



RECM: Relational Evidential c-means algorithm

Marie-Hélène Masson, Thierry Denoeux

► To cite this version:

Marie-Hélène Masson, Thierry Denoeux. RECM: Relational Evidential c-means algorithm. Pattern Recognition Letters, 2009, 30, pp.1015-1026. 10.1016/j.patrec.2009.04.008 . hal-00400273

HAL Id: hal-00400273

<https://hal.science/hal-00400273>

Submitted on 3 Jul 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RECM: Relational Evidential c -means algorithm

Marie-Hélène Masson¹* and Thierry Dencœux²

¹*Université de Picardie Jules Verne* - ²*Université de Technologie de Compiègne*
UMR CNRS 6599 Heudiasyc BP 20529 - F-60205 Compiègne cedex - France

Abstract

A new clustering algorithm for proximity data, called RECM (Relational evidential c -means) is presented. This algorithm generates a credal partition, a new clustering structure based on the theory of belief functions, which extends the existing concepts of hard, fuzzy and possibilistic partitions. Two algorithms, EVCLUS (Evidential Clustering) and ECM (Evidential c -Means) were previously available to derive credal partitions from data. EVCLUS was designed to handle proximity data, whereas ECM is a direct extension of fuzzy clustering algorithms for vectorial data. In this article, the relational version of ECM is introduced. It is compared to EVCLUS using various datasets. It is shown that RECM provides similar results to those given by EVCLUS. However, the optimization procedure of RECM, based on an alternate minimization scheme, is computationally much more efficient than the gradient-based procedure used in EVCLUS.

Key words: Clustering, proximity data, unsupervised learning, Dempster-Shafer theory, belief functions

1 Introduction

The term “cluster analysis” encompasses a number of different algorithms and methods for grouping objects of similar kind into categories. The similarity between objects is either computed using a suitable distance based on numeric attributes describing the objects (vectorial data), or directly available in the form of pairwise similarity or dissimilarity measurements (proximity data). A wide variety of methods for clustering object and proximity data has been

* Email: mmasson@hds.utc.fr. Fax: (33) 3 44 23 44 77. Tel: (33) 3 44 23 49 28

developed. They can be broadly classified into two main families: *hierarchical* and *hard or fuzzy partitioning* methods. Hierarchical methods provide a description of the data in the form of a sequence of nested clusters. Using hard partitioning methods, objects are grouped in an exclusive way, so that if a certain object belongs to a cluster then it cannot be included in another cluster. On the contrary, with fuzzy partitioning, each object may belong to several clusters with different degrees of membership.

For *vectorial data*, one of the most popular algorithms for deriving a fuzzy partition is the *fuzzy c-means* algorithm (FCM) [1]. In this method, the membership degrees of the objects are obtained through iterative minimization of an objective function, subject to the constraint that the sum of membership degrees over the clusters for each object be equal to 1. Several authors, having observed several shortcomings of FCM, among them the inability to detect atypical objects, have proposed variants of the original model. The *possibilistic c-means* algorithm (PCM) [17] and Davé's *noise clustering* (NC) [5] algorithms are two examples of such models. Fuzzy techniques for clustering *proximity data* include Roubens' fuzzy non metric model (FNM) model [23], the assignment-prototype (AP) model [31] and the relational fuzzy *c-means* (RFCM) model [14]. The latter approach was later extended by Hathaway and Bezdek [15] to cope with non-Euclidean dissimilarity data, leading to the non-Euclidean relational fuzzy *c-means* (NERFCM) model. Finally, robust versions of the FNM and RFCM algorithms were proposed by Davé [6].

Recently, a new concept of partition, *the credal partition*, developed in the framework of belief functions theory, has been introduced [8,9,20,21]. This concept generalizes existing concepts of hard, fuzzy (probabilistic) or possibilistic partitions by allowing an object to belong to several *subsets of classes*. Experiments have shown that this additional flexibility has the potential to bring a deeper insight into the data and to increase the robustness against outliers. Two algorithms, EVCLUS (Evidential Clustering) [8,9] and ECM (Evidential *c*-Means) [21] have been proposed in order to derive such credal partitions from data. EVCLUS was designed to handle proximity data, whereas ECM is a direct extension of Davé's algorithm and is only applicable to vectorial data. The determination of the partition in EVCLUS is founded on a gradient-based minimization of a criterion similar to the ones used in multidimensional scaling [3,4]. On the contrary, the search for the optimal parameters in ECM is carried out through an alternate optimization scheme of an objective function similar to FCM. This procedure is much more efficient than EVCLUS, but it is restricted to vectorial data. The aim of this article is thus to propose a version of ECM able to compute a credal partition from proximity data. As it is an evidential counterpart of the Relational Fuzzy *c*-Means algorithm [14], this new algorithm will be called "Relational Evidential *c*-Means" (RECM).

