

OSOM : un algorithme de construction de cartes topologiques recouvrantes

Guillaume Cleuziou

Laboratoire d'Informatique Fondamentale d'Orléans (LIFO)

Université d'Orléans

Rue Léonard de Vinci - 45067 Orléans Cedex 2

Guillaume.Cleuziou@univ-orleans.fr

Résumé. Les modèles de classification recouvrante ont montré leur capacité à générer une organisation plus fidèle aux données tout en conservant la simplification attendue par une structuration en classes strictes. Par ailleurs les modèles neuronaux non-supervisés sont plébiscités lorsqu'il s'agit de visualiser la structure de classes.

Nous proposons dans cette étude d'étendre les cartes auto-organisatrices traditionnelles aux cartes auto-organisatrices recouvrantes. Nous montrons que cette nouvelle structure apporte des solutions à certaines problématiques spécifiques en classification recouvrante (nombre de classes, complexité, cohérence des recouvrements).

L'algorithme OSOM s'inspire de la version recouvrante des nuées dynamiques et de l'approche de Kohonen pour générer de telles cartes recouvrantes. Nous discutons du modèle proposé d'un point de vue théorique (fonction d'énergie associée, complexité, ...). Enfin nous présentons un cadre d'évaluation générale que nous utilisons pour valider les résultats obtenus sur des données réelles.

1 Introduction

La problématique de la classification recouvrante s'inscrit dans le processus général d'extraction de connaissances à partir de données ; elle consiste à faire émerger une organisation synthétique d'un ensemble d'individus à l'aide d'une structure de classes (hiérarchique ou non) dans laquelle chaque individu peut appartenir à plusieurs classes. Souvent plus riches et plus adaptés que leurs analogues non recouvrants, les modèles recouvrants accompagnent les avancées réalisées dans le domaine de la classification non-supervisée en général ; à titre d'exemples on notera les adaptations suivantes : les pyramides (Diday, 1984) généralisent les hiérarchies, OKM (Cleuziou, 2008) généralise l'algorithme des k -moyennes et MOC (Banerjee et al., 2005) correspond à une variante recouvrante des modèles de mélanges.

Outre les problématiques récurrentes inhérentes aux méthodes de classification (paramétrage, évaluation, passage à l'échelle, etc.) la classification recouvrante présente des problématiques propres telles que :

- **le choix du nombre de classes** : dans le cas des modèles recouvrants non hiérarchiques, l'ampleur des recouvrements entre classes est une caractéristique incontournable voire

déterminante sur la décision du nombre approprié de classes ; cependant il reste difficile de donner une préférence *a priori* entre une organisation constituée de peu de classes avec de forts recouvrements ou une organisation avec d'avantage de classes moins recouvrantes. On peut illustrer cette problématique de manière pratique en Recherche d'Information en se demandant si un sous-ensemble de textes qui traitent deux thématiques constitue à lui seul une nouvelle classe ou correspond à l'intersection des deux classes thématiques associées.

- **la cohérence topologique des recouvrements** : ce que nous appelons ici "cohérence topologique" correspond au fait que des classes ne puissent se recouvrir que si elles sont proches au sens de la topologie induite par la métrique utilisée. Celle-ci est presque toujours admise dans le processus même de construction de la structure classificatoire : les modèles hiérarchiques recouvrants limitent de fait les chevauchements à des classes voisines qui correspondent à des zones¹ adjacentes sur l'espace de projection des individus, induit par la classification ; dans le cas des méthodes de types réallocations dynamiques (Cleuziou, 2008; Bezdek, 1981), les affectations multiples d'un individu sont souvent réalisées parmi les classes les plus proches de l'individu en question, ce dernier assurant ainsi une continuité topologique entre les classes qu'il permet de se faire recouvrir.
- **le coût induit par l'élargissement de l'espace des possibilités** : les hiérarchies (resp. partitions) ne sont que des cas particuliers de pyramides (resp. recouvrements) ; l'espace des solutions possibles étant fortement élargi lorsque l'on considère des structures classificatoires recouvrantes, les algorithmes d'exploration peuvent dans certains cas s'avérer plus complexes. Si la complexité théorique de la variante recouvrante (OKM) des k -moyennes reste linéaire sur le nombre d'individus, l'algorithme de construction d'une pyramide est de complexité polynomiale d'ordre au moins trois alors que son analogue non recouvrant est au plus quadratique.

