CECM: Constrained Evidential C-Means algorithm

V. Antoine[a,c,*], B. Quost[a,c], M.-H. Masson[b,c], T. Denœux[a,c]

[a] *Université de Technologie de Compiègne*
[b] *Université de Picardie Jules Verne, IUT de l'Oise*
[c] *Heudiasyc, UMR CNRS 6599, BP 20529, 60205 Compiègne, France*

Abstract

In clustering applications, prior knowledge about cluster membership is sometimes available. To integrate such auxiliary information, constraint-based (or semi-supervised) methods have been proposed in the hard or fuzzy clustering frameworks. This approach is extended to evidential clustering, in which the membership of objects to clusters is described by belief functions. A variant of the Evidential C-means (ECM) algorithm taking into account pairwise constraints is proposed. These constraints are translated into the belief function framework and integrated in the cost function. Experiments with synthetic and real data sets demonstrate the interest of the method. In particular, an application to medical image segmentation is presented.

Keywords: Clustering, semi-supervised learning, pairwise constraints, adaptive metric, active learning, belief functions, Dempster-Shafer theory, evidence theory.

1. Introduction

Clustering methods aim at grouping objects into clusters based on similarity between their descriptors. However, there are some situations in which some background knowledge about the problem is available. Making use of this extra information in a clustering algorithm can help us to guide the method towards a desired solution and to improve the classification accuracy [4, 14]. Prior information can be exploited at different levels such as: the *cluster* level with, for instance, a minimum distance neighborhood [9], the *model* level with the requirement of balanced clusters [32] or the specification of non desired solutions [13], or at the *instance* level.

Wagstaff [28] proposed to introduce two types of instance-level constraints: the first one specifies that two objects have to be in the same cluster (*must-link* constraint) while the second one specifies that two objects should not be put

*Corresponding author: Violaine Antoine, Université de Technologie de Compiègne, Laboratoire Heudiasyc, UMR CNRS 6599, BP 20529, 60205 Compiègne, France.
Email address: violaine.antoine@hds.utc.fr (V. Antoine)

in the same cluster (*cannot-link* constraint). Such pairwise constraints have been considered and integrated in many unsupervised algorithms such as the hard or the fuzzy c-means (FCM), and have recently become a topic of great interest [29, 2, 27, 9]. They have been incorporated in many different ways, generally by including a penalty term in the objective function [1, 15] or by altering the distances between objects with respect to the constraints [29].

In the FCM algorithm, each object may belong to one or more clusters with different degrees of membership. These degrees of membership are stored into a fuzzy partition matrix $U = (u_{ik})$ and are calculated by minimizing a suitable objective function with respect to the constraints

$$u_{ik} \geq 0 \quad \forall\, i, k, \tag{1}$$

and

$$\sum_{k=1}^{c} u_{ik} = 1, \tag{2}$$

where $u_{ik} \in [0, 1]$ denotes the degree of membership of object i to cluster k, and c is the number of clusters. Nevertheless the method sometimes produces counterintuitive results and has poor robustness against noise and outliers. This is the reason why possibilistic methods [12, 19, 8] and, more recently, evidential clustering methods grounded in the theory of belief functions [11, 21, 22, 23] have been proposed.

Evidential clustering is based on a new concept of partition, referred to as a *credal* partition, which extends the existing concepts of hard, fuzzy and possibilistic partitions. This is done by allocating, for each object, a *mass of belief*, not only to single clusters, but also to any subset of the set of clusters $\Omega = \{\omega_1, \ldots, \omega_c\}$. As shown in the experiments reported in [11] and [22], this additional flexibility can be exploited to construct meaningful and robust summaries of the data. For instance, it is possible to compute, for each cluster, a set of objects that *surely* belong to it, and a larger set of objects that *possibly* belong to it. Such qualitative summaries may be argued to be more intuitive and easier to interpret than purely numerical results such as fuzzy partitions, while being much richer than classical hard partitions. Robustness is achieved by assigning outliers to the empty set.

One of the algorithms designed to derive a credal partition from data, called Evidential C-Means (ECM), can be considered as a direct extension of FCM [22]. In this paper, we propose to introduce pairwise constraints in the ECM algorithm, in order to create a new algorithm, called CECM, which combines the advantages of adding background knowledge and using belief functions. Furthermore, we present a formulation of ECM that adapts the metric using a Mahalanobis distance so that the constraints may be more easily satisfied. Finally, we propose an active learning scheme, based on the credal partition, which makes it possible to select efficient pairwise constraints.

The remaining of this paper is organized as follows. Section 2 first recalls the necessary background on belief functions, fuzzy clustering and the ECM algorithm. The basic version of the constrained ECM (CECM) algorithm with

Euclidean distance and a more sophisticated version with an adaptive Mahalanobis distance are then introduced in Sections 3 and 4, respectively. Section 5 describes the experimental settings and the results. Finally, we conclude and present some perspectives in Section 6.

2. Background

In this section, the necessary background on the theory of belief functions (Subsection 2.1), fuzzy clustering (Subsection 2.2) and the ECM algorithm (Subsection 2.3) will first be recalled.

2.1. Belief functions

The Dempster-Shafer theory of evidence [3, 24, 26] (or belief function theory) is a theoretical framework for representing partial and unreliable information.

Let us consider a variable ω taking values in a finite set $\Omega = \{\omega_1, \ldots, \omega_c\}$ called the frame of discernment. Partial knowledge regarding the actual value taken by ω can be represented by a *mass function* m, which is an application from the power set of Ω in the interval $[0, 1]$ such that

$$\sum_{A \subseteq \Omega} m(A) = 1. \tag{3}$$

The subsets A of Ω such that $m(A) > 0$ are called the *focal sets* of m. The value of the focal set $m(A)$ can be interpreted as a fraction of a unit mass of belief that is allocated to A and that cannot be allocated to any subset of A. Complete ignorance is obtained when Ω is the only focal set, and full certainty when the whole mass of belief is assigned to a unique singleton of Ω (m is then said to be a *certain* mass function). If all the focal sets of m are singletons, m is similar to a probability distribution: it is then called a *Bayesian* mass function. A mass function m such that $m(\emptyset) = 0$ is said to be normalized. Under the *open-world* assumption, a mass function $m(\emptyset) > 0$ is interpreted as a quantity of belief given to the hypothesis that the actual value of ω might not belong to Ω [25].

Given a mass function m, it is possible to define a plausibility function $pl : 2^\Omega \to [0, 1]$ and a belief function $bel : 2^\Omega \to [0, 1]$ by:

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B) \quad \forall A \subseteq \Omega, \tag{4}$$

and

$$bel(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B) \quad \forall A \subseteq \Omega. \tag{5}$$

Functions bel and pl are linked by the following relation:

$$pl(A) = 1 - m(\emptyset) - bel(\overline{A}), \tag{6}$$

where $\overline{A}$ denotes the complement of A. The quantity $bel(A)$ is interpreted as a degree of belief in A, taking into account the mass of belief given to A and nonempty subsets of A. In contrast, $pl(A)$ measures to what extent one fails to believe in $\overline{A}$.

In order to make a decision regarding the value of ω, it is possible to transform the mass function into a pignistic probability distribution [26], defined, for a normalized mass function, as:

$$BetP(\omega) = \sum_{\omega \in A} \frac{m(A)}{|A|} \quad \forall \omega \in \Omega, \tag{7}$$

where $|A|$ denotes the cardinality of $A \subseteq \Omega$. If $m(\emptyset) \neq 0$, then a normalization step has to be performed before carrying out the pignistic transformation. Various methods may be applied. In particular, Dempster's normalization consists in dividing all the masses by $1 - m(\emptyset)$, whereas Yager's normalization transfers $m(\emptyset)$ to $m(\Omega)$ [30].

2.2. Fuzzy c-means and variants

Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a collection of vectors in $\mathbb{R}^p$ describing n objects to classify in the set $\Omega = \{\omega_1 \ldots \omega_c\}$. Each cluster ω_k for $k \in \{1, \ldots, c\}$ is represented by a prototype or a center $\mathbf{v}_k \in \mathbb{R}^p$. Let V denote the matrix composed of the cluster centers, and let $U = (u_{ik})$ define a fuzzy partition matrix that contains the degrees of membership of each object to each cluster. The FCM algorithm [5] computes V and U so as to minimize the following objective function:

$$J_{\text{FCM}}(U, V) = \sum_{i=1}^{n} \sum_{k=1}^{c} u_{ik}^{\beta} d_{ik}^2, \tag{8}$$

subject to (1) et (2). In the objective function (8), d_{ik} represents the Euclidean distance between the object $\mathbf{x}_i$ and the centroid $\mathbf{v}_k$ and $\beta > 1$ is a weighting exponent that controls the fuzziness of the partition. The objective function is minimized using an iterative algorithm, which alternatively optimizes the cluster centers and the membership degrees. The update formulas for the masses and the centers are obtained by computing the Lagrangian formulation of the optimization problem and by setting its partial derivatives with respect to the parameters to zero [5]. The algorithm starts from an initial guess for either the partitioning matrix or the cluster centers and iterates until convergence.

To detect noisy data or outliers, Davé [8] has proposed a variant of FCM called the "noise-clustering" algorithm (NC). It consists in adding to the c initial clusters a "noise" cluster, associated to a fixed distance ρ to all objects. The parameter ρ controls the amount of data considered as outliers. The membership u_{i*} of an object i to the noise cluster is given by:

$$u_{i*} = 1 - \sum_{k=1}^{c} u_{ik} \quad i = 1, n. \tag{9}$$

The objective function to be minimized is expressed as follows:

$$J_{\text{NC}}(U,V) = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{\beta} d_{ij}^2 + \sum_{i=n}^{c} \rho^2 u_{i*}^{\beta}. \tag{10}$$

Writing the optimality conditions of the problem leads, as in FCM, to direct adaptation formulas for the membership degrees and the cluster centers.

The Gustafson and Kessel algorithm [16] is another interesting variant of FCM. This algorithm extends FCM by using an adaptive distance, in order to detect clusters of different geometrical shapes. Each cluster has its own norm-inducing matrix S_k defined as its fuzzy covariance matrix [16]. The adaptation formulas of FCM for the membership degrees and the centers remain valid as they do not depend on the metric.

2.3. ECM algorithm

Recently, Masson and Denœux proposed a credibilistic version of Davé's algorithm [22] by replacing the fuzzy partition matrix U with a more general kind of partition M called a *credal partition*. In this framework, partial knowledge regarding the class membership of an object is represented by a mass function on the set Ω of possible classes. Thus, belief mass may be given to any subset A of Ω (any set of classes), and not only to singletons of Ω. This representation makes it possible to model a wide variety of situations ranging from complete ignorance to full certainty, as illustrated in the following example.