The rest of this paper is organized as follows. The concept of credal partition

and the ECM algorithm are first recalled in Section 2. The RECM algorithm is then introduced in Section 3, and an experimental comparison between RECM and EVCLUS using three datasets is presented in Section 4. Finally, Section 5 concludes the paper. Background notions on belief function theory and the EVCLUS algorithm are recalled in Appendices A and B, respectively.

2 Credal clustering

In this section, necessary notions on credal clustering are recalled. The concept of credal partition is first presented in Section 2.1, and the ECM algorithm for generating credal partitions from vectorial data is summarized in Section 2.2. The EVCLUS algorithm, which is used in Section 4 for comparison but is not central in this paper, is briefly described in Appendix B.

2.1 Credal partition

Let us consider a collection $O = \{o_1, \dots, o_n\}$ of n objects, and a set $\Omega = \{\omega_1, \dots, \omega_c\}$ of c classes forming a partition of O . Let us assume that we have only partial knowledge concerning the class membership of each object o_i , and that this knowledge is represented by a bba m_i on the set Ω . We recall that $m_i(\Omega)$ stands for complete ignorance of the class of object i , whereas $m_i(\{\omega_k\}) = 1$ corresponds to full certainty that object i belongs to class k . All other situations correspond to partial knowledge of the class of o_i . For instance, the following bba:

$$\begin{aligned} m_i(\{\omega_k, \omega_\ell\}) &= 0.7 \\ m_i(\Omega) &= 0.3 \end{aligned}$$

means that we have some belief that object i belongs either to class ω_k or to class ω_ℓ , and the weight of this belief is equal to 0.7.

Let $M = (m_1, \dots, m_n)$ denote the n -tuple of bbas related to the n objects. M is called a *credal partition* of O . Two particular cases are of interest:

- when each m_i is a *certain* bba, then M defines a conventional, crisp partition of Ω ; this corresponds to a situation of complete knowledge;
- when each m_i is a *Bayesian* bba, then M specifies a fuzzy partition of Ω , as defined by Bezdek [2].

As underlined in [21], a credal partition is a rich representation that carries a lot of information about the data. In [21], 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 $\text{BetP}_i(\{\omega_k\})$ induced by each bba m_i 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.

It was also proposed to characterize each cluster by two sets of objects. The *lower approximation* ω_k^L of a cluster ω_k is the set of objects that belong with no doubt to cluster ω_k : it is the set of objects assigned to the singleton $\{\omega_k\}$ in the hard credal partition; the *upper approximation* ω_k^U gathers the objects that could *possibly* belong to cluster ω_k : it is the set of objects assigned to subsets of Ω containing ω_k .

EXAMPLE 1 Let us consider a collection O of $n = 5$ objects and $c = 3$ classes. A credal partition M of O is given in Table 1. The class of object o_2 is known with certainty, whereas the class of o_5 is completely unknown. The three other cases correspond to situations of partial knowledge (m_4 is Bayesian). The corresponding pignistic probabilities are given in Table 2. Table 3 shows the hard credal partition. For instance, object 1 is assigned to the pair of clusters $\{\omega_1, \omega_2\}$, whereas object 4 is assigned to the singleton $\{\omega_3\}$. For lower and upper estimations of the clusters, we have, for instance, $\omega_3^L = \{4\}$, as object 4 is the unique object unambiguously assigned to cluster ω_3 , and $\omega_3^U = \{3, 4, 5\}$ as, in addition to object 4, objects 3 and 5 are assigned to sets of clusters containing ω_3 . The lower and upper approximations provide quite easy and intuitive summaries of the clustering results, as will be shown in Section 4.