Le travail que nous proposons dans cet article vise à répondre aux problématiques spécifiques évoquées ci-dessus par l'utilisation du concept des cartes auto-organisatrices (Kohonen, 1984), adapté à la problématique de la classification recouvrante. En effet, la structure classificatoire proposée par ce type de carte permettra :

- de ne plus décider à l'avance du nombre de classes souhaité mais de laisser le processus organiser les individus sur une carte dont le nombre de neurones est généralement très supérieur au nombre de classes finales² ;
- d'assurer de manière simple la cohérence topologique des recouvrements par le biais d'une affectation des individus à plusieurs neurones vérifiant une structure particulière de sous-graphe sur la carte (e.g. cliques) ;
- une pré-organisation des individus dans un formalisme facilitant un processus ultérieur de classification hiérarchique recouvrante, et ainsi de réduire le coût de traitement de ce type de modèles recouvrants.

Par ailleurs, les recouvrements permettant à chaque individu d'être associé à plusieurs neurones sur la carte, l'existence ou non de chevauchements entre neurones peut être facilement visualisé et apporte une information capitale sur l'interprétation visuelle (et éventuellement la segmentation) de la carte topologique résultante.

¹Ces zones peuvent être des intervalles dans le cas de hiérarchies 2D ou des surfaces pour des hiérarchies 3D, etc.

²Le choix de la taille de la carte influe sur la qualité du résultat mais de manière moins déterminante que la décision sur le paramètre k dans un algorithme de type k -moyennes.

L'article s'articule en trois sections majeures, suivies d'une conclusion débouchant sur les perspectives de cette étude. La Section 2 présente le formalisme de notation utilisé et des rappels à la fois sur les méthodes de classification recouvrante et sur les cartes auto-organisatrices. Nous détaillons ensuite (Section 3) l'algorithme OSOM avant d'en proposer une validation expérimentale sur des jeux de données réels multi-classes (Section 4).

2 Cadre de l'étude

Dans la suite on notera $X = \{x_1, \dots, x_N\}$ l'ensemble des données à traiter, décrites dans \mathbb{R}^p muni de la métrique euclidienne $\|\cdot\|$. Ces données sont à organiser en K classes (K paramètre des modèles) notées $\mathcal{C} = \{C_k\}_{k=1}^K$ et dont les représentants seront notés $\{w_k \in \mathbb{R}^p\}_{k=1}^K$. \mathcal{A}_i désignera l'ensemble des classes auxquelles x_i est affecté $\{C_k | x_i \in C_k\}$.

2.1 Variante recouvrante des k -moyennes

Nous nous inspirons de l'approche de Cleuziou (2008) qui présente une variante recouvrante des k -moyennes (MacQueen, 1967). L'algorithme OKM est une généralisation des k -moyennes aussi bien dans le modèle (fonction objective) que dans l'algorithme (processus d'optimisation). Un critère des moindres carrés généralisé est alors défini par

$$E_{OKM} = \sum_{x_i \in X} \|x_i - \phi(x_i)\|^2 \quad (1)$$

où $\phi(x_i)$ est définie comme combinaison (généralement linéaire) des représentants des classes de x_i . Cleuziou (2008) montrera que l'algorithme MOC proposé par Banerjee et al. (2005) comme variante recouvrante de EM (Dempster et al., 1977) est également basé sur la minimisation du critère des moindres carrés généralisé. De par les restrictions imposées à MOC, les deux modèles ne diffèrent finalement qu'à travers la fonction de combinaison $\phi(x_i)$: celle-ci est donnée dans OKM par le barycentre des représentants ; tandis que la combinaison retenue dans MOC est une simple somme :

$$\phi_{OKM}(x_i) = \frac{1}{|\mathcal{A}_i|} \sum_{w_k \in \mathcal{A}_i} w_k \quad ; \quad \phi_{MOC}(x_i) = \sum_{w_k \in \mathcal{A}_i} w_k \quad (2)$$

De manière générale, les algorithmes associés aux modèles MOC et OKM peuvent être interprétés comme des méthodes de classification d'un ensemble d'individus X sur l'ensemble des 2^K parties de \mathcal{C} plutôt que sur l'ensemble \mathcal{C} lui-même, avec une contrainte sur le nombre de paramètres, restreint à K ($\{w_k\}_{k=1}^K$).

2.2 SOM : Self Organizing Maps

2.2.1 L'algorithme SOM

L'algorithme SOM proposé par Kohonen (1984) a été le premier algorithme de classification topologique. L'approche de Kohonen repose sur un réseau de neurones constitué de K cellules (ou neurones) ; à chaque neurone k est associé :

- une position z_k sur une grille (couche de sortie du réseau),

- un vecteur de poids w_k de même dimension que les données (ici \mathbb{R}^p) représentant le centre du cluster C_k associé.

