notre modèle en choisissant quelques exemples de la littérature qui ont été traités par les variantes de ce modèle. En outre, une base réelle a été utilisée. Il s'agit d'un extrait d'une base médicale regroupant des données clinico-biologiques portant sur l'étude de l'obésité (Hôpital Hôtel-Dieu, Paris). Cette base va nous servir à illustrer le potentiel des différentes visualisations des cartes topologiques mixtes et à montrer l'intérêt de diviser le problème global de classement pour augmenter les performances en classement. Pour mesurer la robustesse de notre modèle, nous avons calculé les taux de bon classement.

4.1 Base réelle : prédiction de perte de poids chez les obèses

Cet exemple porte sur des données réelles, issues d'une base de données caractérisant 101 patients, massivement obèses (BMI : Body Mass Index$> 40kg/m^2$), recrutés et suivis dans le service de Nutrition de l'Hôtel Dieu dans le cadre d'une chirurgie de l'obésité, (Crookes (2006)). Ces données constituent une base difficile en classement. On retrouve un taux de bon classement faible quelle que soit la méthode utilisée (46.8% avec "Random Forest", 50.9%

avec l'arbre de décision et 55% avec SVM). Cette base permet d'illustrer le potentiel des différentes visualisations des cartes topologiques mixtes et à montrer l'intérêt de diviser le problème global de classement pour augmenter les performances en classement du SVM. La base de données comporte des variables cliniques et biologiques, recueillies avant l'intervention chirurgicale. Les patients sont classés en deux groupes (oui/non) suivant la médiane de perte de poids observée 3 mois et 6 mois après la chirurgie (gastroplastie par anneau ajustable ou bypass gastrique). Si la perte de poids est supérieure à la médiane, le patient est étiqueté "oui". Sinon il est étiqueté par "non". Chaque patient est caractérisé par 37 variables réelles (par exemple, le poids, le BMI, ALAT, ASAT, HDL, CRP...) et 13 variables qualitatives (exemple : diabète oui/non), caractérisant l'obésité et ses aspects cliniques et métaboliques ainsi que ses complications multiples.

Pour étudier le comportement de notre modèle en classement, nous avons procédé par une validation croisée en variant le nombre de sous-ensembles de la partition et par conséquent le nombre d'observations associées à chaque apprentissage d'un SVM. Ainsi, nous avons découpé la base complète en trois sous bases de même taille, B_1, B_2, B_3. On apprend sur deux bases parmi les trois et on teste les performances en classement sur la troisième en utilisant les deux étiquettes de perte de poids (oui/non) à trois mois seulement. Ainsi, en utilisant le modèle CT-SVM (§3), trois cartes topologiques sont construites de dimension 3×4, ce qui fournit une partition de 12 sous-ensembles. Pour montrer l'importance de la taille de la partition, nous avons calculé les performances en classement en variant le nombre de sous-ensembles de 1 à 12. Dans le premier cas, l'application de notre modèle CT-SVM sur une partition avec un seul sous-ensemble est équivalente à entraîner un SVM binaire classique sur toute la base.

La figure 2 montre les trois variations du taux de bon classement des trois bases de test, en fonction du nombre de sous- ensembles de la même partition. Dans le cas où la partition contiendrait un seul sous-ensemble, un seul SVM est entraîné sur toute la base. Ainsi dans ce cas particulier, la fonction d'affectation des cartes topologiques (formule 6), n'influe pas sur la fonction d'affectation globale de notre modèle CT-SVM (formule 7). On observe aussi dans la figure 2, que l'augmentation du nombre de sous-ensembles de la partition permet d'augmenter les performances en classement sur les trois tests. Par contre, on constate aussi que lorsque la taille de la partition est très grande les performances diminuent. La partition contenant plusieurs sous-ensembles permet d'apprendre autant de SVMs que de sous-ensembles. Ainsi, la fonction d'affectation globale de notre modèle (formule 7) utilise d'abord la fonction d'affectation des cartes topologiques ϕ (formule 6) pour choisir le sous-ensemble, ainsi le SVM associé avec sa fonction d'affectation svm.

Avec le premier test, on obtient au maximum 60.6% avec trois sous-ensembles ; avec le deuxième test, on obtient 70.6% avec trois sous-ensembles. Finalement, avec le troisième test, on obtient 55.9 avec quatre sous-ensembles. Dans l'entraînement des SVMs avec notre modèle CT-SVM, nous avons utilisé la même fonction noyau linéaire.