Example 1. *Let us consider a collection of four objects that need to be classified into two classes. A credal partition is presented in Table 1. The class of the first object is known with certainty, whereas the class of the second object is completely unknown. We have probabilistic knowledge of the actual class of the third object. The last object is considered to be an outlier, which is represented by allocating the whole unit mass to the empty set.*

Table 1: Example of a credal partition

A	$m_1(A)$	$m_2(A)$	$m_3(A)$	$m_4(A)$
$\emptyset$	0	0	0	1
$\{\omega_1\}$	1	0	0.3	0
$\{\omega_2\}$	0	0	0.7	0
Ω	0	1	0	0

A credal partition can thus be seen as a general model of partitioning:

- When each m_i is a *certain* mass function, then M defines a conventional, crisp partition of the set of objects; this corresponds to a situation of complete knowledge;

- When each m_i is a Bayesian mass function, then M specifies a fuzzy partition;
- When the focal elements of all mass functions are restricted to be singletons of Ω or the empty set, a partition with a noise cluster as in the NC algorithm is recovered.

ECM is one of the algorithms proposed to derive a credal partition from data. Let m_{ij} denote the degree of belief that object $\mathbf{x}_i$ belongs to the subset $A_j \subseteq \Omega$. Deriving a credal partition implies determining, for each object $\mathbf{x}_i$, the quantities $m_{ij} = m_i(A_j)\ \forall A_j \neq \emptyset, A_j \subseteq \Omega$ in such a way that a low (respectively, high) value of m_{ij} is found when the distance d_{ij} between $\mathbf{x}_i$ and A_j is high (respectively, low). The distance d_{ij} between an object and a set of classes A_j is defined as follows. Like in fuzzy partitioning, each class ω_l is represented by a center $\mathbf{v}_l \in \mathbb{R}^p$. Then, for each subset $A_j \subseteq \Omega$, $A_j \neq \emptyset$, a centroid $\overline{\mathbf{v}}_j$ is calculated as the barycenter of the centers associated to the classes in A_j:

$$\overline{\mathbf{v}}_j = \frac{1}{|A_j|} \sum_{l=1}^{c} s_{lj} \mathbf{v}_l, \tag{11}$$

with

$$s_{lj} = \begin{cases} 1 & \text{if } \omega_l \in A_j, \\ 0 & \text{else.} \end{cases} \tag{12}$$

The distance d_{ij} between $\mathbf{x}_i$ and the focal set A_j may then be defined by:

$$d_{ij} = ||\mathbf{x}_i - \overline{\mathbf{v}}_j||. \tag{13}$$

The ECM algorithm searches for the M and V matrices that minimize a criterion similar to that of the NC algorithm:

$$J_{\text{ECM}}(M, V) = \frac{1}{2^c n} \left[\sum_{i=1}^{n} \sum_{A_k \neq \emptyset} |A_k|^{\alpha} m_{ik}^{\beta} d_{ik}^2 + \sum_{i=1}^{n} \rho^2 m_{i\emptyset}^{\beta} \right], \tag{14}$$

subject to the constraints $m_{ik} \geq 0$ for all i and k, $m_{i\emptyset} \geq 0$ for all i, and

$$\sum_{k/A_k \subseteq \Omega, A_k \neq \emptyset} m_{ik} + m_{i\emptyset} = 1 \quad \forall i = 1, n, \tag{15}$$

where $m_{i\emptyset}$ denotes the mass of the object $\mathbf{x}_i$ allocated to the empty set. Note that we use a normalized version of J_{ECM} (we divide J_{ECM} by the number of unknowns) which renders the criterion less sensible to the input configurations (number of classes and number of objects). This normalization does not change the update equations of M and V. Due to constraint (15), a large mass is allocated to the empty set when all other masses are small or, equivalently, when the object is far from all subsets A_j. The empty set can thus be interpreted as a "noise cluster" allowing the detection of outliers. Parameter ρ represents the distance of any object to the empty set. An additional weighting coefficient

$|A_k|^\alpha$ is introduced to penalize the allocation of belief to subsets with high cardinality; the exponent α allows us to control the degree of penalization.

As in FCM or NC, the credal partition is found by performing an iterative optimization with the alternate update of the masses and the centroids. The necessary condition of optimality for M gives direct update equations which are very similar to those of the NC algorithm except that there are 2^c values m_{ij} to compute instead of $c+1$ fuzzy membership degrees u_{ik}. A more complex update rule is found for the centroids, since the optimality conditions lead to the resolution of a linear system at each step of the optimization process: each column of V is the solution of a linear system of c equations and c unknowns. More details about these update equations can be found in Appendix A. As FCM and its variants, the algorithm starts with an initial guess for either the credal partition M or the cluster centers V and iterates until convergence, alternating the optimization of M and V.

As underlined in [22], a credal partition is a rich representation that carries a lot of information about the data. In [22], various tools helping the user to interpret the results of ECM were suggested. First, a credal partition can be converted into classical clustering structures. For example, a fuzzy partition can be recovered by computing the pignistic probability $BetP_i(\{\omega_k\})$ induced by each mass function m_i using (7) and interpreting this value as the degree of membership of object i to cluster k.

Another interesting way of synthesizing the information is to assign each object to the subset of classes with the highest mass. In this way, one obtains a partition in at most 2^c groups, which is referred to as a *hard credal partition*. This hard credal partition allows us to detect, on the one hand, the objects that can be assigned without ambiguity to a single cluster and, on the other hand, the objects lying at the boundary of two or more clusters.

3. ECM with constraints

As indicated in the introduction, the constraints that we consider in this paper are must-link and cannot-link constraints, which concern object pairs. A must-link constraint is used to specify that two objects should be associated with the same cluster. A cannot-link constraint is used to specify that two objects should not be associated with the same cluster. In this section, we show how to translate such pairwise constraints in the belief functions framework and how to integrate them in the search for a credal partition.

3.1. Expression of the constraints

Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be two objects associated with mass functions m_i and m_j. A mass function regarding the joint class membership of both objects may be computed from m_i and m_j in the Cartesian product $\Omega^2 = \Omega \times \Omega$. This mass function, denoted $m_{i\times j}$, is the combination of the vacuous extensions of m_i and

Table 2: Credal partition to express constraints

A	$m_1(A)$	$m_2(A)$	$m_3(A)$	$m_4(A)$
$\emptyset$	0	0	0	0
$\{\omega_1\}$	1	1	0	0
$\{\omega_2\}$	0	0	1	0
Ω	0	0	0	1

m_j [26]. As shown in [11], it can be written as:

$$m_{i\times j}(A\times B) = m_i(A)\ m_j(B) \quad A,B\subseteq\Omega, A\neq\emptyset, B\neq\emptyset, \tag{16}$$

$$m_{i\times j}(\emptyset) = m_i(\emptyset)+m_j(\emptyset)-m_i(\emptyset)\ m_j(\emptyset). \tag{17}$$

From $m_{i\times j}$, we can compute the plausibility that objects $\mathbf{x}_i$ and $\mathbf{x}_j$ belong or not to the same class. In Ω^2, the event "Objects $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class" corresponds to the subset $\theta = \{(\omega_1,\omega_1),(\omega_2,\omega_2),\dots(\omega_c,\omega_c)\}$, whereas the event "Objects $\mathbf{x}_i$ and $\mathbf{x}_j$ do not belong to the same class" corresponds to its complement $\overline{\theta}$. The corresponding plausibilities are the following:

$$\begin{aligned} pl_{i\times j}(\theta) &= \sum_{\{A\times B\subseteq\Omega^2 \mid (A\times B)\cap\theta\neq\emptyset\}} m_{i\times j}(A\times B) \\ &= \sum_{A\cap B\neq\emptyset} m_i(A)\ m_j(B), \end{aligned} \tag{18}$$

and

$$\begin{aligned} pl_{i\times j}(\overline{\theta}) &= 1-m_{i\times j}(\emptyset)-bel_{i\times j}(\theta) \\ &= 1-m_{i\times j}(\emptyset)-\sum_{\{A\times B\subseteq\Omega^2 \mid \emptyset\neq(A\times B)\subseteq\theta\}} m_{i\times j}(A\times B) \\ &= 1-m_{i\times j}(\emptyset)-\sum_{k=1}^{c} m_i(\{\omega_k\})\ m_j(\{\omega_k\}). \end{aligned} \tag{19}$$

Example 2. *Let us consider a new collection of four objects to be classified into two classes. A credal partition, which expresses some knowledge about the membership of the objects, is given in Table 2. The associated plausibilities $pl(\theta)$ and $pl(\overline{\theta})$ are given in Table 3.*

This simple example shows how the joint membership of two objects may be represented using the plausibilities $pl_{i\times j}(\theta)$ and $pl_{i\times j}(\overline{\theta})$. In simple terms, the relevant information in Table 3 is contained in the zeros of these plausibilities. For example, nothing can be said about the joint membership of object $\mathbf{x}_1$ and $\mathbf{x}_4$, as both of these plausibilities are equal to 1. On the contrary, the fact that $pl_{1\times 2}(\overline{\theta}) = 0$ indicates that $\mathbf{x}_1$ and $\mathbf{x}_2$ are certainly in the same cluster. Equivalently, the null value of the plausibility $pl_{1\times 3}(\theta)$ express the impossibility that $\mathbf{x}_1$ and $\mathbf{x}_3$ belong to the same class.

Table 3: Plausibilities for the events θ and $\overline{\theta}$

F	$pl_{1\times 2}(F)$	$pl_{1\times 3}(F)$	$pl_{1\times 4}(F)$
θ	1	0	1
$\overline{\theta}$	0	1	1

3.2. Objective function of CECM

Let us now assume that the credal partition is unknown and that we are given some pairwise constraints. As explained in the introduction, we assume that these constraints are must-link or cannot-link constraints. Let $\mathcal{M}$ denote the set of pairs of objects constrained by a must-link and $\mathcal{C}$ the set of pairs of objects constrained by a cannot-link. One has to seek for a credal partition that reflects both the similarities computed from the data and the constraints. A natural requirement is that $pl_{i\times j}(\theta)$ be as low as possible if $(\mathbf{x}_i, \mathbf{x}_i) \in \mathcal{C}$. In the same way, $(\mathbf{x}_i, \mathbf{x}_i) \in \mathcal{M}$ implies that $pl_{i\times j}(\overline{\theta})$ should be as low as possible. To achieve this goal, we suggest integrating a penalty term into the ECM criterion as follows. Let J_{CONST} denote the cost of violating the must-link and the cannot-link constraints, defined by:

$$J_{\text{CONST}} = \frac{1}{|\mathcal{M}| + |\mathcal{C}|} \left[\sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} pl_{i\times j}(\overline{\theta}) + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}} pl_{i\times j}(\theta) \right]. \tag{20}$$

We propose to minimize the following objective function:

$$J_{\text{CECM}}(M, V) = (1 - \xi) J_{\text{ECM}}(M, V) + \xi J_{\text{CONST}}. \tag{21}$$

such that the constraints (15) are respected. Parameter $\xi \in [0, 1]$ is used to control the balance between the constraints and the geometrical model.