INSERT TABLES 1, 2, 3

2.2 ECM algorithm

If the knowledge about the class membership of a set of objects is chosen to be represented by a credal partition, a method for extracting automatically this knowledge from data is needed. Two methods have already been proposed, the first one for proximity data (EVCLUS), the second one for vectorial data (ECM). Only the latter, which constituted the starting point for the work described in the present paper, will be recalled in this section.

Let us assume the available data to consist of a matrix $X = (x_{ik})$ of size $(n \times p)$, where p is the dimension of the feature space. In [21], an algorithm, referred to as ECM (Evidential c -Means) and inspired from Davé's noise clustering algorithm [5] (NC algorithm), was proposed to derive a credal partition from such *vectorial data*.

ECM determines, for each object o_i , a bba m_i in such a way that $m_i(A_j)$ is low (resp. high) when the distance d_{ij} between o_i and the focal set A_j is high (resp. low), where A_j is any non empty subset of Ω . The distance between an object and the subset A_j is defined as follows: as in fuzzy clustering, each class ω_k is represented by a center $\mathbf{v}_k \in \mathbb{R}^p$; then, each subset A_j of Ω is associated to the barycenter $\bar{\mathbf{v}}_j$ of the centers associated to the classes composing A_j . More precisely, with the notation

$$s_{kj} = \begin{cases} 1 & \text{if } \omega_k \in A_j \\ 0 & \text{otherwise,} \end{cases}, \quad (1)$$

the barycenter $\bar{\mathbf{v}}_j$ associated to A_j is computed as:

$$\bar{\mathbf{v}}_j = \frac{1}{c_j} \sum_{k=1}^c s_{kj} \mathbf{v}_k, \quad (2)$$

where $c_j = |A_j|$ denotes the cardinal of A_j and $c = |\Omega|$. The distance d_{ij} is then defined by:

$$d_{ij}^2 \triangleq \|\mathbf{x}_i - \bar{\mathbf{v}}_j\|^2. \quad (3)$$

To derive the credal partition $M = (m_1, \dots, m_n)$ and the matrix V of size $(c \times p)$ containing the cluster centers, ECM minimizes the following objective function:

$$J_{\text{ECM}}(M, V) \triangleq \sum_{i=1}^n \sum_{\{j/A_j \neq \emptyset, A_j \subseteq \Omega\}} c_j^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^n \delta^2 m_{i\emptyset}^\beta, \quad (4)$$

subject to

$$\sum_{\{j/A_j \subseteq \Omega, A_j \neq \emptyset\}} m_{ij} + m_{i\emptyset} = 1 \quad i = 1, \dots, n, \quad (5)$$

where $m_{i\emptyset}$ denotes $m_i(\emptyset)$. Criterion J_{ECM} is similar to that of the NC algorithm. The empty set is assimilated to a noise cluster considered to be at a fixed distance δ from each object. Parameter δ is used to control the number of objects considered as outliers. As in FCM, parameter β is used to tune the hardness of the partition. Note that additional weighting coefficients (c_j^α) are introduced for penalizing the subsets in Ω of high cardinality, the exponent α allowing us to control the degree of penalization.

To minimize J_{ECM} , an alternate optimization scheme, similar to FCM, was proposed in [21]. First, V is considered to be fixed. In [21], it is shown that

M has to be updated using the following equations:

$$m_{ij} = \frac{c_j^{-\alpha/(\beta-1)} d_{ij}^{-2/(\beta-1)}}{\sum_{A_k \neq \emptyset} c_k^{-\alpha/(\beta-1)} d_{ik}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}} \quad i = 1, \dots, n \quad \forall j/A_j \subseteq \Omega, A_j \neq \emptyset \quad (6)$$

and

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij} \quad i = 1, \dots, n. \quad (7)$$

In a second step, M is considered to be fixed. The minimization of J_{ECM} with respect to V is an unconstrained optimization problem. Setting to zero the partial derivatives of J_{ECM} with respect to the centers gives c linear equations in $\mathbf{v}_k$, which can be written as:

$$\sum_i \mathbf{x}_i \sum_{A_j \neq \emptyset} c_j^{\alpha-1} m_{ij}^\beta s_{lj} = \sum_k \mathbf{v}_k \sum_i \sum_{A_j \neq \emptyset} c_j^{\alpha-2} m_{ij}^\beta s_{lj} s_{kj} \quad l = 1, \dots, c. \quad (8)$$