Cette validation croisée avec une variation du nombre de sous-ensembles montre l'intérêt et la difficulté de choisir la bonne partition pour une bonne discrimination. Cette partition est déterminée dans notre cas par expérimentation et visualisation des cartes topologiques. Cette

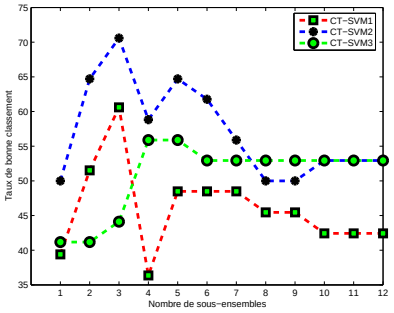


FIG. 2 – *Taux de bon classement avec CT-SVM en fonction du nombre de sous-ensembles. 1 : base d'apprentissage : B_1 et B_2, base de test : B_3 ; 2 : base d'apprentissage : B_1 et B_3, base de test : B_2 ; 3 : base d'apprentissage : B_2 et B_3, base de test : B_1.*

validation croisée montre aussi l'intérêt de subdiviser le problème de classement global en sous-problèmes de classement pour améliorer les performances en classement. Afin de comparer la robustesse en classement de notre modèle CT-SVM aux modèles classiques, nous allons présenter par la suite dans l'exemple 2 trois bases de données fréquemment utilisées dans les expérimentations.

4.1.1 Discussion

Puisque notre modèle utilise les cartes topologiques, on dispose d'un pouvoir de visualisation de la partition. L'application d'abord des cartes topologiques mixtes, va nous permettre d'analyser la répartition des observations et par conséquent les sous-ensembles qui ont servi au classement. L'apprentissage d'une carte de dimension 3×4 cellules effectué sur la base entière des patients, avec l'hyper-paramètre $F = 0.01$, fournie pour chaque cellule un référent $\mathbf{w}_c$ composé de deux parties : la partie quantitative $\mathbf{w}_c^r$ et la partie qualitative $\mathbf{w}_c^b$ codée avec le codage disjonctif binaire.

La figure 3.a présente la répartition des observations. On observe que la partition obtenue a permis de bien distribuer les observations sur 12 cellules de l'ensemble de la partition $\mathcal{P} = \{P_1, ..., P_{12}\}$. La figure 3.b présente la même répartition en distinguant ceux qui ont perdu ou non du poids à 3 mois par rapport à la médiane de l'ensemble des patients. La figure 3.c présente la même répartition de perte de poids à 6 mois. On constate que les sous-ensembles sont mélangés.

A l'aide de cette carte topologique 3×4, il est possible d'effectuer un certain nombre d'analyses de la base étudiée. Notre premier objectif est celui de partitionner les données, en prenant en compte leurs spécificités (données mixtes) pour augmenter les performances en classement. En plus du classement, il est possible d'utiliser le pouvoir de visualisation des cartes topologiques. Pour visualiser la carte topologique, nous nous sommes limités à analyser les effets dus à quelques variables pour lesquels l'exactitude des propriétés médicales retrouvées peuvent

êtres vérifiées. En regardant à la fois les trois figures 3.a, 3.b et 3.c le médecin a détecté globalement trois grands groupes. Pour s'approcher de la partition du médecin, nous avons appliqué la CAH avec les référents de la carte pour avoir 4 sous-ensembles, $\mathcal{P} = \{P_1, P_2, P_3, P_4\}$. La figure 8 présente la partition avec 4 sous-ensembles numérotés de 1 à 4. Cette répartition des données en quatre sous-ensembles et la répartition du médecin en trois sous-ensembles correspondent à la taille de la partition utilisée dans la phase de la validation croisée décrite ci-dessous. En visualisant à la fois les figures 3,4, 5, 6, 7 et la figure 8, il est possible de demander au médecin de définir des profils de patients. Ces profils vont servir à décrire les paramètres (variables) liés à la perte de poids et fournir des hypothèses de travail sur la résistance à la perte de poids fourni par le classifieur.