3.3. Optimization

As in FCM, NC and ECM, we propose an alternate optimization scheme in order to fix the partition matrix M and the centroid matrix V. First, we note that the penalty term added to the objective function of ECM does not depend on the cluster centroids. The same update scheme for the centroids (equations (A.3) to (A.5) of Appendix A) can thus be used in CECM.

Generally, the problem is much more complex for the belief masses, and a direct update equation of the m_{ij} from the optimality conditions is no longer possible. However, if we fix $\beta = 2$, then the objective function (21) becomes quadratic with respect to the m_{ij}. As the constraints are linear, a standard quadratic programming (QP) algorithm can be used and convergence is insured in a reasonable time. In the experiments reported in Section 5, we have used a Matlab implementation of the projective QP method developed in [31].

The overall procedure is summarized in Algorithm 1.

Algorithm 1 CECM algorithm with Euclidean metric

Input: Number c of desired clusters, n objects $\mathbf{x}_1, ..., \mathbf{x}_n$, set of cannot-link $\mathcal{C}$, set of must-link $\mathcal{M}$
Output: Credal partition matrix M, centroid matrix V
Random Initialization of V
repeat
1) Calculate the new masses by solving the quadratic programming problem defined by (21) subject to (15).
2) Calculate the new centroids by solving the linear system defined by equations (A.3) to (A.5).
until No significant change in V between two successive iterations

4. CECM with an adaptive metric

In the ECM algorithm, the distance d_{ik}^2 between an object $\mathbf{x}_i$ and the centroid $\overline{\mathbf{v}}_k$ is the Euclidean distance. Classes are thus supposed to be spherical. However, the use of a Mahalanobis distance may be interesting in case of elliptical clusters. Using an adaptive metric can be highly desirable when using constraints, in particular when these constraints contradict a Euclidean model. A variant of the ECM model with an adaptive Euclidean distance will first be introduced in Subsection 4.1, and the optimization of the corresponding cost function will be detailed in Subsection 4.2.

4.1. Model

To allow metrics to be adapted in the ECM algorithm, we follow an approach inspired from Gustafson and Kessel [16] and well described in [17]. Let S_l denote a $(p \times p)$ matrix associated to cluster ω_l $(l = 1, c)$ inducing a norm $||\mathbf{x}||_{S_l}^2 = \mathbf{x}^t S_l \mathbf{x}$. Using the same approach that we used for the centroids, we compute the matrix $\overline{S}_j$ associated with a non singleton A_j by averaging the matrices associated to the classes $\omega_k \in A_j$:

$$\overline{S}_j = \frac{1}{|A_j|} \sum_{l=1}^{c} s_{lj} S_l, \quad \forall A_j \subseteq \Omega, A_j \neq \emptyset. \tag{22}$$

Matrix $\overline{S}_j$ may be seen as a kind of within-class covariance matrix of the clusters composing A_j; it thus describes the average shape of these clusters. The distance d_{ij}^2 between $\mathbf{x}_i$ and any set $A_j \neq \emptyset$ is then defined by:

$$d_{ij}^2 = ||\mathbf{x}_i - \overline{\mathbf{v}}_j||_{\overline{S}_j}^2 = (\mathbf{x}_i - \overline{\mathbf{v}}_j)^t \overline{S}_j (\mathbf{x}_i - \overline{\mathbf{v}}_j). \tag{23}$$

4.2. Optimization

We first note that the minimization of J_{CECM} with respect to the masses is independent of the metric, so that the way of deriving the masses by a constrained quadratic optimization is unchanged. In their algorithm, Gustafson

and Kessel showed that the update equations of FCM for the cluster centers were not affected by the introduction of a metric associated to each cluster. On the contrary, in CECM, the determination of the centers takes explicitly into account the metric, as shown below.

4.2.1. Optimization with respect to the cluster centers

We first consider that M and the matrices S_l $(l = 1, c)$ are fixed. The minimization of J_{CECM} with respect to V is an unconstrained optimization problem. The partial derivatives of J_{CECM} with respect to the centers are given by:

$$\frac{\partial J_{\text{CECM}}}{\partial \mathbf{v}_l} = \kappa \sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^2 \frac{\partial d_{ij}^2}{\partial \mathbf{v}_l} \quad l = 1, c, \tag{24}$$

where κ is a constant factor equal to $(1 - \xi)/2^c n$. We also have:

$$\frac{\partial d_{ij}^2}{\partial \mathbf{v}_l} = 2(s_{lj})\overline{S}_j(\mathbf{x}_i - \bar{\mathbf{v}}_j)(-\frac{1}{|A_j|}) \quad l = 1, c. \tag{25}$$

From (24) and (25), we thus have:

$$\frac{\partial J_{\text{CECM}}}{\partial \mathbf{v}_l} = -2\kappa \sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha-1} m_{ij}^2 s_{lj} \overline{S}_j(\mathbf{x}_i - \bar{\mathbf{v}}_j) \tag{26}$$

$$= -2\kappa \sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha-1} m_{ij}^2 s_{lj} \overline{S}_j(\mathbf{x}_i - \frac{1}{|A_j|} \sum_k s_{kj} \mathbf{v}_k) \quad l = 1, c. \tag{27}$$

Setting these derivatives to zero gives l equations in $\mathbf{v}_k$ which can be written as:

$$\sum_i \sum_{A_j \neq \emptyset} |A_j|^{\alpha-1} m_{ij}^2 s_{lj} \overline{S}_j \mathbf{x}_i = \sum_k \sum_i \sum_{A_j \neq \emptyset} |A_j|^{\alpha-2} m_{ij}^2 s_{lj} s_{kj} \overline{S}_j \mathbf{v}_k \quad l = 1, c, \tag{28}$$

or, equivalently:

$$\sum_i \sum_{A_j \ni \omega_l} |A_j|^{\alpha-1} m_{ij}^2 \overline{S}_j \mathbf{x}_i = \sum_k \sum_i \sum_{A_j \supseteq \{\omega_k, \omega_l\}} |A_j|^{\alpha-2} m_{ij}^2 \overline{S}_j \mathbf{v}_k \quad l = 1, c. \tag{29}$$

Let $\mathbf{F}^{(l,i)}$ denote the $(p \times p)$ matrix:

$$\mathbf{F}^{(l,i)} = \sum_{A_j \ni \omega_l} |A_j|^{\alpha-1} m_{ij}^2 \overline{S}_j \quad l = 1, c \quad i = 1, n, \tag{30}$$

and $\mathbf{G}^{(l,k)}$ denote the $(p \times p)$ matrix:

$$\mathbf{G}^{(l,k)} = \sum_i \sum_{A_j \supseteq \{\omega_k, \omega_l\}} |A_j|^{\alpha-2} m_{ij}^2 \overline{S}_j \quad k, l = 1, c. \tag{31}$$

Next, we form, from these two $(p \times p)$ matrices, two new matrices $\mathbf{F}$ and $\mathbf{G}$, of size $(cp \times np)$ and $(cp \times cp)$, respectively:

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}^{(1,1)} & \mathbf{F}^{(1,2)} & \dots & \mathbf{F}^{(1,n)} \\ \mathbf{F}^{(2,1)} & \mathbf{F}^{(2,2)} & \dots & \mathbf{F}^{(2,n)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}^{(c,1)} & \mathbf{F}^{(c,2)} & \dots & \mathbf{F}^{(c,n)} \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \mathbf{G}^{(1,1)} & \mathbf{G}^{(1,2)} & \dots & \mathbf{G}^{(1,c)} \\ \mathbf{G}^{(2,1)} & \mathbf{G}^{(2,2)} & \dots & \mathbf{G}^{(2,c)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}^{(c,1)} & \mathbf{G}^{(c,2)} & \dots & \mathbf{G}^{(c,c)} \end{pmatrix}. \tag{32}$$

Let us stack all objects $\mathbf{x}_i$ in a same vector $\mathbf{X}$ of size $(np \times 1)$ and rearrange matrix V in the form of a vector of size $(cp \times 1)$ such that:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{pmatrix}, \quad V = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_c \end{pmatrix}.$$

With all these notations, vector V is solution of the following linear system:

$$\mathbf{G}V = \mathbf{F}\mathbf{X}. \tag{33}$$

We can see that, instead of solving p system of c unknowns as in the case of a Euclidean metric, we have to solve a unique system of cp equations and cp unknowns. This higher complexity is the price to pay for an automatic adaptation of the metric.

4.2.2. Optimization with respect to the metrics S_l

We now consider that M and V are fixed and we want to determine the matrices S_l. We follow the same line of reasoning as Gustafson and Kessel. In order to avoid the degenerate solution consisting of matrices S_l with zero entries, we impose that the clusters have a constant volume using the constraints $\det(S_l) = 1$ for all $l = 1, c$. To solve the constrained minimization problem with respect to $S_1, \cdots, S_c$, we introduce c Lagrange multipliers λ_i and write the Lagrangian:

$$\mathcal{L}(S_1, \cdots, S_c, \lambda_1, \cdots, \lambda_c) = J_{\text{CECM}}(M, V) - \sum_{k=1}^{c} \lambda_k \left(\det(S_k) - 1\right). \tag{34}$$

We recall that the definition of the distance of an object $\mathbf{x}_i$ to a focal set A_j is:

$$d_{ij}^2 = (\mathbf{x}_i - \overline{\mathbf{v}}_j)^t \overline{S}_j (\mathbf{x}_i - \overline{\mathbf{v}}_j) = (\mathbf{x}_i - \overline{\mathbf{v}}_j)^t \left(\frac{1}{|A_j|} \sum_{k=1}^{c} s_{kj} S_k \right) (\mathbf{x}_i - \overline{\mathbf{v}}_j). \tag{35}$$