Let H and U be two matrices of size $(c \times c)$ and $(c \times n)$, respectively, such that:

$$H_{lk} = \sum_i \sum_{A_j \neq \emptyset} c_j^{\alpha-2} m_{ij}^\beta s_{lj} s_{kj} \quad k, l = 1, \dots, c, \quad (9)$$

and

$$U_{li} = \sum_{A_j \neq \emptyset} c_j^{\alpha-1} m_{ij}^\beta s_{lj} \quad l = 1, c \quad i = 1, \dots, n. \quad (10)$$

With these notations, the c equations (8) can be written more compactly in matrix form as:

$$HV = UX. \quad (11)$$

Matrix equation (11) can be solved for V using a standard linear system solver. 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 ¹.

2.3 Discussion

Until now, two credal clustering algorithms are available: EVCLUS (see Appendix B) and ECM. Although founded on the same general model of partitioning, these two algorithms are very different.

¹ MATLAB codes for ECM and EVCLUS are available at <http://www.hds.utc.fr/~tdenoeux>.

EVCLUS is dedicated to proximity data. It does not use any explicit geometrical model of the data, so that it is applicable to both metric and non metric data. When the number of cluster is fixed, only one parameter (λ) has to be tuned. Parameter λ controls the overall complexity of the model, i.e., simultaneously, the mass allocated to the empty set and the mass allocated to non singleton subsets of Ω . The determination of the partition is achieved using gradient-based minimization of a stress function.

In contrast, ECM is in line with FCM and its variants: each cluster is represented by a prototype and the similarity between an object and a cluster is measured using an Euclidean metric. The number of parameters to be fixed in ECM is greater than in EVCLUS, allowing for a finer control of the allocation of the masses. Parameter β is fixed as in standard fuzzy clustering algorithm. Parameters α and δ allow a separate control of, respectively, the uncertainty of the partition, and the outlier rejection rate. The credal partition is obtained using an alternate optimization procedure. It turns out that this procedure is computationally much more efficient than the gradient-based algorithm used in EVCLUS. However it is only applicable to vectorial data. Finding a relational version of ECM, able to deal with proximity data, was thus of great interest. This problem is solved in the next section.

3 Relational formulation of ECM

In this section, the notion of a Euclidean dissimilarity matrix, as well as a criterion for checking this property, will first be recalled (Section 3.1). The derivation of the relational version of ECM will be presented in Section 3.2. Complexity issues and parameter tuning will then be addressed in Sections 3.3 and 3.4, respectively.

3.1 Euclidean embedding of the dissimilarities

We suppose in this section that the input data consists of a matrix $\Delta = (\delta_{ii'})$ of pairwise dissimilarities between the objects.

Δ is called *Euclidean* if there exists a description $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of the objects in a p -dimensional feature space such that $\delta_{ii'} = \|\mathbf{x}_i - \mathbf{x}_{i'}\|^2$. Let $W = (w_{ii'})$ be the matrix of size $(n \times n)$ of dot products of the $\mathbf{x}_i$. It is assumed, without loss of generality, that the centroid of the configuration of objects is placed at the origin. If Δ is Euclidean, the following relation holds [3,4,19]:

$$W = -\frac{1}{2}J\Delta J, \quad (12)$$

where $J = \frac{1}{n}ee^t - I$, with $e = (1, \dots, 1)^t \in \mathbb{R}^n$ and I is the $(n \times n)$ identity matrix.

The following theorem will be used in the experiments to check if the dissimilarities can be embedded in a Euclidean space:

THEOREM 1 (YOUNG AND HOUSEHOLDER [32])
 Δ is Euclidean if and only if W is positive semi-definite.

3.2 From ECM to RECM

The formulation of a relational version of ECM is founded on the explicit hypothesis that Δ is Euclidean. The update equations of ECM have to be expressed solely in terms of input dissimilarities or, equivalently, in terms of dot products between the $\mathbf{x}_i$, since relation (12) is supposed to hold.