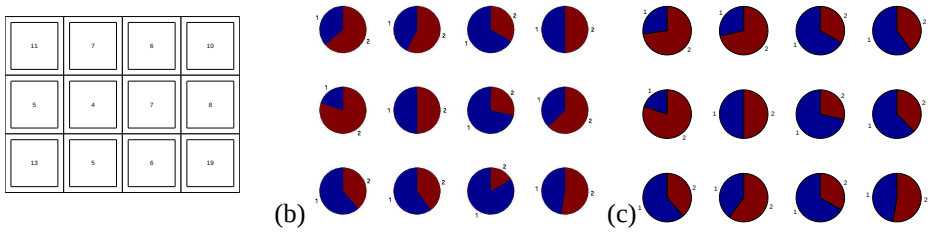
Trois grands profils de patients sont définis selon la cinétique de perte de poids à 3 mois et à 6 mois. Le profil 1 est plutôt un bon profil par rapport aux pertes de poids à trois mois (figure 3.b) et 6 mois (figure 3.c) et correspond aux deux sous-ensembles P_1 et P_2 de la CAH. Le profil 2 est caractérisé par une perte de poids moyenne à 3 mois et 6 mois et correspond approximativement au sous ensemble P_4 de la CAH. Enfin, le profil 3 est caractérisé par une perte de poids médiocre à 3 mois dont l'amplitude diminue à 6 mois, ce qui aboutit à dénommer ce profil comme un "mauvais" profil en terme de perte de poids. Ce profil correspond au sous-ensemble P_3 de la CAH. Nous détaillons par la suite les deux profils 1 et 3 par rapport aux différentes variables clinico-biologiques.

Le profil 1 est caractérisé par un poids, un BMI (Body Mass Index) et une Dépense Energétique de Repos mesurée par calorimétrie (DERm) élevés. Les patients appartenant à ce profil ont une glycémie à jeun et insulinémie élevées (figure 4) sans être diabétiques (figure 5). Il s'agit donc de patients insulinorésistants avant le stade de diabète. Le reste du profil métabolique est caractérisé par des HDL plutôt bas, des triglycérides (TG) et enzymes hépatiques (ASAT, ALAT et GGT) élevés,(figure 4). Dans les classes qualitatives "HTA" (hypertension) ou "SAS" (Syndrome d'apnées du sommeil) ces patients sont classés "oui" (figures 6 et 7). D'un point de vue inflammatoire, la CRP, la férritinémie (FERR), la SAA et l'orosomucoide (ORO), toutes des protéines de la phase aiguë de l'inflammation, sont modérément élevées. Sur le plan nutritionnel, la TSH est basse, le profil protéique (albumine, préalbumine, RBP) et vitaminique est favorable, sans déficit. En conclusion pour ce profil, il s'agit de patients avec un poids très élevé, mais dont le profil métabolique (figure 4) n'est pas trop évolué (sans diabète), sans inflammation importante et un bon profil nutritionnel.

Le profil 2 correspond à des patients ayant un BMI élevé et une leptine élevée (LEP, figure 4). Ils sont insulinorésistants, mais pas diabétiques. Ils ont majoritairement une HTA et un SAS (figures 6 et 7). Les paramètres hépatiques et métaboliques sont normaux. L'adiponectinémie (ADIPO) est plutôt basse. En revanche, les paramètres inflammatoires (SAA et CRP) sont très élevés. Sur le plan nutritionnel, la TSH est normale haute et les marqueurs nutritionnels sont bas (bilan protéique avec albumine, préalbumine et RBP, fer, vitamines A, E, B1, B12). Le profil 3 est un profil intermédiaire en terme de paramètres clinico-biologiques.

En conclusion les deux profils de patients 1 et 2 sont caractérisés par des paramètres clinico-biologiques différents, notamment en terme de marqueurs d'inflammation et nutritionnels et

sont aussi différents en termes de profil de perte de poids à 3 mois et 6 mois. Nous pouvons donc formuler l'hypothèse que le statut nutritionnel et l'état d'inflammation des patients avant chirurgie pourraient être des éléments liés à la résistance à la perte de poids.