Starting from the fact that the derivatives of $\mathbf{x}^t A \mathbf{x}$ and $\det(A)$ with respect to a symmetric matrix A are $\mathbf{x}\mathbf{x}^t$ and $\det(A)A^{-1}$ respectively, we obtain the following derivative of $\mathcal{L}$ with respect to matrix S_l:

$$\frac{\partial \mathcal{L}}{\partial S_l} = \kappa \sum_i \sum_{A_j \neq \emptyset} m_{ij}^2 |A_j|^{\alpha - 1} s_{lj} (\mathbf{x}_i - \overline{\mathbf{v}}_j)(\mathbf{x}_i - \overline{\mathbf{v}}_j)^t - \lambda_l \det(S_l) S_l^{-1} \quad l = 1, c. \tag{36}$$

The derivatives with respect to the Lagrange multipliers lead to the constraints $\det(S_l) = 1$ for all i. Let Σ_l denote the following matrix:

$$\Sigma_l = \sum_i \sum_{A_j \ni \omega_l} m_{ij}^2 |A_j|^{\alpha-1} (\mathbf{x}_i - \overline{\mathbf{v}}_j)(\mathbf{x}_i - \overline{\mathbf{v}}_j)^t \quad l = 1, c. \tag{37}$$

Note that Σ_l can be considered as the analog in the evidential framework of the fuzzy covariance matrix. From (36), we have:

$$\kappa \Sigma_l = \lambda_l S_l^{-1} \quad l = 1, c, \tag{38}$$

and, thus

$$\kappa \Sigma_l S_l = \lambda_l I \quad l = 1, c, \tag{39}$$

where I denote the $(p \times p)$ identity matrix. Taking the determinant of this last equation leads to:

$$\det(\kappa \Sigma_l S_l) = \kappa^p \det(\Sigma_l) \det(S_l) = \det(\Sigma_l) = \lambda_l^p \quad l = 1, c. \tag{40}$$

It follows that

$$\lambda_l = \kappa \det(\Sigma_l)^{\frac{1}{p}} \quad l = 1, c. \tag{41}$$

Replacing λ_l by its expression and using (38), we finally obtain:

$$S_l = \det(\Sigma_l)^{\frac{1}{p}} \Sigma_l^{-1} \quad l = 1, c. \tag{42}$$

Note that Σ_l is invertible since it is symmetric and positive definite. Indeed, each $(\mathbf{x}_i - \overline{\mathbf{v}}_j)(\mathbf{x}_i - \overline{\mathbf{v}}_j)^t$ is symmetric, positive and semi-definite, and so is their weighted sum.

The overall CECM procedure with an adaptive metric is summarized in Algorithm 2.

Algorithm 2 CECM with an adaptive metric

Input: Number c of desired clusters, n objects $\mathbf{x}_1, ..., \mathbf{x}_n$, set of cannot-link $\mathcal{C}$, set of must-link $\mathcal{M}$

Output: Credal partition matrix M, centroid matrix V, set of matrices S_l $l = 1, c$

Random Initialization of V

repeat

1) Calculate the new masses by solving the quadratic programming problem defined by (21) subject to (15).

2) Calculate the new centroids by solving the linear system defined by equations (30) to (33).

3) Calculate the new matrices S_l, $l = 1, c$ using (37) and (42).

until No significant change in V between two successive iterations

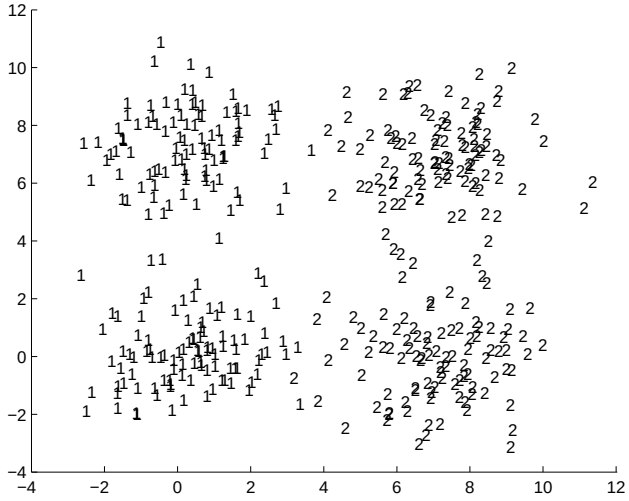

Figure 1: Two-class data set.

5. Evaluation of the proposed method

This section is devoted to the experimental validation of the algorithms introduced above. The experimental set-up (including data sets, performance criterion, alternative methods, constraint selection procedures and parameter setting) will first be described in Subsection 5.1. Results will then be presented and discussed in Subsection 5.2. Finally, the computational complexity issue will be addressed in Subsection 5.3.

5.1. Methodology

5.1.1. Datasets

The performances of CECM were evaluated on four real data sets from the UCI repository (http://www.ics.uci.edu/~mlearn). The main characteristics of these datasets are summarized in Table 4. For the Letters dataset , we kept only the three letters {I,J,L} as done in [2]. These classes were chosen since they are hard to discriminate. In order to illustrate the interest of introducing constraints, we created a simple synthetic dataset (Two-class dataset), represented in Figure 1. It consists of two classes in a two-dimensional space. In each class, patterns were generated according to a mixture of two Gaussians, with means $(0, 0)$ and $(0, 7)$ in the first class, and $(7, 0)$ and $(7, 7)$ in the second one. All the Gaussians have a common covariance matrix $2I$, where I denotes the identity matrix in $\mathbb{R}^2$. In the two classes, the proportions of the Gaussians are the same: 200 points were drawn from each.

Table 4: Data sets used in the experiments.

	Iris	Wine	Glass	Letters	Two-class
number of objects n	150	178	214	227	400
number of classes c	3	3	2	3	2
dimension p	4	13	9	16	2

5.1.2. Performance evaluation

In order to evaluate the accuracy of a clustering algorithm, the crisp partition $\widehat{P}$ found may be compared to a reference partition P. The Rand Index (RI) is the most common similarity measure between two partitions. It is defined as:

$$RI(P, \widehat{P}) = \frac{2(a+b)}{n(n-1)}, \tag{43}$$

where a (respectively, b) is the number of pairs of objects simultaneously assigned to identical classes (respectively, different classes) in P and $\widehat{P}$.

Remark that, with CECM, $\widehat{P}$ was determined by assigning each object to the cluster with maximal pignistic probability after convergence of the algorithm.

5.1.3. Comparison with reference methods

The performances of our algorithm were compared to those of three reference methods integrating must-link and cannot constraints:

- COP-KMEANS [28] is one of the first algorithms proposed for integrating background knowledge in clustering: it is a modification of the hard c-means algorithm which enforces in a hard way all constraints to be satisfied. Note that this algorithm may fail to converge.
- The Constrained Fuzzy C-means algorithm (CFCM) [15] is based on the FCM algorithm. As in CECM, the constraints are integrated as a penalty term in the objective function. Two versions have been implemented. The first one, CFCM-Eucl, is based on Euclidean distances; the second one, CFCM-Mah, uses an adaptive metric based on the fuzzy covariance matrices of the clusters. This algorithm is the closest to CECM.
- The distance metric learning approach (DML) [29] is not a clustering algorithm. It learns a distance metric over the input space that respects the relationships expressed by the pairwise constraints. Using this new metric, the objects can be clustered using any clustering algorithm. For the experiments, we used FCM.

5.1.4. Choice of the constraints

Constraints were defined using two different methods. Random selection consists in randomly selecting two patterns in the dataset. Then, the true

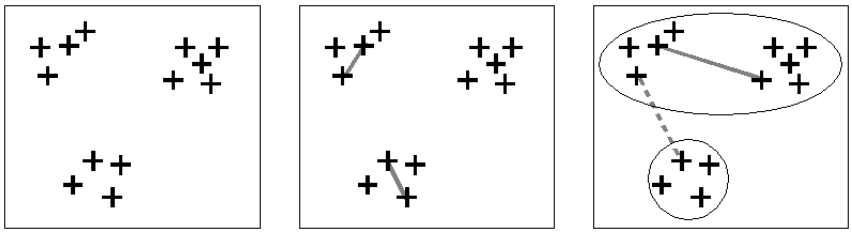

Figure 2: Pairwise constraints patterns for a dataset (a): some are useless (b) and some are informative (c) to lead the algorithm towards a desired solution.

relationship between these points is identified using the true partition of the data. This technique allows us to introduce a high number of constraints, and thus to study the behavior of the algorithm in various situations.

However, in some applications, the constraints may not always be available a priori, but an oracle (a user) may be available to provide the constraints. This scheme, where the system queries the oracle to obtain information, is called *active learning* [7]. Obviously, among the possible pairwise constraints, some of them are informative with respect to the clustering problem, while some of them are useless, as illustrated in Figure 2. Additionally, several authors have observed that a constrained clustering approach with a bad choice of pairs can deteriorate the clustering performances [10, 27]. The goal of active learning is thus to select the pairwise constraints which are the most informative about the underlying structure of the objects, so that the clustering performance can be improved with as few queries as possible.

We propose to introduce constraints incrementally by alternatively running CECM, selecting pairs of objects, and asking an expert to identify the nature of the corresponding constraints, until a specified number of constraints is reached. We use the credal partition obtained with CECM to select the most suitable pairs of objects. These pairs are selected according to the following requirements:

- The first object must be classified with a *high* degree of uncertainty;
- The second object must be classified with a *low* degree of uncertainty.

Indeed, if the uncertainty about the membership of the two objects is low, the constraint may be non informative and conversely, if the uncertainty regarding the classification of both objects is high, the constraint may lead to misclassify both objects.

Different ways of finding such objects thanks to the credal partition or to the centroids can be considered. We propose a strategy that proved experimentally to be efficient. The points for which the uncertainty is high are the points assigned in the hard credal partition (see Section 2.3) to focal sets of cardinality greater than 1. In particular, points associated to focal sets A_j such that $|A_j| =$

2 are likely to be located at the boundary between two clusters. Thus, for the selection of the first object, we propose to select the point associated to the highest mass allocated to focal sets of cardinality equal to 2. For the second object, we pick up the nearest point from one of the centers. The user is then provided with this pair of points, and enters either a must-link or a cannot-link constraint.

5.1.5. Guidelines for setting the parameters

We address here the choice of the parameters used to run the various experiments. In order to obtain a significant level of non-specificity, so that the credal partition computed differs from a fuzzy partition, parameter α was set to 1.

The values of parameter ρ differ according to the data processed. This parameter represents the fixed distance between each object and the noise cluster. It thus controls the number of objects considered as outliers. For each experiments reported here, we fixed ρ to a value greater than the maximum distance between two points, so that no rejection was considered. Note that, if rejection is needed, the user can decrease the value of ρ so as to achieve a given rejection rate.

