Classification évidentielle avec contraintes d'étiquettes

Violaine Antoine*, Nicolas Labroche**

*Université Blaise Pascal, LIMOS, UMR CNRS 6158, BP 10125, 63000 Clermont-Ferrand,
violaine.antoine@univ-bpclermont.fr,
**Université François Rabelais de Tours, LI EA 6300, Campus de Blois, 41000 Blois,
nicolas.labroche@univ-tours.fr.

Résumé. Ce papier propose une version améliorée de l'algorithme de classification automatique évidentielle semi-supervisée SECM. Celui-ci bénéficie de l'introduction de données étiquetées pour améliorer la pertinence de ses résultats et utilise la théorie des fonctions de croyance afin de produire une partition crédale qui généralise notamment les concepts de partitions dures et floues. Le pendant de ce gain d'expressivité est une complexité qui est exponentielle avec le nombre de classes, ce qui impose en retour l'utilisation de schémas efficaces pour optimiser la fonction objectif. Nous proposons dans cet article une heuristique qui relâche la contrainte classique de positivité liée aux masses de croyances des méthodes évidentielles. Nous montrons sur un ensemble de jeux de données de test que notre méthode d'optimisation permet d'accélérer sensiblement l'algorithme SECM avec un schéma d'optimisation classique, tout en améliorant également la qualité de la fonction objectif.

1 Introduction

Ce papier propose un nouveau mécanisme d'optimisation pour l'algorithme de classification automatique évidentielle semi-supervisée SECM (Antoine et al., 2014), qui est le premier à reposer sur des contraintes exprimées sous la forme de données étiquetées. Les algorithmes de classification évidentielle (Masson et Denœux, 2008, 2009) reposent sur le cadre théorique des fonctions de croyance et permettent de représenter tous les types d'affectations partielles grâce au concept de partition crédale qui étend la notion de partition stricte, floue et possibiliste. Ces méthodes évidentielles ont été étendues dans le cadre semi-supervisé (Antoine et al., 2012, 2014) pour pouvoir tirer partie de contraintes de type Must-Link (ML) et Cannot-Link (CL) qui spécifient si deux données doivent ou non appartenir à la même classe. La transformation des informations disponibles a priori en ce type de contraintes peut néanmoins induire une perte de connaissance. L'algorithme SECM a été proposé récemment pour tirer partie de données partiellement étiquetées (Antoine et al., 2014). Cependant, l'algorithme SECM initial repose sur une optimisation stricte qui respecte l'ensemble des contraintes et notamment la positivité des masses de croyances associées à l'affectation d'un point à une classe. Cette contrainte théorique entraîne la formation d'un problème complexe. Nous proposons donc de modifier le mécanisme d'optimisation en relâchant la contrainte de positivité, à l'instar de ce

qui est fait dans (Bouchachia et Pedrycz, 2006), et en s'assurant a posteriori de l'optimisation que les masses de croyances sont positives. Nos résultats expérimentaux montrent que notre heuristique ne dégrade pas les performances de l'algorithme SECM et permet de gagner de manière significative en complexité sur nos jeux de tests.

Ce papier est organisé comme suit : la section 2 présente les concepts fondamentaux de la théorie des fonctions de croyance et les principales méthodes de classification automatique sous contraintes. L'algorithme semi-supervisé SECM (Antoine et al., 2014) est ensuite décrit dans la section 3 et un nouveau schéma d'optimisation est proposé. Enfin, les résultats de l'algorithme sont présentés dans la section 4. Le papier conclut sur l'intérêt de la nouvelle méthode d'optimisation.

2 Travaux existants

2.1 Les fonctions de croyance

L'intérêt principal d'un algorithme de classification évidentielle est de pouvoir représenter le doute concernant l'affectation d'un point à un cluster. Pour ce faire, ces méthodes reposent sur la théorie de l'évidence de Dempster-Shafer, également appelée théorie des fonctions de croyance (Shafer, 1976; Smets et Kennes, 1994). Soit ω une variable prenant ses valeurs dans un ensemble fini $\Omega = \{\omega_1, \ldots, \omega_c\}$ appelé cadre de discernement. La connaissance partielle concernant la valeur de ω peut être représentée par une fonction de masses m, qui est une application de l'ensemble des parties de Ω dans l'intervalle $[0, 1]$ vérifiant $\sum_{A \subseteq \Omega} m(A) = 1$.