We first note that the update equations (6) in ECM requires the computation of the Euclidean distance d_{ij} between each object $\mathbf{x}_i$ and the barycenter $\bar{\mathbf{v}}_j$ associated to each non empty subset A_j of Ω . This distance, defined by equation (3), can easily be expressed in terms of dot products in the feature space:

$$\begin{aligned} d_{ij}^2 &= (\mathbf{x}_i - \bar{\mathbf{v}}_j)^t (\mathbf{x}_i - \bar{\mathbf{v}}_j) \\ &= (\mathbf{x}_i - \frac{1}{c_j} \sum_{k=1}^c s_{kj} \mathbf{v}_k)^t (\mathbf{x}_i - \frac{1}{c_j} \sum_{l=1}^c s_{lj} \mathbf{v}_l) \\ &= \mathbf{x}_i^t \mathbf{x}_i - \frac{2}{c_j} \sum_{k=1}^c s_{kj} \mathbf{x}_i^t \mathbf{v}_k + \frac{1}{c_j^2} \sum_{k=1}^c \sum_{l=1}^c s_{kj} s_{lj} \mathbf{v}_k^t \mathbf{v}_l. \end{aligned} \quad (13)$$

Only the dot products $\mathbf{x}_i^t \mathbf{x}_i$ are directly available, thanks to (12). A way to compute the other products $\mathbf{x}_i^t \mathbf{v}_k$ and $\mathbf{v}_k^t \mathbf{v}_l$ must be found. It turns out that they can be easily derived from (11). Let us introduce the following notations: let X denote the matrix of size $(n \times p)$ of objects coordinates, let $Q = (q_{kk'})$ be the matrix of size $(c \times c)$ of dot products of the $\mathbf{v}_k$ ($q_{kk'} = \mathbf{v}_k^t \mathbf{v}_{k'}$) and $R = (r_{ki})$ the matrix of size $(c \times n)$ of dot products between the $\mathbf{v}_k$ and the $\mathbf{x}_i$ ($r_{ki} = \mathbf{v}_k^t \mathbf{x}_i$). The following relations hold: $W = XX^t$, $R = VX^t$ and $Q = VV^t$. Let P be any matrix introduced before. Notations $P_{\cdot i}$ and $P_{i \cdot}$ are used, respectively, to denote the i th column and the i th row of P .

Starting from (11), matrices Q and R can be determined in two successive steps:

- (1) **Determination of R.** By right multiplying both sides of (11) by $\mathbf{x}_i$, we

get:

$$H \begin{pmatrix} \mathbf{v}_1^t \mathbf{x}_i \\ \vdots \\ \mathbf{v}_c^t \mathbf{x}_i \end{pmatrix} = U \begin{pmatrix} \mathbf{x}_1^t \mathbf{x}_i \\ \vdots \\ \mathbf{x}_n^t \mathbf{x}_i \end{pmatrix}, \quad (14)$$

or, equivalently,

$$H R_{.i} = U W_{.i}, \quad (15)$$

so that each column $R_{.i}$ ($i = 1, n$) of R is obtained by solving a system of c linear equations (14) with c unknowns $\mathbf{v}_k^t \mathbf{x}_i$, $k = 1, c$. The whole matrix R is thus obtained by solving n such linear systems.

- (2) **Determination of Q .** By right multiplying both sides of (11) by $\mathbf{v}_i$, we see that each column $Q_{.i}$ ($i = 1, n$) of Q is in turn solution of a linear system with c equations and c unknowns $\mathbf{v}_k^t \mathbf{v}_i$, $k = 1, c$:

$$H \begin{pmatrix} \mathbf{v}_1^t \mathbf{v}_i \\ \vdots \\ \mathbf{v}_c^t \mathbf{v}_i \end{pmatrix} = U \begin{pmatrix} \mathbf{x}_1^t \mathbf{v}_i \\ \vdots \\ \mathbf{x}_n^t \mathbf{v}_i \end{pmatrix}, \quad (16)$$

or, equivalently,

$$H Q_{.i} = U R_{.i}. \quad (17)$$

The whole matrix Q is thus obtained after solving c linear systems.