Une topologie est définie sur l'ensemble des neurones W , par le biais d'une matrice de voisinage (ou matrice d'interactions latérales) H où chaque élément $h_{kl} \in \mathbb{R}^+$ ne dépend ni de W ni de X ; en général h_{kl} est une fonction décroissante de la distance sur la grille (notée $d(z_k, z_l)$) entre les deux neurones k et l . Nous considérerons dans la suite une fonction de voisinage gaussien de la forme

$$h_{kl}^t = \exp\left(-\frac{d(z_k, z_l)}{2\sigma_t^2}\right) \quad (3)$$

dont le paramètre de variance σ_t décroît à mesure des itérations de l'algorithme. Ce dernier procède par itération de deux étapes. Pour un exemple $x_i \in X$ choisi à l'itération t :

1. rechercher le neurone gagnant $g(W, x_i)$ correspondant au neurone le plus proche

$$g(W, x_i) = \arg \min_k \|x_i - w_k\|^2 \quad (4)$$

2. mettre à jour l'ensemble des vecteurs de poids par la règle suivante

$$\Delta w_k = \epsilon_t \cdot h_{g(W, x_i), k}^t \cdot (x_i - w_k) \quad (5)$$

avec ϵ_t un paramètre d'apprentissage décroissant avec t .

Le règle de mise à jour (5) permet, à chaque présentation d'un exemple x_i , de réduire les distances entre, d'une part l'exemple x_i et le représentant $w_{g(W, x_i)}$ du neurone gagnant, et d'autre part entre l'exemple x_i et les représentants des voisins³ du neurone gagnant ; ce faisant les neurones voisins sur la grille voient leurs représentants se rapprocher et ainsi satisfaire une contrainte de similarité entre neurones proches.

2.2.2 Variante de Heskes

Heskes (1999) montre que l'algorithme SOM original ne peut être exprimé comme une méthode d'optimisation d'une fonction objective (ou fonction d'énergie). Il propose alors une variante de SOM en modifiant la définition du neurone gagnant. En choisissant comme neurone gagnant celui qui minimise l'erreur locale suivante

$$g(W, x_i) = \arg \min_k \sum_l h_{kl} \|x_i - w_l\|^2 \quad (6)$$

plutôt que la distance Euclidienne (4), il montre que la règle de mise à jour utilisée dans SOM permet d'assurer la décroissance de la fonction d'énergie

$$E_{Heskes} = \frac{1}{N} \sum_{x_i \in X} \sum_k h_{k, g(W, x_i)} \|x_i - w_k\|^2 \quad (7)$$

On note que la modification apportée dans le choix du neurone gagnant ne remet pas en cause l'idée de base consistant à choisir le neurone le plus "proche" de l'exemple x_i . La "proximité" entre un exemple x_i et un neurone k peut être définie à partir du seul représentant w_k (4) ou en tenant compte du voisinage de k (6).

³Au sens de la topologie induite par H .

Par ailleurs, Lesot et al. (2003) montrent que si H correspond à un voisinage gaussien avec $\forall k, h_{kk} = 1$, la fonction d'énergie E_{Heskes} se décompose en une somme $E_{Heskes} = E_1 + E_2$

$$E_1 = \frac{1}{N} \sum_{x_i \in X} \|x_i - w_{g(W, x_i)}\|^2 \quad \text{et} \quad E_2 = \frac{1}{N} \sum_{x_i \in X} \sum_{k \neq g(W, x_i)} h_{k, g(W, x_i)} \|x_i - w_k\|^2 \quad (8)$$

Cette décomposition permet de faire apparaître d'une part le critère des moindres carrés classique utilisé dans l'algorithme des k -moyennes (E_1) et d'autre part un terme visant à imposer l'organisation des neurones voisins sur la grille (E_2).

Les modèles ainsi que les résultats théoriques revus dans cette section permettront de présenter le nouveau modèle dans un formalisme identique. Nous utiliserons alors les notions de *combinaison de représentants* (utilisée en classification recouvrante) et d'*erreur locale* (utilisée dans la variante de Heskes) afin de proposer un modèle de construction de cartes auto-organisatrices recouvrantes qui soit généraliste et théoriquement fondé.

3 OSOM : modèle et algorithme

3.1 Le modèle OSOM

Le caractère recouvrant d'une classification implique d'autoriser une donnée à appartenir à plusieurs classes. Dans le cas des réseaux de neurones, nous autoriserons chaque exemple présenté à être affecté à plusieurs cellules. Pour ce faire, la présentation d'un exemple x_i donnera lieu à la recherche d'un sous-ensemble de neurones gagnants $G \subset W$. De plus l'existence d'une topologie H sur W offre la possibilité de contrôler la cohérence des recouvrements en limitant les sous-ensembles gagnants potentiels à une classe particulière \mathcal{G} de sous-graphes de la grille. Par exemple, en choisissant l'ensemble des cliques comme classe de sous-graphes, les sous-ensembles de neurones gagnants possibles sont définis par

$$\{G \in \mathcal{P}(W) | \forall (w_k, w_l) \in G \times G, d(z_k, z_l) \leq 1\}$$

avec $\mathcal{P}(W)$ l'ensemble des parties de W et, rappelons-le, z_k et z_l désignent les positions des neurones k et l sur la grille. On notera que la classe des cliques n'est pas le seul choix possible mais reste intuitivement privilégié dans la mesure où il assure que chaque individu n'appartient qu'à des classes proches et offre une facilité de visualisation de la carte.