Les sous-ensembles $A \subseteq \Omega$ tels que $m(A) > 0$ sont appelés les éléments focaux de m. La quantité $m(A)$ s'interprète comme la quantité de croyance allouée à A et qui, faute d'information complémentaire, ne peut être allouée à aucun autre sous-ensemble de A. L'ignorance totale correspond à $m(\Omega) = 1$ alors qu'une certitude totale se rapporte à l'allocation complète de la masse de croyance sur un unique singleton de Ω. Si tous les ensembles focaux de m sont des singletons, alors la fonction de masses de croyances est équivalente à une distribution de probabilités. La quantité $m(\emptyset)$ peut être interprétée comme la croyance que la valeur réelle de ω n'appartient pas à Ω. Quand $m(\emptyset) = 0$, la fonction de croyance est dite normalisée. La connaissance exprimée par une fonction de croyance peut aussi être représentée par une fonction de plausibilité $pl : 2^{\Omega} \rightarrow [0, 1]$ définie comme suit :

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \qquad \forall A \subseteq \Omega. \tag{1}$$

La quantité $pl(A)$ est interprétée comme le degré maximal de croyance qui peut potentiellement être affecté à l'hypothèse selon laquelle la vraie valeur de ω appartient à A. Quand une décision doit être prise concernant la valeur de ω, il est intéressant de transformer une fonction de masses en probabilité pignistique (Smets et Kennes, 1994) :

$$BetP(\omega) = \sum_{\omega \in A} \frac{m(A)}{|A|}, \qquad \forall \omega \in \Omega, \tag{2}$$

où $|A|$ dénote la cardinalité de $A \subseteq \Omega$. Quand il existe $m(\emptyset) \neq 0$, une étape de normalisation doit précéder la transformation pignistique. La normalisation de Dempster, qui consiste à diviser toutes les masses par $1 - m(\emptyset)$, est une méthode classique de normalisation.

2.2 Algorithme des c-moyennes évidentielles

La version évidentielle des k-moyennes, ECM, est un algorithme de classification automatique qui construit une partition crédale à partir des données. Dans ce formalisme, la connaissance partielle concernant l'appartenance d'un objet $\mathbf{x}_i$ est représentée par une fonction de croyance m_i sur l'ensemble Ω des classes possibles. Ainsi, un degré de croyance peut être affecté aux singletons (comme dans les approches floues et possibilistes) mais également à tous les sous-ensembles de Ω. Soit $\{\mathbf{x}_1, \ldots \mathbf{x}_n\}$ un ensemble d'individus dans $\mathbb{R}^p$ à classer dans un ensemble $\Omega = \{\omega_1, \ldots \omega_c\}$ de c classes. Pour chaque objet x_i, la fonction de croyance m_i est calculée en plaçant une grande (resp. petite) quantité de croyance sur le sous-ensemble proche (resp. éloigné) en terme de distance de $\mathbf{x}_i$. La distance d_{ij} est une métrique définie entre un objet $\mathbf{x}_i$ et une représentation dans $\mathbb{R}^p$ d'un sous- ensemble $A_j \subseteq \Omega$. Similairement à l'algorithme des c-moyennes floues, chaque classe ω_k est représentée par un prototype $\mathbf{v}_k$. Pour chaque sous-ensemble $A_j \subseteq \Omega$, $A_j \neq \emptyset$, un centre $\overline{\mathbf{v}}_j$ est calculé comme le barycentre des centres associés aux classes composant A_j :

$$\overline{\mathbf{v}}_j = \frac{1}{|A_j|} \sum_{k=1}^{c} s_{kj} \mathbf{v}_k \quad \text{avec } s_{kj} = \begin{cases} 1 & \text{si } \omega_k \in A_j, \\ 0 & \text{sinon.} \end{cases} \tag{3}$$

La distance d_{ij}^2 peut être définie comme une distance euclidienne (Masson et Denœux, 2008). Plus récemment, une variante a été proposée pour prendre en compte une distance de Mahalanobis (Antoine et al., 2012). Similairement aux travaux de (Gustafson et Kessel, 1979), cette distance permet de détecter des clusters ayant différentes formes géométriques, grâce à une matrice de covariance floue $\mathbf{S}_k$ associée à chaque cluster ω_k et qui doit être optimisée.

Ensuite, similairement à ce qui est fait pour les prototypes, pour chaque sous-ensemble de A_j qui n'est pas un singleton, une matrice $\overline{\mathbf{S}}_j$ est calculée en moyennant les matrices incluses dans A_j. La distance d_{ij}^2 entre un objet $\mathbf{x}_i$ et un centre $\overline{\mathbf{v}}_j$ est alors $(\mathbf{x}_i - \overline{\mathbf{v}}_j)^T \overline{\mathbf{S}}_j (\mathbf{x}_i - \overline{\mathbf{v}}_j)$.

L'algorithme ECM minimise la fonction objectif suivante en fonction des matrices $\mathbf{M}$, $\mathbf{V}$ et $\mathbf{S}$ précédentes :

$$J_{ECM}(\mathbf{M}, \mathbf{V}, \mathbf{S}) = \sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} d_{ij}^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^{\beta}, \tag{4}$$

avec :

$$\sum_{j/A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} = 1 \quad \forall i = 1 \ldots n, \tag{5}$$

et

$$m_{ij} \geq 0, \quad \forall i = 1 \ldots n, \quad \forall A_j \subseteq \Omega, \tag{6}$$

où $m_{ij} = m_i(A_j)$ et $m_{i\emptyset} = m_i(\emptyset)$. Comme $m_{i\emptyset}$ correspond à la croyance que $\mathbf{x}_i$ est un point aberrant, son cas est traité séparément du reste des autres sous-ensembles. Le paramètre

δ indique la distance de l'ensemble des objets à l'ensemble vide. Il est intéressant de remarquer à ce niveau qu'une pénalité des sous-ensembles $A_j \in \Omega$ avec une grande cardinalité a été introduite avec la pondération $|A_j|^\alpha$. L'exposant α permet de contrôler le degré de cette pénalisation.