(a) (b) (c)

FIG. 3 – *Cartes topologiques* 3×4 *(*$\mathcal{P} = \{P_1, P_2, ..., P_{12}\}$*). (a) Cardinalité des sous-ensembles (b) et (C) Répartition des pertes de poids respectivement à 3 mois et à 6 mois. . 1 : Pas de perte de poids ; 2 : perte de poids.*

4.2 Bases issues de la littérature

Dans cet exemple, trois bases d'apprentissage comportant un nombre variable d'observations ont été utilisées, (table 1) : Iris, Glass, Letter (Blake et al (1998)). Ces bases d'apprentissage et de test sont identiques à ceux pris dans l'article Benabdeslem (2006). Ceci va nous permettre de comparer nos résultats aux méthodes présentés dans l'article de Benabdeslem (2006).

nom/base	#Apprentissage	#Test	#classe	#variables
$Iris$	100	50	3	4
$Glass$	142	72	6	9
$Letter$	10000	5000	26	16

TAB. 1 – *Base d'apprentissage et de test.*

Puisque toutes les variables sont quantitatives, l'utilisation des cartes topologiques mixtes se réduit pour ces bases à l'application de cet algorithme avec l'hyper-paramètre $F = 0$ qui correspond à la version batch des cartes topologiques classiques de Kohonen. Afin de comprendre le déroulement de notre modèle CT-SVM, l'application sur l'exemple des Iris sera détaillée par la suite.

L'apprentissage d'une carte avec 4×3, avec la base d'apprentissage d'Iris, permet d'observer sur la figure 9 une partition $\mathcal{P} = \{P_1, P_2, ...P_{12}\}$. Après application du vote majoritaire sur chaque cellule, on observe que la carte est constituée de trois sous-ensembles. La partie en haut de la carte est majoritairement dédiée à la classe 1, la partie centrale de la carte est dédiée à la classe 2, et le reste est majoritairement de la classe 3 mais mélangé à la classe 2.