Par fusion du modèle OKM et de la variante SOM de Heskes, nous proposons le modèle OSOM (*Overlapping Self-Organizing Maps*) basé sur la fonction d'énergie suivante

$$E_{OSOM} = \frac{1}{N} \sum_{x_i \in X} \sum_{G_r \in \mathcal{G}} h_{r, g(\mathcal{G}, x_i)} \|x_i - \bar{w}_r\|^2 \quad (9)$$

On retrouve dans ce modèle la somme sur chaque exemple des erreurs locales, exprimées cette fois non-plus sur l'ensemble des neurones W mais sur l'ensemble des sous-graphes de la classe \mathcal{G} considérée. Dans l'expression (9), G_r désigne un sous-ensemble de neurones (formant un sous-graphe de type \mathcal{G} sur la grille), $h_{r, s}$ est une fonction de voisinage (à définir) sur les éléments de \mathcal{G} , $g(\mathcal{G}, x_i)$ détermine le sous-ensemble de neurones gagnant pour x_i et enfin la notation \bar{w}_r est utilisée ici pour représenter le vecteur de poids associé à G_r .

OSOM : un algorithme de construction de cartes topologiques recouvrantes

Le modèle étant posé nous proposons dans ce qui suit une définition pour \bar{w}_r , g et H dans le cadre recouvrant et décrivons un algorithme de minimisation de la fonction d'énergie E_{OSOM} .

3.2 L'algorithme OSOM

De même que pour OKM le représentant \bar{w}_r d'un ensemble de neurones G_r peut être défini naturellement par combinaison linéaire des vecteurs de poids associés aux neurones qui composent G_r . En choisissant le barycentre comme combinaison on pose alors

$$\bar{w}_r = \frac{1}{|G_r|} \sum_{w_k \in G_r} w_k \quad (10)$$

On notera que si x_i est affecté à l'ensemble G_r , ce dernier joue le rôle de \mathcal{A}_i vis à vis de x_i et \bar{w}_r correspond exactement à la définition de $\phi(x_i)$ donnée en (2).

L'idée directrice de l'approche consiste alors à assigner un exemple x_i à un sous-ensemble de cellules (ou classes) G_r lorsque le barycentre \bar{w}_r associé à G_r est d'avantage représentatif de l'exemple x_i que tout autre vecteur de poids initial w_k et que toute autre combinaison de ces vecteurs.

La fonction qui détermine le sous-ensemble G_r gagnant est donnée par g . À la manière de Heskes nous définissons une erreur locale liée à l'affectation d'un exemple x_i à un ensemble de classes $G_r \in \mathcal{G}$; la plus petite erreur locale identifie la combinaison de classes gagnante :

$$g(\mathcal{G}, x_i) = \arg \min_{\{r | G_r \in \mathcal{G}\}} \sum_{G_s \in \mathcal{G}} h_{rs} \|x_i - \bar{w}_s\|^2 \quad (11)$$

L'erreur locale ainsi définie n'a de sens que si la matrice H de taille $|\mathcal{G}| \times |\mathcal{G}|$ induit une topologie sur \mathcal{G} . Autrement-dit, h_{rs} doit refléter la distance physique des sous-ensembles G_r et G_s sur la grille. Nous conservons une forme de voisinage gaussien, et choisissons comme distance physique entre ensembles, une distance de Hausdorff (Hausdorff, 1962) :

$$h_{rs}^t = \exp\left(-\frac{d_H(G_r, G_s)}{2\sigma_t^2}\right) \quad \text{où} \quad d_H(G_r, G_s) = \max \left\{ \begin{array}{l} \max_{w_k \in G_r} \min_{w_l \in G_s} d(z_k, z_l) \\ \max_{w_l \in G_s} \min_{w_k \in G_r} d(z_k, z_l) \end{array} \right\} \quad (12)$$

On observera ici que si \mathcal{G} se restreint à la classe des singletons W , la distance de Hausdorff correspond à la distance physique d et les fonctions de voisinage (12) et (3) sont identiques.