Tout comme pour les c-moyennes floues, la partition est construite selon un processus itératif qui optimise alternativement les matrices **M**, **V** et **S**. La complexité d'un algorithme évidentiel est linéaire avec le nombre de données mais exponentiel avec le nombre de classes. En conséquence, il est crucial pour ce type de méthodes de minimiser les calculs réalisés dans les phases d'optimisation comme cela est proposé dans cet article.

2.3 Algorithmes semi-supervisés

La plupart des méthodes de classification automatique ont été améliorées pour prendre en compte la connaissance experte sous la forme de contraintes soit entre paires de données de type Must-Link (ML) ou Cannot-Link (CL) qui indiquent si deux points doivent ou non appartenir au même cluster, soit sous la forme de données étiquetées (Wagstaff et al., 2001). Citons par exemple des algorithmes de type k-moyennes (Wagstaff et al., 2001; Basu et al., 2002), hiérarchiques (Davidson et Ravi, 2005), basés sur la densité (Ruiz et al., 2010; Lelis et Sander, 2009), des méthodes spectrales (Wang et Davidson, 2010) ainsi que des algorithmes dédiés aux flux de données (Ruiz et al., 2009). D'autres travaux se sont intéressés à l'intégration de contraintes dans l'algorithme des c-moyennes floues (Grira et al., 2006; Pedrycz, 1985; Bensaid et al., 1996; Pedrycz et Waletzky, 1997a). Pour palier les limitations des algorithmes flous en présence de bruit ou de points aberrants, des méthodes possibilistes (Krishnapuram et Keller, 1993; Sen et Davé, 1998) et plus récemment évidentielles ont été proposées (Masson et Denœux, 2008, 2009). Ces dernières ont également été étendues au cas semi-supervisé pour bénéficier des avantages des modèles basés sur les fonctions de croyance dans la prise en compte de la connaissance experte. Les travaux proposés reposent soit sur des contraintes ML et CL (Antoine et al., 2012, 2014) soit, plus récemment, sur des données partiellement étiquetées avec l'algorithme SECM (Antoine et al., 2014).

D'un point de vue formel, deux approches ont été proposées dans la littérature pour prendre en compte les contraintes et les étiquettes pendant le processus de classification automatique. En premier lieu, il est possible de modifier le processus des algorithmes de classification, soit durant la phase d'initialisation (Basu et al., 2002), soit pendant la phase de convergence. Dans ce dernier cas, on peut soit imposer un respect strict des contraintes comme dans COP Kmeans (Wagstaff et al., 2001), soit modifier la fonction objectif pour pénaliser les solutions qui ne respectent pas complètement les contraintes (Pedrycz et Waletzky, 1997b). Par exemple, dans (Bouchachia et Pedrycz, 2003), les auteurs décrivent un FCM amélioré dont la fonction objectif introduit un terme de pénalité qui considère à la fois l'appartenance actuelle des points i aux classes k notée u_{ik}, mais également l'appartenance telle qu'elle devrait être à partir des contraintes de l'expert notée $\tilde{u}_{ik}$ comme le montre l'équation (7).

$$J(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{n}\sum_{k=1}^{c} u_{ik}^{\beta} d_{ik}^2 + \gamma \sum_{i=1}^{n}\sum_{k=1}^{c} (u_{ik} - \tilde{u}_{ik})^{\beta} d_{ik}^2, \tag{7}$$

A	$m_1(A)$	$m_2(A)$	$m_3(A)$	$m_4(A)$
$\emptyset$	1	0	0	0
ω_1	0	1	0	0
ω_2	0	0	0	1
Ω	0	0	1	0

TAB. 1 – *Exemple de partition crédale.*

A	$pl_1(A)$	$pl_2(A)$	$pl_3(A)$	$pl_4(A)$
ω_1	0	1	1	0
ω_2	0	0	1	1

TAB. 2 – *Plausibilités calculées à partir de la partition crédale.*

où **U** et **V** dénotent respectivement la matrice d'appartenance et les coordonnées des centres des clusters. γ est un paramètre de régulation qui permet d'équilibrer l'importance du respect des contraintes dans la fonction objectif.

En second lieu, d'autres méthodes proposent d'adapter la métrique en fonction des contraintes et étiquettes fournies par l'expert comme dans l'algorithme MPC k-means (Bilenko et al., 2004). Par exemple, dans (Bouchachia et Pedrycz, 2006), les auteurs proposent une méthode pour adapter une distance de Gustafson-Kessel (Gustafson et Kessel, 1979). Nous proposons dans ce papier de considérer un modèle de contraintes flexibles avec une modification de la fonction objectif qui pénalise les solutions ne respectant pas les données étiquetées.