The above equations allow the formulation of RECM, a variant of ECM dealing with proximity data. It can be summarized in the following steps:

- Step 1. Fix $c, \alpha, \beta, \delta^2$ and ϵ ;
- Step 2. Generate randomly the initial credal partition $M^{(0)}$;
- Step 3. (a) Set $k \leftarrow 0$;
- (b) Compute W from Δ using (12);
- Step 4. **Repeat**
- (a) $k \leftarrow k + 1$;
- (b) Compute $H^{(k)}$ and $U^{(k)}$ from $M^{(k-1)}$ using (9) and (10);
- (c) For $i = 1, n$
- Compute the i th column of $R^{(k)}$ using (14);
- EndFor
- (d) For $i = 1, c$
- Compute the i th column of $Q^{(k)}$ using (16);
- EndFor
- (e) Compute the d_{ij} from $R^{(k)}$ and $Q^{(k)}$ using (13);
- (f) Update $M^{(k)}$ using (6) and (7);
- Until** $\|M^{(k)} - M^{(k-1)}\| < \epsilon$

REMARK 1 It is interesting to note that a kernelized version of ECM may be easily derived from the RECM algorithm by replacing the dot products $\mathbf{x}_i^t \mathbf{x}_{i'}$ in matrix W by kernel functions $K(\mathbf{x}_i, \mathbf{x}_{i'})$. This approach has been exploited by several authors for formulating clustering algorithms in a kernel-induced feature space (see, for example, [12]).

3.3 Complexity analysis

In [9], the algorithmic complexity of EVCLUS was analyzed. It was shown that one iteration of the optimization procedure necessitates $O(f^3 n^2)$ operations where f denotes the number of focal elements ($f = 2^c$ for the complete model). A reduction of the complexity was thus suggested by restricting the focal elements to the singletons, the empty set and Ω , so that $f = c + 2$. In this way, calculations involving a few hundreds objets and a limited number of classes were made tractable.

We analyze below the different substeps inside in the main loop (step 4) of the optimization procedure in RECM:

- substep (b): it requires $nf c^2$ and $nf c$ operations for computing H and U ;
- substep (c): for each of the n systems to be solved, we need nc operations to compute UW_i and the resolution of one system is $\mathcal{O}(c^2)$. Considering that $n \gg c$, the overall complexity of this step is $\mathcal{O}(n^2 c)$;
- substep (d): for each of the c systems to be solved, we need nc operations to compute UR_i and the resolution of one system is $\mathcal{O}(c^2)$. The overall complexity of this step is $\mathcal{O}(nc^2)$; the complexity of steps (c) and (d) can thus be evaluated to $\mathcal{O}(n^2 c)$.
- substep (e): the complexity of this step is $\mathcal{O}(nf c^2)$;
- substep (f): the complexity of this step is $\mathcal{O}(nf)$;

The overall complexity of one iteration in RECM is thus of $\mathcal{O}(nf c^2 + n^2 c)$. Considering that $c < f < n$, it is clear that this complexity is lower than the complexity of EVCLUS. In particular, the complexity of RECM is only linear with respect to the number f of focal elements, which is a major improvement over the cubic complexity of EVCLUS. Moreover, the number of iterations needed until convergence is by far lower with RECM, as will be shown in the experiments of Section 4. These points make RECM more attractive than EVCLUS when the number of objects to be classified is high.

3.4 Guidelines for parameter setting

The RECM algorithm depends on three parameters (α , β and δ^2) that have to be tuned to achieve good results. Parameter β has the same meaning as the fuzzification constant h of fuzzy clustering algorithms such as RFCM. A usual choice for h is 2. However, with this value for β , it was observed that RECM may either not converge, or converge toward a degenerate solution for some datasets. This phenomenon, which is also observed with fuzzy algorithms, occurs when the data is not Euclidean, or has a particular structure. In those cases, β should be lowered. Checking the positivity of the eigenvalues of W can be a valuable way to define a starting value for β . We recommend to start anyway with $\beta = 1.5$ and to lower β is needed.

Parameter α controls the fraction of mass allocated to non singleton subsets of Ω . A value $\alpha = 1$ can be considered as a good starting point and can be modified according to what is expected from the user: the higher α , the more the focal sets of cardinality greater than one are penalized, i.e., the more the partition moves toward a fuzzy partition. On the contrary, if α is set to zero, all focal elements are penalized in the same way.

Parameter δ^2 , as in the NC clustering method of Davé, represents the fixed distance which is assumed between each object and the noise cluster (represented by the empty set). It controls the number of objects considered as

outliers. If its value is greater than the maximum found in Δ , no rejection will be considered. It can then be lowered to achieve a given rejection rate.