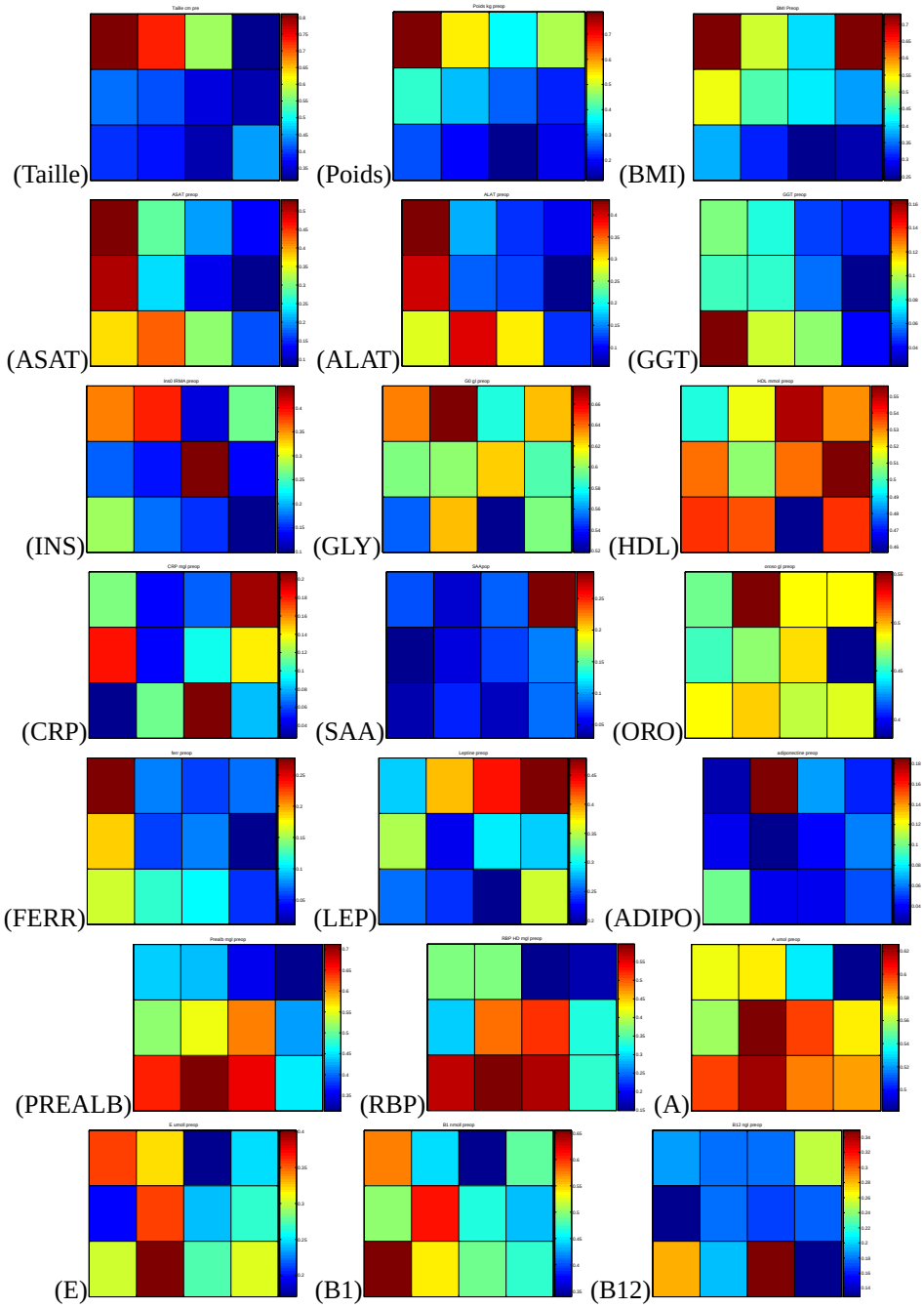


FIG. 4 – *Cartes topologiques décrivant la variation sur les variables Taille,poids,BMI (Body Mass Index), ALAT, ASAT,GGT, INS(insuline), GLY (glycémie),HDL, CRP,SAA,ORO (orosomucoide),FERR (férritinémie),LEP (leptine),ADIPO (adiponectinémie),PREALB (préalbumine),RBP, A, E, B1, B12.*

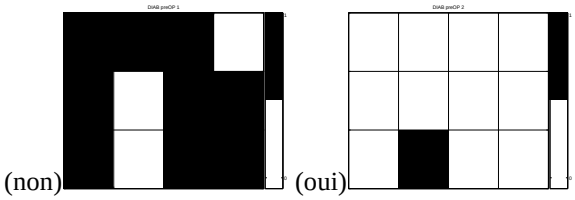


FIG. 5 – *Cartes topologiques représentant les deux modalités non et oui de la variable qualitative Diabète.1 et 0 représente respectivement la présence ou l'absence de la modalité.*

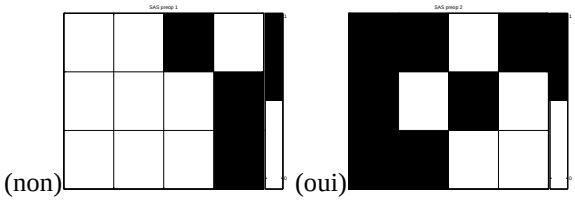


FIG. 6 – *Cartes topologiques représentant les deux modalités non et oui de la variable qualitative SAS. 1 et 0 représentent respectivement la présence ou l'absence de la modalité.*

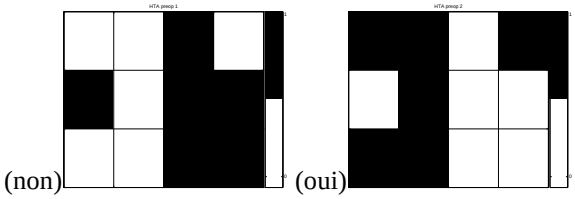


FIG. 7 – *Cartes topologiques représentant les deux modalités non et oui de la variable qualitative HTA. 1 et 0 représentent respectivement la présence ou l'absence de la modalité.*

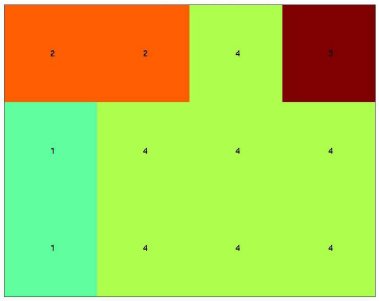


FIG. 8 – *Carte topologique* 3×4 *après le partitionnement de la CAH.* $\mathcal{P} = \{P_1, P_2, P_3, P_4\}$.

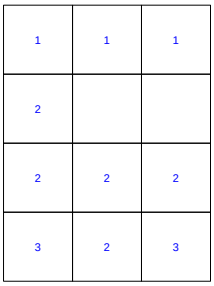


FIG. 9 – *Carte topologique* 4×3 *étiquetée sur la base Iris. Les cellules non numérotées représentent des cellules vides. Les numéro 1, 2,3 représente l'étiquette du vote majoritaire de chaque cellule. La partition* $\mathcal{P} = \{P_1, P_2, ...P_{12}\}$.