3 Algorithme SECM

3.1 Formalisation du problème

L'idée principale de l'algorithme proposé dans (Antoine et al., 2014) est d'ajouter un terme de pénalité dans la fonction objectif de ECM afin de prendre en compte un ensemble d'objets étiquetés. La démarche suivie est la même que dans (Bouchachia et Pedrycz, 2003) mais rapportée aux algorithmes évidentiels. L'expression d'un objet étiqueté peut se traduire sous la forme d'une fonction quantifiant la croyance sur l'appartenance de l'objet à une classe. Considérons dans un premier temps une partition crédale connue et définie par le tableau 1. Elle représente la connaissance partielle de l'appartenance de quatre objets à deux classes. Il est alors possible de calculer la plausibilité de chaque objet $\mathbf{x}_i$ pour chaque classe ω_k, comme illustré par le tableau 2. On remarque alors qu'une plausibilité nulle permet de déduire avec certitude qu'un élément n'appartient pas à une classe. Ainsi, l'observation de $pl_i(\omega_1) = 0$ permet de déduire que $\mathbf{x}_1$, un objet atypique, et $\mathbf{x}_4$, un objet affecté avec certitude dans la classe ω_2, ne font pas partie de la classe ω_1. En revanche, les objets dont la plausibilité pour une classe est élevée ont des chances d'appartenir à cette classe. Ainsi, $pl_i(\omega_1) = 1$ apparaît pour l'objet $\mathbf{x}_2$, qui appartient à la classe ω_2 avec certitude, et pour l'objet $\mathbf{x}_3$, qui appartient soit à ω_1, soit à ω_2.

Supposons maintenant que l'on ne dispose pas de la partition crédale, mais qu'il existe des contraintes sous formes d'étiquettes. Par exemple, l'objet $\mathbf{x}_i$ est inclus dans la classe ω_k. Il est alors possible d'imposer la contrainte $pl_i(\omega_k) = 1$. L'effet sera d'exiger :

— une croyance élevée pour les fonctions de masse ayant un sous-ensemble comprenant ω_k, donc toutes les fonctions de masses qui ont un degré de croyance plus ou moins fort sur le fait que $\mathbf{x}_i$ appartienne à ω_k,

— des valeurs faibles pour toutes les fonctions de masses ayant un sous-ensemble qui n'incluent pas ω_k.

La contrainte entre un objet $\mathbf{x}_i$ et la classe ω_k est donc respectée pour de nombreuses solutions allant de la certitude totale que $\mathbf{x}_i$ appartienne à ω_k jusqu'à l'incertitude complète de l'affectation de $\mathbf{x}_i$ entre ω_k ou plusieurs autres classes de Ω. La contrainte est donc flexible car elle permet si nécessaire de garder un doute quant à l'affectation de l'objet à la classe. Par conséquent, cela limite l'influence négative d'une contrainte bruitée.

Lorsqu'un expert crée des contraintes d'étiquettes, il peut avoir un doute entre plusieurs classes pour un unique objet. Par exemple, l'objet $\mathbf{x}_i$ appartient à une des classes du sous-ensemble $A_j \in \Omega$. Cette information se modélise alors sous la forme d'une contrainte sur la plausibilité de A_j : $pl_i(A_j) = 1$. Cela revient à favoriser les fonctions de masses ayant au moins une classe dans A_j. Cette contrainte, qui généralise la précédente, permet d'établir un terme de pénalité à ajouter à la fonction objectif de ECM :

$$J_S = \sum_{i=1}^{n} \sum_{A_j \in \Omega, A_j \neq \emptyset} b_{ij}(1 - pl_i(A_j)), \tag{8}$$

avec

$$b_{ij} = \begin{cases} 1 & \text{si } \mathbf{x}_i \in \omega_k \text{ et } \omega_k \in A_j. \\ 0 & \text{sinon.} \end{cases} \tag{9}$$

La nouvelle fonction objectif est alors la suivante :

$$J_{SECM} = (1 - \gamma)\frac{1}{2^c n} J_{ECM} + \gamma \frac{1}{s} J_S, \tag{10}$$

sous les contraintes (5) et (6). Le paramètre s correspond au nombre de contraintes existantes et les coefficients $\frac{1}{2^c n}$ et $\frac{1}{s}$ sont ajoutés afin de normaliser chaque terme. Ainsi, le paramètre $\gamma \in [0, 1]$ est utilisé pour contrôler l'importance donnée aux contraintes par rapport au modèle géométrique. L'allocation de croyance aux sous-ensembles de fortes cardinalités est pénalisée uniquement par le terme J_{ECM}, par le biais du coefficient $|A_k|^\alpha$. Ce système permet d'adapter plus aisément les contraintes à la structure inhérente des données.