La définition de l'ensemble de neurones gagnant a été soigneusement choisie de manière analogue à l'approche de Heskes, la fonction d'énergie E_{OSOM} possède alors les mêmes propriétés de dérivabilité que la fonction d'énergie E_{Heskes} . Ainsi on peut montrer que la mise à jour des vecteurs de poids w_1, \dots, w_K qui assure la décroissance de la fonction d'énergie E_{OSOM} est donnée par la règle globale suivante :

$$\Delta w_k = \frac{\epsilon_t}{N} \sum_{x_i \in X} \sum_{G_r \in \mathcal{G}_k} \frac{h_{r,g(\mathcal{G}, x_i)}^t}{|G_r|} \cdot (x_i - \bar{w}_r) \quad (13)$$

Contrairement à la règle habituelle (5), la mise à jour d'un vecteur de poids w_k dans le modèle OSOM fait intervenir non seulement les informations sur le neurone k mais également

celles des neurones inclus dans un sous-ensemble contenant k . Ce sous-ensemble de \mathcal{G} est noté $\mathcal{G}_k = \{G_r \in \mathcal{G} | w_k \in G_r\}$.

La règle (13) correspond à une descente de gradient globale. Dans le cas classique d'une version d'algorithme *on-line* (Figure 1), la mise à jour intervient après chaque présentation d'un exemple x_i par la règle locale

$$\Delta w_k = \epsilon_t \sum_{G_r \in \mathcal{G}_k} \frac{h_{r,g(\mathcal{G},x_i)}^t}{|G_r|} \cdot (x_i - \bar{w}_r) \quad (14)$$

OSOM (*on-line*)

Étant donnés : un ensemble X d'individus dans \mathbb{R}^p , K neurones organisés sur une grille, une classe de sous-graphes \mathcal{G} , un nombre T d'itérations

1. Tirer aléatoirement K individus de X pour initialiser les vecteurs de poids $\{w_k\}$
 2. Générer par (10) les vecteurs de poids \bar{w}_r associés aux sous-ensembles $G_r \in \mathcal{G}$
 3. Pour $t : 1..T$
 4. tirer aléatoirement $x_i \in X$ et rechercher le sous-ensemble gagnant par (11)
 5. mettre à jour chaque vecteur de poids w_k par la règle (14)
 6. recalculer les vecteurs de poids \bar{w}_r des sous-ensembles par (10)
-

FIG. 1 – L'algorithme OSOM.

3.3 Discussion sur la méthode

La variante de Heskes est justifiée par des considérations théoriques de convergence de l'algorithme. En revanche la complexité de l'algorithme s'en trouve considérablement augmentée. Cette observation est également valable pour l'algorithme OSOM qui, de surcroît, élargit l'espace des solutions en autorisant les recouvrements.

Ainsi, chaque itération de l'algorithme OSOM tel que présenté en Figure 1, a une complexité de l'ordre de $O(R^2)$ où R désigne la cardinalité de la classe des sous-graphes \mathcal{G} . Cette complexité, due à l'étape de recherche du sous-ensemble de neurones gagnant (11), pourrait ne pas être acceptable lorsqu'on observe qu'en théorie⁴, la taille R de \mathcal{G} est exponentielle sur le nombre initial K de neurones sur la grille ($R \propto 2^K$). Cependant, l'intérêt d'utiliser le formalisme des cartes topologiques est de pouvoir contrôler la cohérence des recouvrements en limitant \mathcal{G} à une sous-classe particulière de graphes sur la grille.

Par exemple, le fait de choisir comme classe l'ensemble des cliques possibles sur la grille, limite de fait \mathcal{G} à un ensemble de taille proportionnelle à K ; dans le cas d'une grille 2D avec un voisinage carré, on peut montrer que le nombre de cliques R est inférieur à $10.K$, avec K le nombre de cellules.

Par ailleurs il est facile d'envisager une version accélérée de l'algorithme OSOM en procédant de manière analogue à l'algorithme SOM initial proposé par Kohonen (1984). Une extension simple au cas recouvrant pourrait consister lors de l'étape de recherche du sous-ensemble de neurones gagnant à :

⁴*i.e.* sans contraindre la classe \mathcal{G}

OSOM : un algorithme de construction de cartes topologiques recouvrantes

- rechercher un sous-ensemble singleton gagnant en utilisant (4)
- élargir le sous-ensemble à un sous-graphe $G_r \subset \mathcal{G}$ par une heuristique de type glouton.

Nous ne détaillerons pas cette variante accélérée de OSOM faute de place, mais nous l'utiliserons dans la section suivante pour présenter des résultats sur des cartes de tailles 10×10 .

Pour terminer cette discussion, nous revenons sur le modèle OSOM déterminé par la fonction d'énergie (9) et observons la même décomposition que pour la fonction d'énergie de Heskes. En effet, E_{OSOM} se décompose en une somme de deux termes $E_1 + E_2$ avec

$$E_1 = \frac{1}{N} \sum_{x_i \in X} \|x_i - \bar{w}_{g(G, x_i)}\|^2 \quad \text{et} \quad E_2 = \frac{1}{N} \sum_{x_i \in X} \sum_{G_r \neq g(G, x_i)} h_{r, g(G, x_i)} \|x_i - \bar{w}_r\|^2 \quad (15)$$

Cette décomposition fait apparaître dans E_2 un terme visant à imposer l'organisation des sous-ensembles de neurones voisins sur la grille et surtout dans E_1 , le critère des moindres carrés généralisé utilisé dans OKM. De ce point de vue, OSOM peut être considéré comme une généralisation de la variante de Heskes.