For the choice of ξ in (21), which controls the compromise between the pairwise constraints and the fit of the geometrical model to the data, several experiments were carried out with various settings: data sets, metric, number of constraints, and value of ξ. Some of these experiments are reported in Tables 5 and 6. The results were averaged over 100 trials with random selection of the constraints. It turns out that the main factor influencing the clustering performances is the number of constraints. The choice of ξ, although impacting also the Rand Index, is not so critical. Results are very stable. We found experimentally that a value $\xi = 0.5$ generally yields acceptable results.

5.2. Results

5.2.1. Interest of adding constraints

We first illustrate the interest of introducing constraints using the synthetic Two-class data set. The ECM algorithm was run using the Euclidean distance, with $\rho^2 = 100$. The credal partition obtained shows a diagonal boundary between the two classes. The direction of the boundary (from upper left to lower right, or from lower left to upper right) depends on the initialization of the centroids. Figure 3 represents one of the hard credal partitions obtained. Here, each point is associated with the non-empty subset $A \subseteq \Omega$ that received the highest amount of belief mass. The two large crosses represent the centroids obtained after convergence. The RI is equal to 0.56.

The Euclidean distance implicitly assumes that the classes are spherical, which is obviously not the case for this dataset. If we use the Mahalanobis distance instead (with the same parameter values as before), we obtain either a horizontal or a vertical boundary between the classes. Figure 4 shows one of the solutions obtained, where the boundary is horizontal. In this case, the credal partition does not correspond to the true partition of the data, and the RI is equal to 0.5.

Table 5: Average Rand Index and standard deviation over 100 trials as a function of ξ for $C = 20$ and $C = 50$ randomly chosen constraints.

C	ξ	Iris (Mah.)	Wine (Eucl.)	Glass (Mah.)
20	0	0.87 ± 0.00	0.95 ± 0.00	0.85 ± 0.00
	0.1	0.93 ± 0.02	0.95 ± 0.01	0.87 ± 0.03
	0.2	0.94 ± 0.03	0.95 ± 0.01	0.86 ± 0.03
	0.3	0.94 ± 0.03	0.95 ± 0.01	0.86 ± 0.04
	0.4	0.94 ± 0.03	0.95 ± 0.01	0.87 ± 0.03
	0.5	0.94 ± 0.03	0.95 ± 0.01	0.87 ± 0.03
	0.6	0.94 ± 0.02	0.95 ± 0.01	0.87 ± 0.03
	0.7	0.94 ± 0.03	0.95 ± 0.01	0.86 ± 0.04
	0.8	0.94 ± 0.03	0.95 ± 0.01	0.86 ± 0.03
	0.9	0.94 ± 0.03	0.95 ± 0.01	0.87 ± 0.03
50	0	0.87 ± 0.00	0.95 ± 0.00	0.85 ± 0.00
	0.1	0.96 ± 0.02	0.95 ± 0.01	0.89 ± 0.02
	0.2	0.96 ± 0.02	0.96 ± 0.01	0.89 ± 0.02
	0.3	0.96 ± 0.02	0.96 ± 0.01	0.89 ± 0.03
	0.4	0.96 ± 0.02	0.96 ± 0.01	0.90 ± 0.03
	0.5	0.96 ± 0.02	0.96 ± 0.01	0.90 ± 0.03
	0.6	0.96 ± 0.02	0.96 ± 0.01	0.90 ± 0.03
	0.7	0.96 ± 0.02	0.96 ± 0.01	0.89 ± 0.03
	0.8	0.96 ± 0.02	0.96 ± 0.01	0.89 ± 0.05
	0.9	0.96 ± 0.02	0.96 ± 0.01	0.90 ± 0.03

Table 6: Average Rand Index and standard deviation over 100 trials as a function of ξ for $C = 100$ and $C = 200$ randomly chosen constraints.

C	ξ	Iris (Mah.)	Wine (Eucl.)	Glass (Mah.)
100	0	0.87 ± 0.00	0.95 ± 0.00	0.85 ± 0.00
	0.1	0.97 ± 0.02	0.96 ± 0.01	0.92 ± 0.02
	0.2	0.97 ± 0.02	0.97 ± 0.01	0.92 ± 0.02
	0.3	0.97 ± 0.02	0.97 ± 0.01	0.92 ± 0.02
	0.4	0.97 ± 0.02	0.97 ± 0.01	0.93 ± 0.02
	0.5	0.97 ± 0.02	0.98 ± 0.01	0.93 ± 0.02
	0.6	0.97 ± 0.02	0.98 ± 0.01	0.94 ± 0.02
	0.7	0.97 ± 0.02	0.98 ± 0.01	0.93 ± 0.02
	0.8	0.97 ± 0.02	0.98 ± 0.01	0.93 ± 0.02
	0.9	0.97 ± 0.02	0.98 ± 0.01	0.93 ± 0.02
200	0	0.87 ± 0.00	0.95 ± 0.00	0.85 ± 0.00
	0.1	0.99 ± 0.01	0.97 ± 0.01	0.94 ± 0.01
	0.2	0.99 ± 0.01	0.98 ± 0.01	0.96 ± 0.02
	0.3	0.99 ± 0.01	0.98 ± 0.01	0.96 ± 0.01
	0.4	0.99 ± 0.01	0.99 ± 0.01	0.96 ± 0.02
	0.5	0.99 ± 0.01	0.99 ± 0.01	0.97 ± 0.02
	0.6	0.99 ± 0.01	0.99 ± 0.01	0.97 ± 0.02
	0.7	0.99 ± 0.01	0.99 ± 0.01	0.97 ± 0.02
	0.8	0.99 ± 0.01	0.99 ± 0.01	0.97 ± 0.02
	0.9	0.98 ± 0.02	0.99 ± 0.01	0.97 ± 0.02

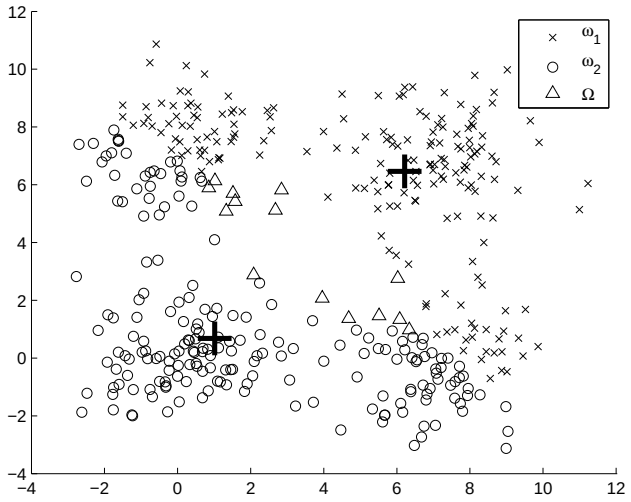

Figure 3: Hard credal partition obtained using ECM with Euclidean metric.

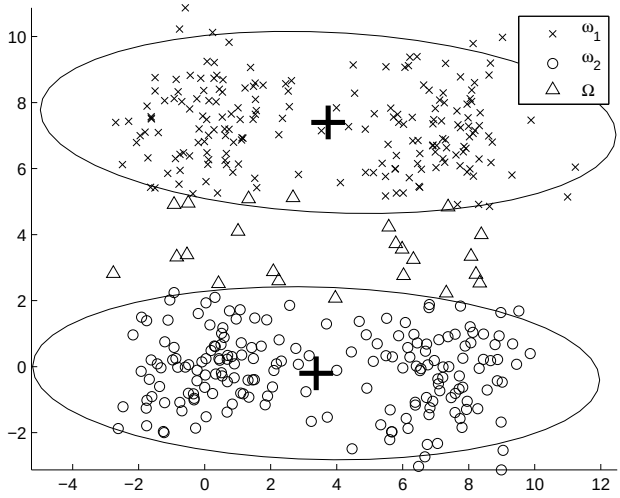

Figure 4: Hard credal partition obtained using ECM with an adaptive metric.

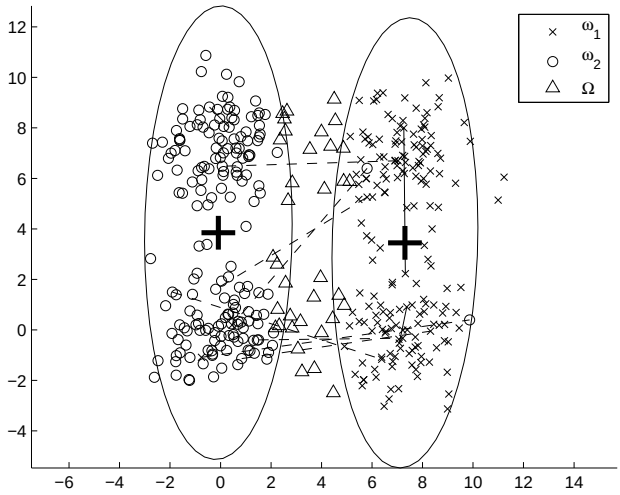

Figure 5: Hard credal partition obtained using CECM with adaptive metric and 10 constraints. Solid and dashed lines represent must-link and cannot-link constraints, respectively.

The addition of a small number of randomly chosen constraints allows us to guide the algorithm towards the desired solution. For example, by using only ten constraints, CECM finds the desired classes, as shown in Figure 5. Here, a solid line segment between two points corresponds to a must-link constraint between two objects and a dashed line segment between two points corresponds to a cannot-link constraint.

5.2.2. Random constraint selection vs. active learning

Here, we examine the behavior of the algorithm and make a comparison between an active learning scheme and a random constraint selection method. We do not report all the results.

Figure 6 shows the evolution of the average RI (computed over 100 trials) according to the number of pairwise constraints, for the Iris data set, chosen for illustration, together with the 95% confidence interval. We used parameter values $\xi = 0.5$ and $\rho^2 = 1000$. The pairwise constraints were randomly selected. The average RI was computed both over all objects and over unconstrained objects.

We remark that the RI computed over unconstrained objects increases with the number of constraints. Therefore, introducing constraints does not only improve the classification of constrained objects, but also allows us to guide the algorithm towards a better solution. This kind of behavior was generally observed with a majority of datasets.