3.2 Optimisation

L'optimisation du nouveau critère consiste, de la même manière que pour l'algorithme ECM, à minimiser alternativement les matrices **M**, **V** et **S**. Le terme de pénalité J_S ne dépendant ni de **V**, ni de **S**, leur mise à jour est similaire à ECM, et leur formule est présentée dans (Masson et Denœux, 2008). La partition crédale **M** est au contraire présente dans J_S. En fixant β à 2 alors la minimisation de la fonction objectif par rapport à **M** devient un problème quadratique à contraintes linéaires. Ce problème peut être résolu par une méthode classique d'optimisation (Ye et Tse, 1989), néanmoins de nombreux auteurs se trouvant dans

un contexte similaire proposent d'optimiser directement la fonction objectif sans prendre en compte les contraintes de positivité sur la partition (6), afin de réduire le temps de convergence de l'algorithme.

Afin de résoudre le problème de minimisation contraint par (5), des multiplicateurs de Lagrange $\lambda_1, \dots \lambda_n$ sont introduits et le Lagrangien défini :

$$\begin{aligned}\mathcal{L}(\mathbf{M}) &= \xi(\sum_{i=1}^{n}\sum_{A_j\neq\emptyset}|A_j|^{\alpha}m_{ij}^2d_{ij}^2+\sum_{i=1}^{n}\delta^2m_{i\emptyset}^2)+\chi(\sum_{i=1}^{n}\sum_{A_j\neq\emptyset}b_{ij}(1-pl_i(A_j))) \\ &\quad -\sum_{i=1}^{n}\lambda_i\left(\sum_{A_j}m_{ij}+m_{i\emptyset}-1\right), \end{aligned} \tag{11}$$

avec, afin de simplifier l'écriture des équations, $\xi = (1-\gamma)\frac{1}{2^c n}$ et $\chi = \gamma\frac{1}{s}$.
Les dérivées partielles du Lagrangien sont donc :

$$\frac{\partial\mathcal{L}}{\partial m_{ij}} = 2\xi|A_j|^{\alpha}m_{ij}d_{ij}^2-\chi\left(\sum_{\omega_k\in A_j}b_{ik}\right)-\lambda_i, \tag{12}$$

$$\frac{\partial\mathcal{L}}{\partial m_{i\emptyset}} = 2\xi|A_j|^{\alpha}m_{i\emptyset}d_{ij}^2, \tag{13}$$

$$\frac{\partial\mathcal{L}}{\partial \lambda_i} = \sum_{A_j\subseteq\Omega, A_j\neq\emptyset}m_{ij}+m_{i\emptyset}-1. \tag{14}$$

Annuler les dérivées partielles permet d'obtenir les équations suivantes :

$$\frac{\partial\mathcal{L}}{\partial m_{ij}}=0 \Rightarrow m_{ij}=\frac{\lambda_i}{2\xi}\times\frac{1}{|A_j|^{\alpha}d_{ij}^2}+\frac{\chi\sum_{\omega_k\in A_j}b_{ik}}{2\xi|A_j|^{\alpha}d_{ij}^2}, \tag{15}$$

$$\frac{\partial\mathcal{L}}{\partial m_{i\emptyset}}=0 \Rightarrow m_{i\emptyset}=\frac{\lambda_i}{2\xi\delta^2}, \tag{16}$$

$$\frac{\partial\mathcal{L}}{\partial \lambda_i}=0 \Rightarrow \sum_{A_j\subseteq\Omega, A_j\neq\emptyset}m_{ij}+m_{i\emptyset}=1. \tag{17}$$

En utilisant (15) et (16) dans (17), il est possible d'écrire :

$$\frac{\lambda_i}{2\xi}=\frac{1-\sum_{A_j\subseteq\Omega,A_j\neq\emptyset}\chi\left(\sum_{\omega_k\in A_j}b_{ik}\right)\left(2\xi|A_j|^{\alpha}d_{ij}^2\right)^{-1}}{\sum_{A_j\subseteq\Omega,A_j\neq\emptyset}\left(|A_j|^{\alpha}d_{ij}^2\right)^{-1}+\delta^{-2}} \tag{18}$$

Cette équation peut finalement être utilisée dans (15) et (16) pour obtenir la mise à jour des fonctions de masse, $\forall i = 1, n$ et $\forall j / A_j \subseteq \Omega, A_j \neq \emptyset$:

$$m_{ij}=\frac{1-\sum_{A_j\subseteq\Omega,A_j\neq\emptyset}\chi\left(\sum_{\omega_k\in A_j}b_{ik}\right)\left(2\xi|A_j|^{\alpha}d_{ij}^2\right)^{-1}}{|A_j|^{\alpha}d_{ij}^2\left(\sum_{A_j\subseteq\Omega,A_j\neq\emptyset}\left(|A_j|^{\alpha}d_{ij}^2\right)^{-1}+\delta^{-2}\right)}+\frac{\chi\sum_{\omega_k\in A_j}b_{ik}}{2\xi|A_j|^{\alpha}d_{ij}^2}, \tag{19}$$

Nom	# objets	#attributs	# classes	Métrique
Iris	150	4	3	Mahalanobis
Wine	178	13	3	Euclidienne
LettersIJL	227	16	2	Mahalanobis
Ionosphere	351	34	2	Mahalanobis