4 Évaluation expérimentale

Nous présentons dans cette section quelques jeux de données réelles adaptés à la problématique de la classification recouvrante ainsi qu'un cadre d'évaluation permettant une analyse des cartes recouvrantes selon plusieurs points de vues.

4.1 Présentation des données

La classification recouvrante est adaptée au traitement de données par nature multi-classes. Parmi les domaines traditionnels d'application on peut citer les données textuelles (documents multi-thématiques), images (indexation multiple) et biologiques (gènes multi-fonctions). Nous choisissons alors des bases de tests⁵ parmi ces domaines où non seulement une classification de référence est proposée mais de plus l'organisation correspond à une classification recouvrante. Autrement-dit, une ou plusieurs étiquettes sont associées à chaque donnée. A titre indicatif nous évaluons également l'algorithme sur la base de données Iris (D.J. Newman et Merz, 1998). Les données sont présentées dans le Tableau 1 à travers les informations sur le nombre d'instances, la dimensionalité (attributs), le nombre de classes de référence ainsi qu'une information sur l'importance des recouvrements (nombre moyen d'étiquettes par instance).

Données	Domaine	Instances	Attributs	Classes	Recouv.
Iris	Botanique	150	4	3	1.00
Emotions	musique	593	72	6	1.87
Scene	multimédia	2407	294	6	1.07
Yeast	biologie	2417	103	14	4.24

TAB. 1 – *Profils des jeux de données utilisés.*

⁵Ces bases sont disponible sur le site du projet "Learning from Multi-Label Data" à l'adresse : <http://mlkd.csd.uth.gr/multilabel.html>

Les données *emotions* concernent des extraits musicaux de 30 secondes caractérisés par leurs rythmes et timbres. Des experts ont étiqueté chaque extrait avec une ou plusieurs catégories émotionnelles (happy, sad, calm, surprised, quiet, angry).

Scene est composé d'images décrites par des attributs de type *spatial color moments*. Initialement une seule étiquette était associée à chaque image (ou scène) parmi les classes suivantes : *beach, sunset, fall foliage, field, mountain, urban*. Ces données ont été ré-étiquetées de telle sorte que environ 7% des instances sont multi-catégories.

Enfin, les données *yeast* sont des gènes caractérisés par leur profils d'expression et phylogénétique. Chaque gène est associé à une ou plusieurs classes de fonctions métaboliques.

4.2 Méthode d'évaluation

L'évaluation de classifications est une tâche délicate dans un contexte non-supervisé. Néanmoins un certain nombre de pratiques permettent d'évaluer la qualité des schémas obtenus. Jain et Dubes (1988) différencient deux types de pratiques : les mesures d'évaluation interne ou externe. Dans le premier cas il s'agit de quantifier l'information restituée dans un schéma de classification par rapport à l'information initialement disponible ; l'évaluation externe consiste en revanche à comparer une classification de référence (pré-établie) avec une classification générée par l'algorithme, indépendamment des informations fournies pour la construire.

Nous proposons un modèle général permettant d'évaluer, en fonction de l'instanciation, des cartes topologiques recouvrantes ou non selon différents points de vue. Nous utilisons pour cela la norme de Frobenius sur une différence de matrices ($N \times N$) :

$$Q(A, B) = \sqrt{\frac{1}{N^2} \sum_{i,j} (A_{i,j} - B_{i,j})^2} \quad (16)$$

Nous définissons ensuite quatre matrices contenant chacune une information stratégique :

- la matrice V est telle que $V_{i,j}$ contient une information de dissimilarité entre les labels de classes associés aux exemples x_i et x_j . Dans un cas simple mono-label il suffirait d'affecter 1 à $V_{i,j}$ si x_i et x_j ont la même étiquette et 0 sinon. Dans un cas multi-label nous choisissons d'utiliser l'indice de Jaccard (Jaccard, 1912) sur les listes de labels l_i et l_j associées aux exemples x_i et x_j : $V_{i,j} = 1 - \frac{|l_i \cap l_j|}{|l_i| + |l_j|}$
 - la matrice $U^{(1)}$ (normalisée) contient les distances sur la grille topologique entre les classes associées aux exemples
 - la matrice $U^{(2)}$ (normalisée) contient les distances entre les vecteurs représentant des classes associées aux exemples
 - la matrice $U^{(3)}$ (normalisée) contient les distances initiales entre les individus de X
- $$U_{i,j}^{(1)} = \frac{d_H(G_g(\mathcal{G}, x_i), G_g(\mathcal{G}, x_j))}{Norm.} ; U_{i,j}^{(2)} = \frac{\|\bar{w}_{g(\mathcal{G}, x_i)} - \bar{w}_{g(\mathcal{G}, x_j)}\|^2}{Norm.} ; U_{i,j}^{(3)} = \frac{\|x_i - x_j\|^2}{Norm.}$$

Le Tableau 2 exprime les différentes instanciations possibles du modèle $Q(A, B)$ et le sens donné à chacune des mesures induites.