Cette visualisation des données avec les cartes topologiques nous amène à appliquer la CAH (classification hiérarchique) sur l'ensemble des référents $\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{12}\}$ pour avoir trois grands sous-ensembles ($S = 3$). Ceci permet de définir la nouvelle partition $\mathcal{P} = \{P_1, P_2, P_3\}$. A partir de cette partition, nous avons calculé la table de confusion 2, qui permet de voir la répartition des classes dans chacun des sous-ensembles. Il est tout à fait possible pour cette base de chercher le nombre de sous-ensembles qui permet d'augmenter les performances en classement, comme nous l'avons fait pour la validation croisée avec la base d'obésité (§4.1). On constate avec cet exemple que la visualisation de la carte topologique fournit une indication du nombre de sous-ensembles.

Partition/Classe	1	2	3
P_1	0	4	0
P_2	0	34	33
P_3	37	0	0

TAB. 2 – *Table de confusion de la partition* $\mathcal{P} = \{P_1, P_2, P_3\}$, P_1 *est majoritairement de la classe 3.* P_3 *est majoritairement de la classe 1.* $\mathcal{I}_p = \{1, 3\}$.

La table 2 montre que la partition $\mathcal{P}$ contient deux sous-ensembles purs P_1 et P_3 qui sont respectivement de la classe 3 et 1 ($\mathcal{I}_p = \{1, 3\}$). A partir de cette table, on conclu que notre partition nécessite l'entraînement d'un seul SVM binaire pour le sous-ensemble P_2 avec les deux classes étiquetées 2 et 3. Notre partitionnement par les cartes topologiques nous a permis d'avoir une projection des données en deux dimensions et de simplifier le SVM en passant du SVM multi-classe, sur toute la base d'apprentissage, à un seul SVM binaire avec un sous-ensemble de taille réduite ($\#P_2 = 52$ observations). Ceci permet un gain de temps et une simplification du SVM. Pour ce sous-ensemble nous avons entraîné un SVM avec une fonc-

tion noyau de type RBF (*Radial Basic Function*).

Le même phénomène, de réduction du nombre de classe, a été observé sur la base Glass. L'apprentissage a été réalisé avec une carte topologique de dimension 9×7, puis le regroupement avec la CAH pour avoir une partition $\mathcal{P}$ constituée de deux sous-ensembles P_1 et P_2. Le premier sous-ensemble nécessite l'apprentissage d'un SVM multi-classe avec 5 classes et le deuxième un SVM multi-classe avec 6 classes. Dans les deux cas, la taille de la base d'apprentissage est réduite. Pour l'apprentissage des deux SVM, nous avons utilisé une fonction noyau de type RBF. Pour nos trois exemples, nous nous sommes basés pour trouver les hyperparamètres du SVM sur les travaux de Hsu et al (2001).

Afin de mesurer la robustesse de notre système, l'apprentissage de notre modèle CT-SVM est réalisé sur les bases d'apprentissage présentées dans la table 1. L'affectation des observations de la base de test est réalisée à l'aide de la fonction d'affectation de notre modèle CT-SVM, présentée par la formule (7).

Base/modèle	one against one	one against all	MLP	DHSVM	**CT-SVM**
$Iris$	97.3	96.7	92.5	97.6	**97.6**
$Glass$	71.5	71.9	70.3	76.8	**81.9**
$Letter$	97.9	97.9	85.2	**98.0**	95.0

TAB. 3 – *Comparaison des performances en classement avec les algorithmes classiques. SVM one against one, SVM one against all, MLP : Multi-Layer Perceptron, DHSVM :Descendant Hierarchical Support Vector Machine.*

La table 3 indique les performances atteintes avec notre modèle CT-SVM sur les bases de test des trois exemples en rappelant ceux du SVM classiques et l'algorithme DHSVM (Descendant Hierarchical Support Vector Machine). Dans la première base "Iris", le taux de bon classement est équivalent à celui du DHSVM et il est de l'ordre de 97.6%. Avec la deuxième base "Glass" une nette amélioration du taux de bon classement est constatée. On passe d'un taux de 71.5% avec le SVM "one againt one" à 81.9% avec notre modèle CT-SVM. Avec la troisième base, on constate que notre modèle CT-SVM arrive à un taux de 95.0% qui mieux que le MLP qui est de $\%85.2$, mais moins bon que le SVM classique et le DHSVM qui a le meilleur taux de 98.0%.