TAB. 3 – *Description des jeux de données.*

et

$$m_{i\emptyset} = 1 - \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij}. \tag{20}$$

Comme le numérateur de la première partie de l'équation (19) permet l'obtention de valeurs négatives, la fonction de masse m_{ij} peut être négative. Ces valeurs négatives sont accentuées par l'importance donnée aux contraintes. Si l'utilisateur reste dans un cas normal d'utilisation des contraintes, c'est-à-dire $\gamma \leq 0.8$ (cf. partie 4.2), les valeurs de $m_{ij} < 0$ seront proches de 0. Une fonction de réajustement est donc envisageable sans que l'optimisation soit dégradée :

$$m^*_{ij} = \begin{cases} \frac{m_{ij}}{1+a_i} & \text{si } m_{ij} > 0, \\ 0 & \text{sinon,} \end{cases} \qquad \forall i = 1, n, \quad \forall A_j \subseteq \Omega \tag{21}$$

avec

$$a_i = - \sum_{A_j \subseteq \Omega} \min(m_{ij}, 0). \tag{22}$$

La nouvelle fonction de masse m^*_i pour l'objet i est alors comprise entre 0 et 1 et respecte la contrainte (5).

4 Expérimentations

Les expériences menées sur plusieurs jeux de données consistent à comparer les résultats obtenus par SECM lorsque la mise à jour des fonctions de masse utilise l'optimisation de (Ye et Tse, 1989), noté SECM-classic, avec SECM et l'optimisation proposée, noté SECM-do.

4.1 Données et méthode d'évaluation

Jeux de données : Plusieurs jeux de données issue de l'UCI Machine Learning Repository ont été employés. Le tableau 3 indique leurs caractéristiques ainsi que la métrique utilisée pour les expériences. Il faut noter que LettersIJL correspond au jeu de données Letters modifié comme (Bilenko et al., 2004).

Méthode d'évaluation : La partition réelle des jeux de données utilisés est initialement connue. Elle peut donc être comparée avec la partition calculée par SECM en utilisant le maximum de probabilité pignistique afin d'évaluer sa qualité. Pour cela, l'indice de Rand Ajusté

	ECM	SECM-classic			SECM-do		
	0%	10%	20%	30%	10%	20%	30%
Iris	3.43	5.39	3.87	3.34	2.77	2.19	1.88
Wine	3.53	4.92	3.96	3.57	1.87	1.49	1.29
LettersIJL	25.78	24.63	18.07	14.02	17.31	12.27	9.53
Ionosphere	1.07	11.31	8.55	6.94	6.94	5.36	4.49

TAB. 4 – *Temps CPU moyens en secondes trouvés avec SECM-classic et SECM-do pour 10, 20 et 30% de contraintes. Le temps CPU moyen de l'algorithme ECM est également renseigné.*

(ARI), qui représente un indice de Rand normalisé par rapport à une distribution hypergéométrique, est utilisé : $ARI = \frac{\text{Indice}-\text{Indice espéré}}{\text{indice maximum}-\text{indice espéré}}$. Une autre manière intéressante de synthétiser l'information contenue dans une partition crédale est d'affecter chaque objet au sous-ensemble de plus forte masse. Ainsi, il est possible d'obtenir une partition nommée partition crédale dure avec au plus 2^c groupes.

Paramétrages : Pour chaque expérience, α est fixé à 1 afin de faiblement pénaliser les sous-ensemble de fortes cardinalités et δ^2=1000 de manière à éviter d'allouer de la croyance au sous-ensemble vide. En effet, les jeux de données utilisés ne contiennent pas d'objets atypiques. Le coefficient γ permettant d'ajuster l'importance attribuée aux contraintes par rapport à la structure globale des classes a été testé pour trois valeurs : 0.3, 0.5 et 0.8. Enfin, les contraintes sont choisies aléatoirement et varient entre 5% et 30% de la taille d'un jeu de données.

Protocole expérimental : Une expérience consiste, pour un certain pourcentage de contraintes, à exécuter 25 fois l'algorithme SECM avec 25 jeux de contraintes différents. Afin d'éviter les optima locaux, chaque exécution teste cinq initialisations aléatoires des centres de gravité et récupère les résultats obtenus par l'initialisation ayant la fonction objectif minimale.

4.2 Résultats

Jeux de données réelles : La figure 1 montre l'évolution de l'indice de Rand moyen obtenu avec SECM-classic et SECM-do par rapport au pourcentage de contraintes pour Iris et Wine. Des résultats similaires ont été trouvés avec Ionosphere et LettersIJL. Le coefficient γ est fixé à 0.5. Il est ainsi possible de remarquer (1) que l'ajout progressif de contraintes améliore l'indice de Rand et (2) que l'algorithme SECM-do présente de meilleurs résultats que SECM-classic. Pour ces expériences, nous avons également constaté que les valeurs des fonctions objectif de SECM-do sont plus petites que celles de SECM-classic. Des résultats similaires ont été trouvés pour $\gamma = 0.3$ et $\gamma = 0.8$. La nouvelle optimisation permet donc d'obtenir un meilleur minimum grâce à sa relaxation des contraintes, ce qui implique de meilleurs résultats de classification. Il faut également noter que pour $\gamma = 0.8$, les résultats obtenus prouvent que la fonction de réajustement de SECM-do ne dégrade en rien les solutions.