In some cases, the RI computed over constrained objects may decrease with

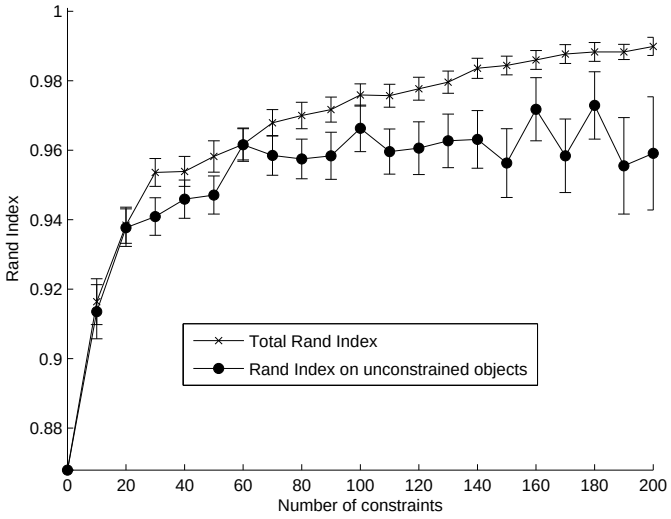

Figure 6: Averaged Rand Index and 95% confidence interval as a function of the number of randomly selected constraints (Iris data set).

the number of constraints. The reason is that a constraint involving a data point misclassified with a high degree of belief may have a negative effect on the clustering. Indeed, in this case, the class centers may move in undesired directions, and the other constrained point, previously well classified, may switch to the wrong class.

Active learning, being a way of introducing constraints on carefully selected points, seems a good way of avoiding such a situation. Figures 7 to 10 compare the active learning scheme with random constraint selection. Note that active learning does not involve any random selection of constraints, hence the absence of confidence intervals. Overall, active learning allows faster convergence than does random selection. In the case of the Iris dataset, the optimum is obtained with 40 constraints when using active learning, whereas it is still not obtained with 200 constraints using random selection. Remark that active learning may be outperformed by random selection (Figures 9 and 10), especially with a small number of constraints. In this case, active learning tends to introduce constraints on data that belong to specific regions of the input space. This may result in undesired moves of the class centers. As a consequence, other data points whose distances to the center increase may switch to other classes.

5.2.3. Performance comparison

Performance comparisons were carried out using the four real data sets. For each data set, six algorithms were compared using a varying number of constraints between 0 and 200: CECM and CFCM with a Mahalanobis or an Eu-

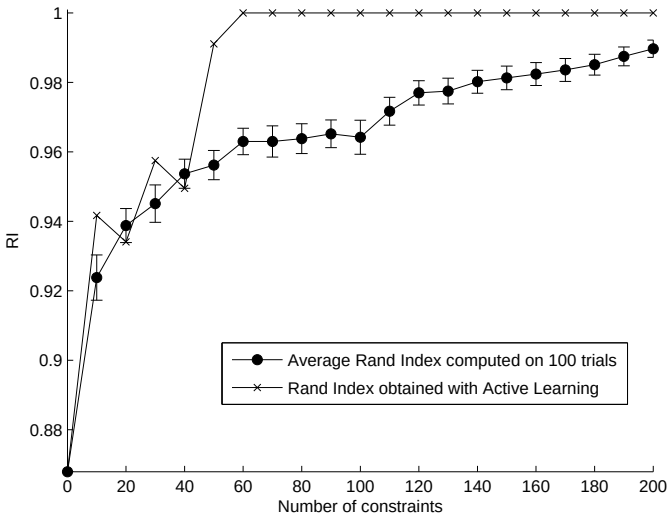

Figure 7: Rand Index obtained using Active learning, and average Rand Index and 95% confidence interval obtained using randomly selected constraints, as a function of the number of constraints (Iris data set).

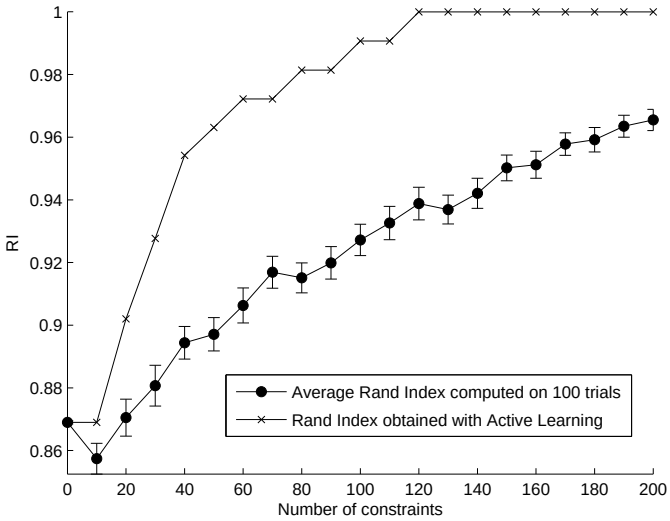

Figure 8: Rand Index obtained using Active learning, and average Rand Index and 95% confidence interval obtained using randomly selected constraints, as a function of the number of constraints (Glass data set).

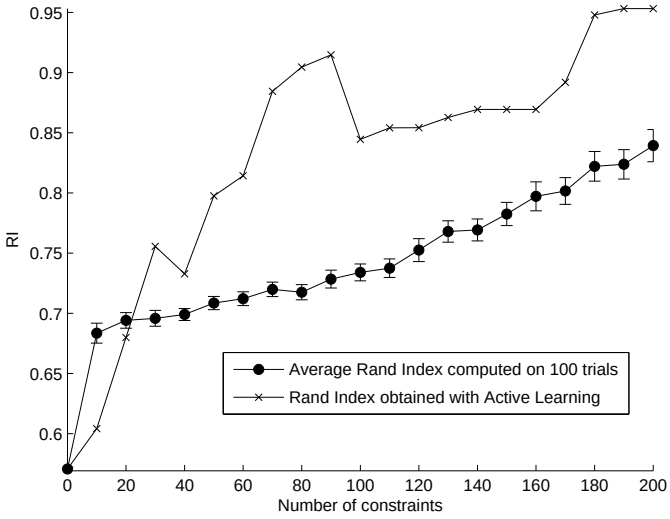

Figure 9: Rand Index obtained using Active learning, and average Rand Index and 95% confidence interval obtained using randomly selected constraints, as a function of the number of constraints (Letters data set).

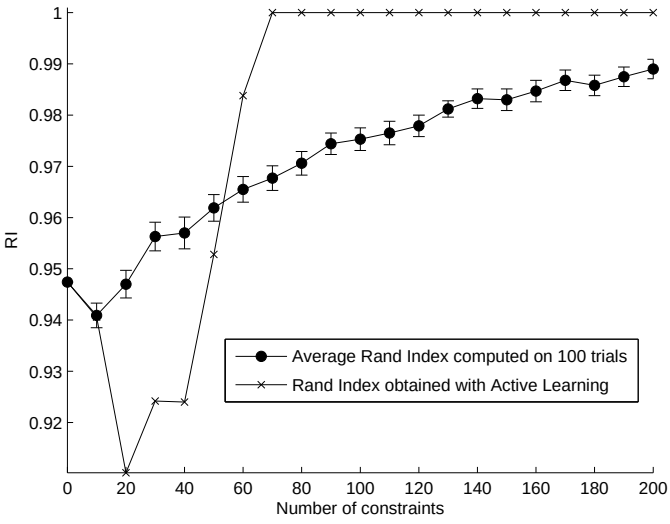

Figure 10: Rand Index obtained using Active learning, and average Rand Index and 95% confidence interval obtained using randomly selected constraints, as a function of the number of constraints (Wine data set).

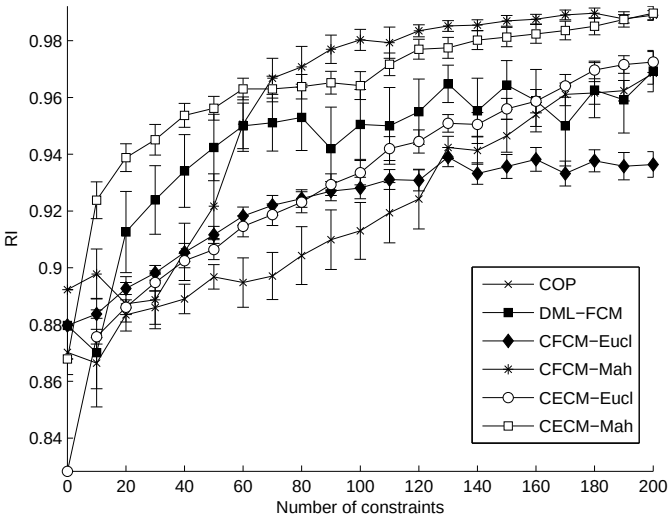

Figure 11: Results obtained by different clustering algorithms on the Iris data set.

clidean distance (CECM-Mah, CECM-Eucl, CFCM-Mah, CFCM-Eucl), COP-kmeans (COP) and Xing's approach followed by FCM (DML-FCM).

The results are presented in Figures 11 to 14. Each of these figures displays the mean RI and its 95% confidence interval computed over 100 trials with random constraint selection. A first remark is that the best results are obtained by CECM and CFCM. On the glass data set, CECM with a Mahalanobis distance outperforms CFCM whatever the number of constraints. On the Iris and Letters data sets, with the Mahalanobis distance for both algorithms, the RI is better for CFCM when the number of constraints is high. However, interestingly, in case of a small number of constraints, CECM always yields the best results in terms of classification accuracy. This is an interesting feature of CECM, since obtaining constraints may be a hard or expensive task. On the Wine data set, the Euclidean distance is the most suitable and the two algorithms provide similar results.

5.2.4. Application to medical image segmentation

The interest of CECM will now be illustrated using an example in medical imaging taken from [6]. An image of a pathological brain was acquired using magnetic resonance imaging (see Figure 15). In this image, three main areas may be distinguished: the brightest area corresponds to a pathological area, the dark gray to normal brain tissues and intermediate gray levels correspond to ventricles and cerebrospinal fluid. The aim was to isolate the pathological area from the other parts of the brain by looking for a partition into $c = 2$ clusters.

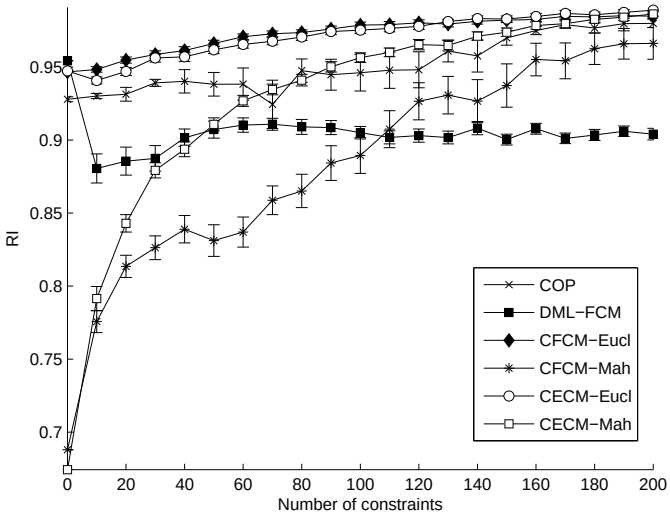

Figure 12: Results obtained by different clustering algorithms on the Wine data set.