4.3 Résultats expérimentaux

Le Tableau 3 présente les résultats obtenus sur les 4 jeux de données mentionnés. Pour chaque jeu, l'algorithme OSOM et la variante de Heskes pour SOM sont comparés dans les mêmes conditions (mêmes initialisations). Les valeurs reportées correspondent à des moyennes

OSOM : un algorithme de construction de cartes topologiques recouvrantes

A	B	sémantique de l'évaluation induite	sigle
$U^{(3)}$	V	pertinence des étiquettes vs. distances initiales	Q_{labels}
$U^{(1)}$	V	topologie de la grille vs. labels de classes	Q_{topo}^{ext}
$U^{(2)}$	V	classification obtenue vs. labels de classes	$Q_{classif}^{ext}$
$U^{(1)}$	$U^{(3)}$	topologie de la grille vs. distances initiales	Q_{topo}^{int}
$U^{(2)}$	$U^{(3)}$	classification obtenue vs. distances initiales	$Q_{classif}^{int}$

TAB. 2 – Différentes instanciations possibles du modèle d'évaluation $Q(A, B)$.

sur 10 expériences et chaque expérience comporte 100 itérations (1 itération = présentation de tous les exemples). Les paramètres choisis sont les suivants : voisinages gaussiens tels que présentés dans les modèles, décroissances exponentielles des paramètres ϵ_t et σ_t , grilles de taille $K = 16 (4 \times 4)$ avec une structure de voisinage carré. Enfin pour l'algorithme OSOM, la classe de sous-graphes \mathcal{G} considérée est la classe des cliques.

	Q_{label}	Q_{topo}^{ext}	$Q_{classif}^{ext}$	Q_{topo}^{int}	$Q_{classif}^{int}$
IRIS					
SOM	0.473	0.411	0.427	0.387	0.091
OSOM	0.473	0.394	0.429	0.375	0.084
EMOTIONS					
SOM	0.640	0.522	0.574	0.473	0.175
OSOM	0.640	0.507	0.572	0.469	0.163
SCENE					
SOM	0.522	0.481	0.497	0.292	0.174
OSOM	0.522	0.458	0.533	0.281	0.153
YEAST					
SOM	0.450	0.477	0.392	0.312	0.270
OSOM	0.450	0.447	0.377	0.280	0.274

TAB. 3 – Comparaison SOM vs. OSOM : résultats quantitatifs.

Les mesures Q pouvant être vues comme des distances matricielles, une faible valeur de Q exprime une forte corrélation entre les informations comparées. Le principal résultat concerne le fait que le modèle OSOM est presque systématiquement meilleur que son analogue non recouvrant, et ceci aussi bien sur des critères d'évaluation internes (ce qui était prévisible) que sur des critères externes (ce qui l'était moins). On observe de plus que lorsque OSOM n'améliore pas le critère $Q_{classif}^{ext}$, il s'agit de jeux de données pas ou très peu recouvrants (*scene* et *Iris*) ; ce résultat confirme le fait que l'algorithme OSOM est plutôt destiné à traiter des données naturellement recouvrantes. Enfin, nous observons de manière générale que d'une part les critères internes donnent logiquement lieu à de meilleures valeurs que leurs analogues externes et que d'autre part la différence est d'autant moins importante que la classification de référence est en forte adéquation avec les distances initiales (Q_{label} faible). Cette dernière observation vient conforter la cohérence globale de la méthodologie d'évaluation employée.

Pour terminer nous présentons en Figure 2 des exemples de cartes topologiques obtenues sur des grilles 10×10 avec la version initiale de SOM et la version accélérée de OSOM, sur les données *Emotions*.

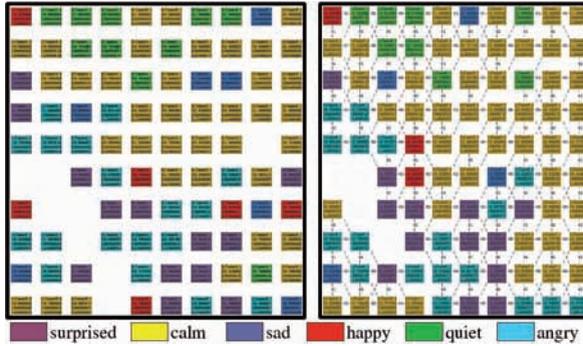


FIG. 2 – Exemple de cartes topologiques obtenues sur les données Emotions avec les algorithmes SOM (carte de gauche) et OSOM (carte de droite)

La couleur associée à chaque cellule de la grille a été choisie par un vote majoritaire sur les labels des exemples affectés à la cellule. De plus on ajoute pour les cartes recouvrantes la visualisation des recouvrements entre classes *via* des arêtes entre cellules ; cette information supplémentaire peut apporter une aide significative à la lecture et à l'interprétation de la carte.