Pour ces mêmes expériences, le temps CPU a été observé afin de comparer la vitesse d'exécution des deux algorithmes. Le tableau 4 présente les résultats obtenus. Il est ainsi aisé de voir que l'algorithme SECM-do est plus rapide que l'algorithme SECM-classic.

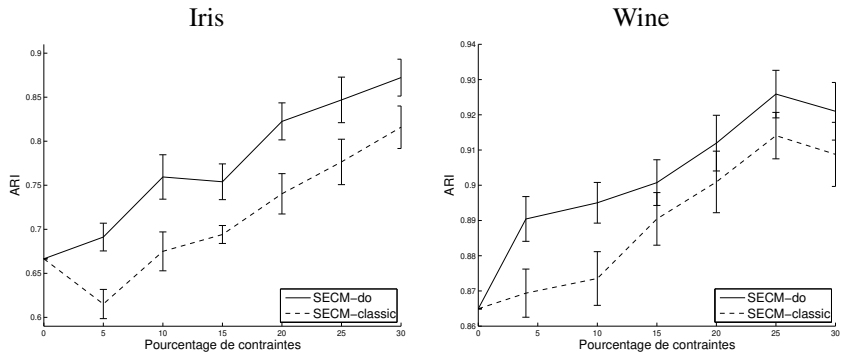

FIG. 1 – *ARI moyens et intervalle de confiance à 95% pour Iris et Wine.*

(a) (b) (c) (d)

FIG. 2 – *(a) image originale, (b) partition crédale obtenue avec SECM, (c) contraintes sur ω_1 et ω_2, (d) partition crédale obtenue avec SECM. Les zones blanches (resp. noires) des partitions représentent ω_1 (resp. ω_2) et les zones grises Ω.*

Application à la segmentation d'image : L'intérêt de SECM est maintenant illustré avec un exemple de segmentation d'image, ici un avion (cf. Figure 2(a)). Le but est d'isoler l'avion du reste de l'image. Dans une première expérience, nous avons utilisé ECM avec une distance de Mahalanobis. Nous avons considéré qu'il n'existe pas d'objet atypique, donc nous avons fixé δ^2 à une forte valeur. De plus, nous avons choisi de réduire l'incertitude trouvée par la partition finale en fixant α à une valeur élevée. Ainsi, ECM avec $c = 2$, $\alpha = 3$ et $\delta^2 = 1000$ trouve la partition crédale dure représentée par la figure 2(b). Nous pouvons remarquer que ECM ne permet pas d'isoler correctement l'avion. Dans une seconde expérience, nous introduisons des contraintes sur la partition comme illustré Figure 2(c). Chaque pixel de la première (respectivement seconde) zone est affectée à ω_1 (respectivement ω_2). L'algorithme SECM est alors exécuté avec les mêmes paramètres que ECM. La partition crédale résultante est présentée Figure 2(d). Nous pouvons constater que les contraintes ont permis de lever l'indétermination de la plupart des pixels alloués à Ω.

5 Conclusion

Nous avons présenté dans cet article une nouvelle méthode d'optimisation pour l'algorithme de classification automatique intitulé SECM. Ce dernier est une variante de l'algorithme évidentiel ECM prenant en compte des contraintes d'étiquettes. Il repose sur la minimisation

d'une fonction objectif avec des contraintes linéaires et non linéaires, ce qui impose l'utilisation de méthodes d'optimisation avancées. De plus, l'utilisation des fonctions de masses liées aux méthodes évidentielles rend la complexité de l'algorithme linéaire par rapport au nombre d'objets et exponentielle par rapport aux nombre de classes. Nous proposons donc de relâcher les contraintes de positivité sur les fonctions de masses, c'est-à-dire sur les contraintes non linéaires, afin de réduire l'optimisation de la partition à la méthode des multiplicateurs de Lagrange. Le respect des contraintes de positivité est ensuite vérifié par une méthode de réajustement. Nous avons montré sur un ensemble de jeux de données que cette nouvelle technique permet non seulement d'augmenter la rapidité de SECM mais qu'elle permet également d'améliorer les performances en trouvant de meilleurs minima. Les travaux futurs porteront sur l'étude de nouveaux formalismes plus rapides permettant de conserver la majeure partie de l'expressivité des méthodes évidentielles avec une complexité largement réduite pour permettre de traiter des jeux de données avec un nombre de classes plus important.

Références

Antoine, V., N. Labroche, et V.-V. Vu (2014). Evidential seed-based semi-supervised clustering. In *SCIS ISIS 2014, Joint 7th International Conference on Soft Computing and Intelligent Systems and 15th International Symposium on Advanced Intelligent Systems*.