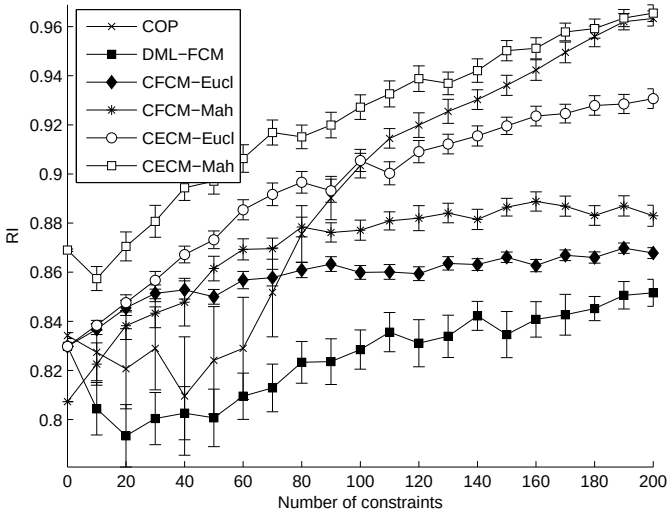

Figure 13: Results obtained by different clustering algorithms on the Glass data set.

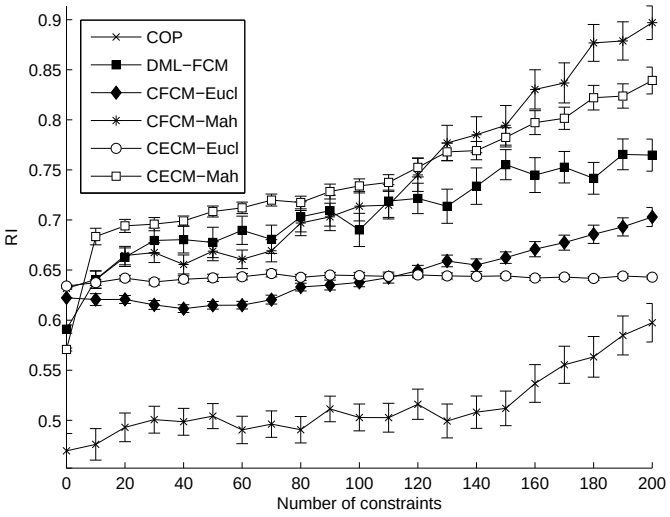

Figure 14: Results obtained by different clustering algorithms on the Letters data set.

To make the computations tractable, the gray levels of the 156×141 pixels of the original image were quantified in 400 prototypes using a basic learning vector quantization algorithm [18]. The clustering was performed on this set of prototypes and the pixels in the image were assigned to the class of the nearest prototype.

Starting from the gray levels of the pixels (rescaled between 0 and 1), ECM, with $c = 2$, $\alpha = 2$, and $\rho^2 = 10$, finds the hard credal partition represented in Figure 16. White and light grays represent two clusters and the darker gray is used for pixels assigned to Ω in the hard credal partition.

In a next experiment, imitating what could be done by an expert, we introduced constraints as indicated in Figure 17. White areas correspond to pixels related by a must-link constraint and these two areas are mutually linked by a cannot-link constraint. The hard credal partition obtained by applying CECM with the adaptive metric (with $\xi = 0.5$ and $\alpha = 2$, $\rho^2 = 10$) is shown in Figure 18. It may be seen that the constraints made it possible to remove the ambiguity concerning the pixels allocated to Ω and thus to properly isolate the pathological area. As a matter of comparison, the partitions computed from the pignistic probabilities obtained by ECM and CECM are given in Figure 19.

5.3. Computational complexity

As discussed in [22], the number of parameters in a credal partition is exponential in the number of clusters and linear in the number of objects. At each step of the algorithm CECM, we have to solve a quadratic programming

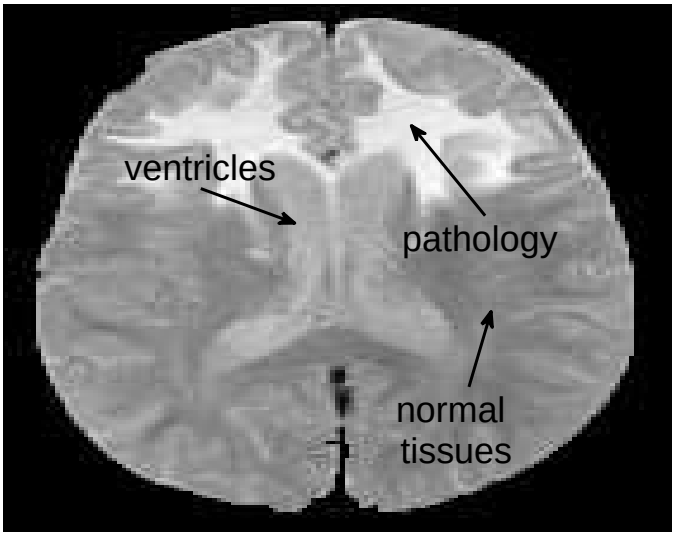

Figure 15: Original image of the brain. The bright, dark gray and intermediate gray correspond, respectively, to the pathology, normal brain tissues, and ventricles and cerebrospinal fluid.

Figure 16: Hard credal partition obtained by ECM with a Euclidean metric (white: ω_1, light gray: ω_2, dark gray: Ω).

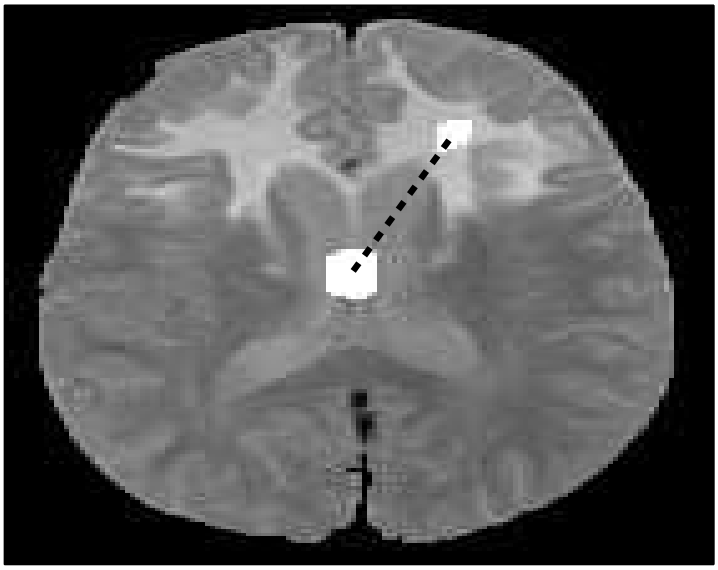

Figure 17: Must-link constraints (white areas) and cannot-link constraint (dashed line) introduced by an expert.

Figure 18: Hard credal partition obtained from CECM with an adaptive metric (white: ω_1, light gray: ω_2, dark gray: Ω).

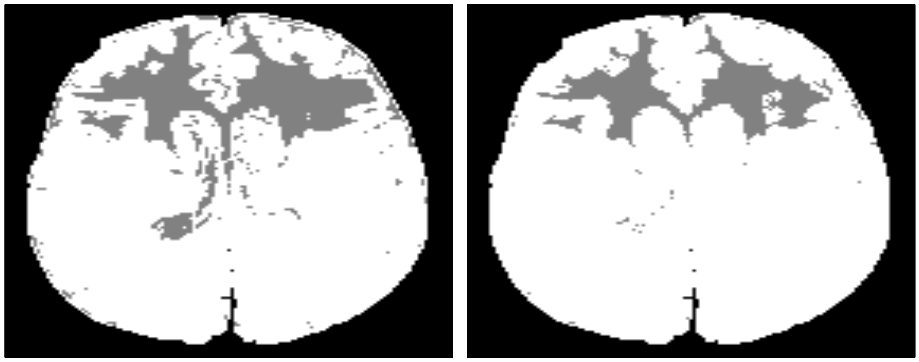

Figure 19: Partitions computed from the pignistic probabilities obtained with ECM (left) and CECM (right).

Table 7: Mean CPU times (in seconds) and standard deviation over 20 trials with 100 randomly chosen constraints.

Data	CFCM-Eucl	CFCM-Mah	CECM-Eucl	CECM-Mah
Iris	5.2 ± 1.31	6.28 ± 1.78	2.74 ± 0.61	5.26 ± 1.17
Wine	6.47 ± 1.37	15.55 ± 4.91	4.36 ± 0.51	9.75 ± 2.73
Glass	0.69 ± 0.05	0.99 ± 0.08	2.5 ± 0.32	5.68 ± 1.98
Letters	79.4 ± 45.24	63.07 ± 27.58	82.11 ± 29.42	37.46 ± 10.5

problem for the masses and a linear system for the centroids. As a consequence, our approach remains thus limited to applications of moderate size (say, less than 10 classes and a few hundred objects).

We performed an experimental comparison between the computing times of CECM and CFCM, the two algorithms that yielded the best classification accuracy in the previous experiments. All algorithms were implemented in Matlab and were run on a PC with a Dual Core AMD Opteron processor 885 and 32 Go of RAM. The tests were conducted using the four real data sets and a fixed number of constraints equal to 100. The CPU times and numbers of iterations, averaged over 20 trials, are shown in Tables 7 and 8, respectively. It can be seen that calculations with CECM are easily tractable for the four data sets used in the experiments. CECM is often faster than CFCM because of a smaller number of iterations to reach the solution, as shown in Table 8.

Furthermore, as for ECM, it is possible to reduce the complexity of CECM by considering only a subclass of mass functions with a limited number of focal sets. For example, we may constrain the focal sets to be either Ω or subsets composed of at most two classes. By this way, the number of parameters is reduced from $2^c n$ to $c^2 n$ and an acceptable tradeoff between flexibility of the method and computational tractability is achieved. As an illustration, let us consider the

Table 8: Average and standard deviation of the number of iterations over 20 trials with 100 randomly chosen constraints.

Data	CFCM-Eucl	CFCM-Mah	CECM-Eucl	CECM-Mah
Iris	25.23 ± 5.55	28.34 ± 8.02	5 ± 2	10.2 ± 3.02
Wine	26.1 ± 3.52	54.54 ± 15.87	5.4 ± 1.1	12.95 ± 4.71
Glass	42.3 ± 3.23	48.76 ± 4.31	5.65 ± 1.35	16 ± 8.01
Letters	212.41 ± 122.13	161.3 ± 70.83	64.1 ± 25.05	29.85 ± 10.48

Table 9: Comparison between the full version (CECM-1) and the limited version (CECM-2) of the CECM algorithm applied to the Two-Class dataset with $C = 4$. The average and standard deviation of the CPU time and the number of iterations were computed over 20 trials with 100 randomly chosen constraints.