D'une manière générale on observe que la structure globale de la carte est conservée d'une version à l'autre. Malgré la sur-représentation des exemples de la classe *calm*, on peut noter de légères variations lors du passage à la version recouvrante ; notamment une reconcentration des passages musicaux étiquetés *surprised* vers le bas de la carte et leur rapprochement avec les passages étiquetés *happy*.

5 Conclusion et perspectives

Dans cette étude nous nous sommes intéressé au couplage de deux approches de classification à savoir la classification recouvrante d'une part et les cartes auto-adaptatives d'autre part. La fusion de ces deux approches au sein d'un même modèle est motivée par les limites qu'elle permet de dépasser (complexité) et les contraintes qu'elle permet de contourner (paramétrage, cohérence topologique) en particulier pour les approches de classification recouvrante.

Nous avons ainsi présenté l'algorithme OSOM comme une généralisation de l'algorithme SOM bien connu et s'inspirant d'une variante recouvrante des k -moyennes. Les choix théoriques posés permettent d'assurer la convergence de l'algorithme et de répondre aux attentes liées à cette étude : limitation de la complexité, cohérence topologique assurée, etc. Le modèle a été validé par un ensemble d'expérimentations sur des domaines d'application cibles de la classification recouvrante ; à cette occasion un cadre d'évaluation globale a été proposé.

Nous envisageons de poursuivre cette étude autour de trois axes. Un premier axe visera à faire évoluer ce modèle vers des variantes pondérées et/ou kernélisées qui pourraient permettre de dépasser d'autres problématiques en classification recouvrante. Un second axe concernera une étude théorique sur la mise en correspondance des cartes topologiques recouvrantes avec les grilles supports de hiérarchies recouvrantes de type pyramides. Enfin, nous proposerons d'établir un lien théorique entre les problématiques de classification recouvrante d'une part et de classification reposant sur la théorie de l'évidence (Masson et Denoeux, 2008) d'autre part.

Références

- Banerjee, A., C. Krumpelman, J. Ghosh, S. Basu, et R. J. Mooney (2005). Model-based overlapping clustering. In *KDD '05 : Proceeding of the eleventh ACM SIGKDD*, New York, NY, USA, pp. 532–537. ACM Press.
- Bezdek, J. C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. *Plenum Press, New York*.
- Cleuziou, G. (2008). An extended version of the k-means method for overlapping clustering. In *19th ICPR Conference*, Tampa, Florida, USA, pp. 1–4.
- Dempster, A., N. Laird, et D. Rubin (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of Royal Statistical Society B* 39, 1–38.
- Diday, E. (1984). Une représentation visuelle des classes empiétantes : Les pyramides. Technical report, INRIA num.291, Rocquencourt 78150, France.
- D.J. Newman, S. Hettich, C. B. et C. Merz (1998). UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences.
- Hausdorff, F. (1962). *Set theory*. Chelsea.
- Heskes, T. (1999). Energy functions for self-organizing maps.
- Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytol.* 11, 37–50.
- Jain, A. K. et R. C. Dubes (1988). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey.
- Kohonen, T. (1984). Self-Organization and Associative Memory. *Springer*.
- Lesot, M.-J., F. d'Alché Buc, et G. Siolas (2003). Évaluation des cartes auto-organisatrices et de leur variante à noyaux. In PUG (Ed.), *Conférence CAp'2003*, Laval, pp. 139–154.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability*, Volume 1, Berkeley, pp. 281–297. University of California Press.
- Masson, M.-H. et T. Denoeux (2008). ECM : An evidential version of the fuzzy c-means algorithm. *Pattern Recognition* 41(4), 1384–1397.

Summary

Overlapping clustering models are known to provide data organizations that are more fitted to the input data. Moreover, unsupervised neural networks bring efficient solutions to visualize class structures. The topic of the present paper is then about the extension of the usual (crisp) self-organizing maps (SOM) to overlapping ones. We show that Overlapping-SOM allows to solve problems recurrent in overlapping clustering: number of clusters, complexity of the algorithm and coherence of the overlaps.

We present the algorithm OSOM that uses, on one hand, an overlapping variant of the c-means clustering algorithm and, on the other hand, the Kohonen approach, in order to build overlapping topologic maps. The algorithm is discussed on a theoretical point of view (associated energy function, complexity, etc.) and as regards to experimental results on real world data.