Antoine, V., B. Quost, M.-H. Masson, et T. Denœux (2012). CECM : Constrained evidential -means algorithm. *Computational Statistics & Data Analysis 56*(4), 894 – 914.

Antoine, V., B. Quost, M.-H. Masson, et T. Denœux (2014). Evidential clustering with instance-level constraints for proximity data. *Soft Computing 18*(7), 1321–1335.

Basu, S., A. Banerjee, et R. J. Mooney. (2002). Semi-supervised clustering by seeding. In *Proceeding of the 19th International Conference on Machine Learning*, pp. 27–34.

Bensaid, A. M., L. O. Hall, J. C. Bezdek, et L. P. Clarke (1996). Partially supervised clustering for image segmentation. *Pattern Recognition 29*(5).

Bilenko, M., S. Basu, et R. J. Mooney (2004). Integrating constraints and metric learning in semi-supervised clustering. In *Intl. Conference on Machine Learning*, pp. 81–88.

Bouchachia, A. et W. Pedrycz (2003). A semi-supervised clustering algorithm for data exploration. In *Proc. Internat. Fuzzy Systems Association World Congress*, pp. 328–337.

Bouchachia, A. et W. Pedrycz (2006). Enhancement of fuzzy clustering by mechanisms of partial supervision. *Fuzzy Sets and Systems 157*, 1733–1759.

Davidson, I. et S. Ravi (2005). Agglomerative hierarchical clustering with constraints : Theoretical and empirical results. In *Proceeding of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 59–70.

Grira, N., M. Crucianu, et N. Boujemaa (2006). Fuzzy clustering with pairwise constraints for knowledge-driven image categorization. *IEEE Vision, Image, Processing 153*(3), 299–304.

Gustafson, D. et W. Kessel (1979). Fuzzy clustering with a fuzzy covariance matrix. In *Proc. IEEE Conf. Decision and Control*, pp. 61–766.

Krishnapuram, R. et J. Keller (1993). A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems 1*(2), 98–110.

Lelis, L. et J. Sander (2009). Semi-supervised density-based clustering. In *Proceedings of the 9th IEEE International Conference on Data Mining*, Washington, DC, USA, pp. 842–847.

Masson, M.-H. et T. Denœux (2008). ECM : An evidential version of the fuzzy c-means algorithm. *Pattern Recognition 41*, 1384–1397.

Masson, M.-H. et T. Denœux (2009). RECM : Relational evidential c-means algorithm. *Pattern Recognition Letters 30*, 1015–1026.

Pedrycz, W. (1985). Algorithm of fuzzy clustering with partial supervision. *Pattern Recognition Letters 3*, 13–20.

Pedrycz, W. et J. Waletzky (1997a). Fuzzy clustering with partial supervision. *IEEE Transactions on systems, Man, and Cybernetics 27*(5), 787–795.

Pedrycz, W. et J. Waletzky (1997b). Fuzzy clustering with partial supervision. *IEEE Transactions on systems, Man, and Cybernetics 27*(5), 787–795.

Ruiz, C., E. Menasalvas, et M. Spiliopoulou (2009). C-denstream : Using domain knowledge on a data stream. In S.-V. B. Heidelberg (Ed.), *DS 2009, LNAI 5808*, pp. 287–301.

Ruiz, C., M. Spiliopoulou, et E. Menasalvas (2010). Density-based semi-supervised clustering. *Data Mining and Knowledge Discovery 21*(3), 345–370.

Sen, S. et R. Davé (1998). Clustering of relational data containing noise and outliers. In *Fuzzy Systems Proceedings*, Volume 2, pp. 98–110.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton Univ. Press. Princeton, NJ.

Smets, P. et R. Kennes (1994). The transferable belief model. *Artificial Intelligence 66*, 191–234.

Wagstaff, K., C. Cardie, S. Rogers, et S. Schroedl (2001). Constrained k-means clustering with background knowledge. In *Proceedings of the 18th International Conference on Machine Learning*, pp. 577–584.

Wang, X. et I. Davidson (2010). Flexible constrained spectral clustering. In *Proceeding of the Conference on Knowledge Discovery and Data Mining*, pp. 563–572.

Ye, Y. et E. Tse (1989). An extension of Karmarkar's projective algorithm for convex quadratic programming. *Mathematical Programming 44*(1), 157–179.

Summary

This paper proposes an improved optimization for the SECM evidential clustering algorithm. SECM benefits from the introduction of labelled objects to guide its output partition towards a desired solution. It also takes advantage of the belief functions theory to generate a credal partition that generalizes the concept of crisp and fuzzy partition. The counterpart of this gain of expressivity is the complexity which grows exponentially with the number of clusters. Thus, efficients methods should be used in order to optimize the objectif function. We propose in this article a heuristic that releases the classic constraint of positivity related to the mass functions coming from evidential methods. We show on several datasets the efficiency of our new optimization method in term of accuracy and speed.