5 Conclusion

Dans ce papier, nous avons présenté un modèle de classement hybride, associant une méthode de partitionnement et une méthode de classement qui sont respectivement, les cartes topologiques et les SVMs. Ce modèle utilise l'organisation des données fournie par les cartes topologiques mixtes pour subdiviser l'espace des données afin d'apprendre un SVM spécifique pour chaque sous-espace des données. Notre modèle CT-SVM utilise la partition résultat des cartes topologiques, pour associer un SVM à chaque sous-ensemble de la partition avec des hyperparamètres différents si cela est nécessaire. Les expériences effectuées montrent la robustesse

de celui-ci à traiter des bases classiques avec uniquement des données réelles ou des données mixtes. D'autres part, dans le cadre d'une application médicale réelle, nous avons vu que la quantité d'information fournie par ce modèle CT-SVM à travers les cartes topologiques mixtes est très importante et le pouvoir de classement avec les SVMs est très performant. Nous avons aussi constaté, qu'il existe une base *letter* pour laquelle la méthode de classement DHSVM et SVM, sont meilleurs, Wolpert et al (1997); Benabdeslem (2006). Ceci nous conduit à réfléchir sur un indice permettant d'estimer la capacité de notre approche à traiter les problèmes de classement. L'autre amélioration qui peut être apporté est d'estimer le nombre de sous-ensembles et par conséquent la partition idéale pour une bonne discrimination des données, Vesanto et al (2000).

Références

Benabdeslem, K. (2006). Descendant hierarchical support vector machine for multi-class problems. International joint conference on neural network (IJCNN 2006) Vancouver .

Ben-Hur, A., D. Horn, H.T. Siegelmann, and V. Vapnik (2001). Support vector clustering, Journal of Machine Learning Research, vol. 2, pp. 125.

Bishop, C. M., M. Svensen and C.K.I. Williams (1998). GTM : The Generative Topographic Mapping. Neural Computation, 10(1), 215-234.

Blake C.L and C.J. Merz (1998). "UCI repository of machine learning databases". Technical report. University of California, Department of information and Computer science, Irvine, CA. available at : *ftp ://ftp.ics.uci.edu/pub/machine-learning-databases*.

Clancey, J.W. (1985). Heuristic Classification. Artificial Intelligence, 27 :p.289-350.

Cawley, G. C, (2000). MATLAB Support Vector Machine Toolbox ($v0.55\beta$) *http ://theoval.sys.uea.ac.uk/gcc/svm/toolbox*, University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ.

Crookes, P.F. (2006). Surgical treatment of morbid obesity. Vol. 57 : 243-264. annurev.med.56.062904.144928.

Egmont-Petersen, M., W. R. M. Dassen, and J. H. C. Reiber (1999). Sequential selection of discrete features for neural networks-A Bayesian approach to building a cascade. Pattern Recognition Letters, 20(11-13) :1439-1448.

Gamma, J. and P. Brazdil (2000) Cascade generalization. Machine Learning, 41(3) :315-343.

Golub, T.R., D. K. Slonim, P. Tamayo, C. H. M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander (1999). Molecular classification of cancer : class discovery and class prediction by gene expression monitoring. In Science, volume 286, p. 531-537.

Hsu, C-W. and C-J. Lin (2001). A comparison of methods for multi-class support vector machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 19.

Hyun-Chul. K, P. Shaoning, J. Hong-Mo, K. Daijin, Y.B. Sung (2003). Constructing support vector machine ensemble. Pattern Recognition vol. 36, no. 12, pp. 2757-2767.

Kohonen, T. (1995). Self-Organizing Map. Springer, third edition Berlin.

Kuncheva, L. I. (2004) Combining Pattern Classifiers, Methods and Algorithms. A Wiley-Interscience publication. ISBN 0-471-21078-1.

Kuncheva, L. I. (2002). Switching between selection and fusion in combining classifiers : An experiment. IEEE Transactions on Systems, Man, and Cybernetics, 32(2) :146-156.

Lebbah, M., Thiria. S, Badran. ESANN, Topological Map for Binary Data, ESANN 2000, Bruges, April 26-27-28, 2000, Proceedings.