	CECM-1 Eucl.	CECM-2 Eucl.	CECM-1 Mah.	CECM-2 Mah.
CPU (sec.)	141.27 ± 52.08	70.14 ± 25.52	144.79 ± 41.20	76.03 ± 29.15
Nb. iter.	27.85 ± 11.65	22 ± 9.15	21.95 ± 7.16332	21.05 ± 9.02

Two-class data set and let us search for a partition into $c = 4$ classes. We have compared the full version of CECM (CECM-1) (2^4 focal element by object) to a limited version, CECM-2, with Ω and subsets composed of at most two classes. Both versions gave a RI equal to 0.99. The results (CPU time and number of iterations), averaged over 20 trials, are shown in Table 9. It can be seen that a significant reduction of computing time was obtained using the constrained version without sacrificing the clustering accuracy.

6. Conclusion

In this paper, we addressed the problem of introducing constraints in a classification task. Our work is based within the theoretical framework of belief functions. In this framework, the ECM algorithm computes a credal partition of the data: each pattern is associated with a belief function that describes its membership to the classes. Our contribution is twofold. We introduced the Mahalanobis distance in the ECM algorithm, in order to handle non-spherical classes. We also presented an extension of the ECM algorithm, called CECM, which takes additional information into account in the clustering process. This information takes the form of pairwise constraints: a must-link constraint indicates that two patterns must be classified into the same class; a cannot-link constraint, that they must be classified into different classes. We also proposed an active-learning procedure, in which an expert is questioned about the relationships between pairs of data. Selecting these pairs is obviously a crucial issue for introducing relevant constraints. In our algorithm, the selection step may be easily conducted using the semantics of belief functions.

Our experiments show that introducing constraints improves the accuracy of the partition obtained, by guiding the algorithm towards desired solutions. When complex models are used, such as the Mahalanobis metric for computing distances between data, constraints allow us to compute parameter estimates that better fit the problem considered. We also showed that the number of constraints required to obtain an accurate clustering of the data does not need to very large. In particular, much fewer constraints were necessary to reach the optimal partition when using our active-learning procedure than when constraints were randomly chosen.

The performances of CECM were also compared to those of several other algorithms. It turns out that CECM yields the best results in most experiments, especially when the number of constraints is low. Finally, we demonstrated the interest of our approach on a medical image segmentation problem. The aim was to process the image of a pathological brain in order to detect a tumor. The mere application of the ECM algorithm did not lead to a satisfactory solution, as several parts of the image are associated with a high degree of indetermination. However, introducing a few constraints made it possible to clear up the ambiguity between pathological and healthy cells and to provide an accurate segmentation of the image.

This research may be extended in several directions. Some authors [20] proposed to add soft constraints rather than crisp ones. A soft constraint may be seen as a relationship between two objects, accompanied with a degree of certainty that this relationship holds. The interest of adding such constraints is twofold. First, one may hope to reduce the negative effect of a small set of constraints on the accuracy of the clustering. Furthermore, the problem of the consistency between the constraints themselves may be tackled to some extent. Introducing soft constraints in ECM is thus an interesting perspective.

Finally, we intend to further study the application of our method to complex real-world applications where background knowledge can be provided by experts. In particular, our active-learning scheme seems particularly promising for medical image segmentation problems. Indeed, a physician may easily label parts of an image as homogeneous regions, or instead require that two regions be classified into different classes.

Acknowledgements

The authors wish to express their thanks to Prof. Catherine Adamsbaum (Hôpital St Vincent de Paul, Paris, France) and Prof. Isabelle Bloch (École Nationale Supérieure des Télécommunications, Paris, France) for providing the brain image.

References

[1] S. Basu, A. Banerjee, and R.J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM International Conference on Data Mining*, pages 333–344, Lake Buena Vista, FL, USA, 2004.

[2] S. Basu, M. Bilenko, A. Banerjee, and R.J. Mooney. Probabilistic semi-supervised clustering with constraints. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 71–98. MIT Press, 2006.

[3] C. Baudrit and D. Dubois. Practical representations of incomplete probabilistic knowledge. *Computational Statistics & Data Analysis*, 51(1):86 – 108, 2006. The Fuzzy Approach to Statistical Analysis.

[4] I. Berget, B. Mevik, and T. Næs. New modifications and applications of fuzzy c-means methodology. *Computational Statistics & Data Analysis*, 52(5):2403 – 2418, 2008.

[5] J. C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, New York, 1981.

[6] I. Bloch. Defining belief functions using mathematical morphology: Application to image fusion under imprecision. *International Journal of Approximate Reasoning*, 48:437–465, 2008.

[7] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[8] R.N. Davé. Clustering relational data containing noise and outliers. *Pattern Recognition Letters*, 12:657–664, 1991.

[9] I. Davidson and S. S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. In *Proceedings of the Fifth SIAM International Conference on Data Mining*, page 138, Newport Beach, CA, USA, 2005. Society for Industrial Mathematics.

[10] I. Davidson, K. L. Wagstaff, and S. Basu. Measuring constraints-set utility for partitional clustering algorithms. In *Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases*, pages 115–126, Berlin, Germany, 2006.

[11] T. Denœux and M.-H. Masson. EVCLUS: evidential clustering of proximity data. *IEEE Trans. Systems, Man and Cybernetics: B*, 34:95–109, 2004.

[12] C. Dring, M.-J. Lesot, and R. Kruse. Data analysis with fuzzy clustering methods. *Computational Statistics & Data Analysis*, 51(1):192 – 214, 2006. The Fuzzy Approach to Statistical Analysis.

[13] D. Gondek and T. Hofmann. Non-redundant data clustering. *Knowledge and Information Systems*, 12(1):1–24, 2007.

[14] A. D. Gordon. A survey of constrained classification. *Computational Statistics & Data Analysis*, 21(1):17 – 29, 1996.

[15] N. Grira, M. Crucianu, and N. Boujemaa. Active semi-supervised fuzzy clustering. *Pattern Recognition*, 41(5):1851–1861, 2008.

[16] D.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, volume 17, pages 761–765, San Diego, CA, USA, 1978.

[17] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy cluster analysis: methods for classification, data analysis and image recognition.* John Wiley and Sons, New York, 1999.

[18] T. Kohonen. *Self-organizing Maps.* Springer, Berlin, 1997.

[19] R. Krishnapuram and J. M. Keller. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(2):98–110, 1993.

[20] M. H. C. Law, A. Topchy, and A. K. Jain. Clustering with soft and group constraints. *Lecture Notes in Computer Science*, 31:662–670, 2004.

[21] M.-H. Masson and T. Denœux. Clustering interval-valued data using belief functions. *Pattern Recognition Letters*, 25(2):163–171, 2004.

[22] M.-H. Masson and T. Denœux. ECM: An evidential version of the fuzzy c-means algorithm. *Pattern Recognition*, 41:1384–1397, 2008.

[23] M.-H. Masson and T. Denœux. RECM: Relational evidential c-means algorithm. *Pattern Recognition Letters*, 30:1015–1026, 2009.

[24] G. Shafer. *A mathematical theory of evidence.* Princeton University Press, Princeton, NJ, USA, 1976.

[25] P. Smets. The transferable belief model for quantified belief representation. In *Hanbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 1, pages 267–301, Norwell, MA, USA, 1998. Kluwer Academic Publishers.

[26] P. Smets and R. Kennes. The Transferable Belief Model. *Artificial Intelligence*, 66:191–234, 1994.

[27] K. Wagstaff. Value, cost, and sharing: Open issues in constrained clustering. *Lecture Notes in Computer Science*, 4747:1, 2007.

[28] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, Williamstown, MA, USA, 2001.

[29] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512, Cambridge, MA, 2002. MIT Press.

[30] R. R. Yager. On the normalization of fuzzy belief structures. *International Journal of Approximate Reasoning*, 14(2-3):127–153, 1996.

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

[32] S. Zhong and J. Ghosh. Scalable, balanced model-based clustering. In *Proc. 3rd SIAM Int. Conf. Data Mining*, pages 71–82, San Francisco, CA, USA, 2003.

Appendix A. Update equations for the ECM algorithm

For minimizing $J_{\mathrm{ECM}}(M, V)$, the necessary conditions of optimality for M gives the following adaptation rule for the mass functions:

$$m_{ij} = \frac{|A_j|^{-\alpha/(\beta-1)} d_{ij}^{-2/(\beta-1)}}{\sum_{A_k \neq \emptyset} |A_k|^{-\alpha/(\beta-1)} d_{ik}^{-2/(\beta-1)} + \rho^{-2/(\beta-1)}} \quad i = 1, n \quad \forall A_j \neq \emptyset \quad \text{(A.1)}$$

and

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij} \quad i = 1, n. \quad \text{(A.2)}$$

A more complex update rule is found for the centroids, since the optimality conditions lead to the resolution of a linear system at each step of the optimization process. Let $\mathbf{B}$ be a matrix of size $(c \times p)$ defined by:

$$\mathbf{B}_{lq} = \sum_{i=1}^{n} x_{iq} \sum_{A_j \neq \emptyset} |A_j|^{\alpha-1} m_{ij}^{\beta} s_{lj} = \sum_{i=1}^{n} x_{iq} \sum_{A_j \ni \omega_l} |A_j|^{\alpha-1} m_{ij}^{\beta} \quad l = 1, c \quad q = 1, p, \quad \text{(A.3)}$$

and $\mathbf{H}$ a matrix of size $(c \times c)$ given by:

$$\mathbf{H}_{lk} = \sum_{i} \sum_{A_j \neq \emptyset} |A_j|^{\alpha-2} m_{ij}^{\beta} s_{lj} s_{kj} = \sum_{i} \sum_{A_j \supseteq \{\omega_k, \omega_l\}} |A_j|^{\alpha-2} m_{ij}^{\beta} \quad k, l = 1, c. \quad \text{(A.4)}$$

With these notations, V is solution of the following linear system:

$$\mathbf{H}V = \mathbf{B}, \quad \text{(A.5)}$$

which can be solved using a standard linear system solver. The way of deriving equations (A.1) to (A.5) from the optimality conditions of the problem is detailed in reference [22]. Note that, in practice, the resolution of system (A.5) is performed columnwise: each column of V is the solution of a linear system of c equations and c unknowns.