Lebbah, M., A. Chazottes, S. Thiria and F. Badran. ESANN (2005) Mixed Topological Map, ESANN 2005, Bruges, April 26-27-28, Proceedings.

Lebrun, G., C. Charrier, O. Lezoray, H. Cardot (2004). Réduction du temps d'apprentissage des SVM par Quantification Vectorielle . CORESA (COmpression et REpréesentation des signaux Audiovisuels), pp 223-226.

Liu, R. and B. Yuan.(2001). Multiple classifier combination by clustering and selection. Information Fusion, 2 :163-168.

Peng, H., F. Long, C. Ding (2005). Feature Selection Based on Mutual Information : Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence, v.27 n.8, p.1226-1238.

Platt, J., N. Cristianini, J. Shawe-Taylor (2000) "Large Margin DAGs for Multiclass Classification", in Advances in Neural Information Processing Systems 12, pp. 547-553, MIT Press.

Platt, J. C. (1999). "Fast training of support vector machines using sequential minimal optimization", in Advances in Kernel Methods - Support Vector Learning, (Eds) B. Scholkopf, C. Burges, and A. J. Smola, MIT Press, Cambridge, Massachusetts, chapter 12, pp 185-208.

Philip, M., L. Vinsensius B. Vega. (2003). Boosting and Microarray Data, Machine Learning, v.52 n.1-2, p.31-44.

Ripley, B.D. Pattern Recognition and Neural networks. Cambridge University Press, Cambridge, 1996.

Rybnik, M., A. Chebira, K. Madani (2003). Auto-adaptive Neural Network Tree Structure Based on Complexity Estimator. IWANN (1) : 558-565.

Shaoning, P., D. Kim, S.Y. Bang (2005). Face Membership Authentication Using SVM Classification Tree Generated by Membershipbased LLE Data Partition, IEEE Trans. on Neural Network, 16(2) 436-446.

Sungmoon, C., Sang Hoon Oh Soo-Young Lee (2004). Support Vector Machines with Binary Tree Architecture. Neural Information Processing. Letters and Reviews Vol. 2, No. 3.

Tamayo, P., D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, T. R. Golub (1999). Interpreting patterns of gene expression with self-organizing maps : Methods and application to hematopoietic differentiation, Proc. National Academy Science of USA 96, p. 2907-2912.

Vapnik, V.N. (1995). "The Nature of Statistical Learning Theory", Springer-Verlag, New York, ISBN 0-387-94559-8.

Vesanto, J., J. Himberg, E. Alhoniemi and J. Parhankangas (2000). "SOM Toolbox-Team". Helsinki University of Technology. P.O.Box 5400, FIN-02015 HUT, FINLAND. http ://www.cis.hut.fi/projects/somtoolbox/.

Vesanto, J., Alhoniemi, E.(2000). "Clustering of the Self-Organizing Map", IEEE Transactions on Neural Networks.

Weston, J., J. Cai and W.N. Grundy (2001). Gene functional classification from heterogeneous data Paul Pavlidis. Proceedings of RECOMB.

Wolpert, D.H., and W.G. Macready (1997). No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation.

Wu, W., X. Liu, M. Xu, J. Peng, R. Setiono (2004) A Hybrid SOM-SVM Method for Analyzing Zebra Fish Gene Expression. 17th ICPR'04 - Volume 2 pp. 323-326.

Xing, E.P., M. I. Jordan, R.M. Karp (2001). Feature selection for high-dimensional genomic microarray data, Proceedings of the Eighteenth International Conference on Machine Learning, p.601-608, June 28-July.

Yacoub, M., F. Badran and S. Thiria (2001). A Topological Hierarchical Clustering : Application to Ocean Color Classification ICANN proceedings.

Summary

This paper introduces a classification model combining mixed topological map and support vector machines. The non supervised model is dedicated for clustering and visualizing mixed data. The supervised model is dedicated to classification task. In the present paper, we propose a combination of two models performing a data visualization and classification. The task of our model is to train topological map in order to cluster data set on organized subset. For each subset, we propose to train a SVM model. The global classification problem is devided into classification sub problem corresponding to the number of subset. The model is validated on forth data bases. The first one is related to the obesity problem, which is provied by Nutrition team located in hospital Hôtel-Dieu in Paris. The others are taken from public data set repository.