It should be emphasized that RECM, as any clustering algorithm, is an exploratory data analysis tool and that it always requires some kind of subjective analysis of the results. Although general guidelines for parameter setting can be given (as was done in this section), we do not think that the parameter tuning process can be performed in a fully automatic way for all datasets. In practice, we recommend selecting parameter values by trial and error, starting from default values, with the help of graphical displays such as MDS maps.

4 Experiments

In this section, we compare the behavior of EVCLUS and RECM using the three datasets presented in [9]. We also report the results obtained with five classical fuzzy algorithms for proximity data: the assignment-prototype algorithm (AP) [31], the Fuzzy Non Metric algorithm (FNM) [23], the Relational Fuzzy c -means algorithm (RFCM) [14], and its “Noise” version (NRFCM) [6], and the non-Euclidean RFCM algorithm (NERF) [15]. Among these five algorithms, three of them have a fuzzification constant h , similar to β , that controls the degree of “hardness” of the resulting fuzzy partition. NRFCM has another parameter, the distance to the noise cluster, which plays a role similar to δ .

The first dataset is a synthetic one, the other two contain real data. Comparisons focus on the ability to discover meaningful partitions, on the ability to provide interpretable representations of the data, and on the computational efficiency of the algorithms. For the comparison of the computational complexity of RECM and EVCLUS, the mean running times over 50 runs of the algorithms are reported together with the coefficient of variation of the final stress function (ratio of the standard deviation to the mean) and the number of iterations needed to achieve a given level of convergence. Note that, for a fair comparison, the same stopping criterion was used in the two algorithms: the procedure was stopped when the norm of the difference between $M^{(k)}$ and $M^{(k-1)}$ was below a given threshold (equal to 10^{-5}). The limited version of EVCLUS (keeping only singleton elements, Ω and the empty set) will be referred to in the experiments as EVCLUS-1, as opposed to EVCLUS-2 that designates the full version with all the 2^c focal elements.

4.1 Diamond dataset

The first example is inspired from a classical dataset proposed by Windham [31]. It is composed of 13 objects. Twelve objects (objects 2 to 13) are represented in a two-dimensional space, as shown in Figure 1, and their dissimilarities are computed as squared Euclidean distances. Object 13 is an outlier and has been added to Windham’s dataset to test the ability of the algorithm to detect outlying observations. Additionally, the dissimilarities to a 13th object (object 1) have been added in the data matrix. This object has no representation in the attribute space; it may be called an “inlier”, as is quite similar to all other objects (the dissimilarity matrix is given in [9]).

In [9], we compared the results obtained with EVCLUS and the fuzzy algorithms. These results are reproduced in Figure 2, which shows the membership degrees obtained by the fuzzy algorithms as well as the bbas generated by EVCLUS. As discussed in [9], every algorithm finds a reasonable partition of the data, but EVCLUS is the only algorithm able to detect atypical objects like object 1 and object 13.

We now compare these results with the ones obtained by RECM. The data is non Euclidean because of object 1, as confirmed by some negative eigenvalues of W (see Figure 3). Consequently, we used $\beta = 1.5$; a moderate penalization $\alpha = 1$ was chosen and δ^2 was fixed to 25. We can see in Figure 4 that RECM gives correct results: the two natural clusters are recovered, the mass allocated to the empty set allows the detection of the outlier, and object 7, which is between the two clusters is characterized by a high mass on Ω . The difference between EVCLUS and RECM lies in the masses allocated for the inlier (object 1). EVCLUS allocates the totality of the mass to Ω , whereas RECM distributes the mass equally between the singletons, what can be considered as an equally valuable solution.

Concerning the comparison of the computational complexity, the results are given in Table 4. We can see that RECM outperforms EVCLUS in the two cases considered (without or with inlier), even if the differences tend to be smaller in the second case where the data is non Euclidean: the cpu time is shorter, the number of iterations needed for convergence is lower and the results are less variable.

INSERT FIGURES 1 TO 4

INSERT TABLE 4

4.2 “Cat cortex” dataset

This second dataset consists of a matrix of connection strengths between 65 cortical areas of the cat brain. It was used by several authors for illustrating algorithms for visualizing, classifying or clustering proximity data [13,16] and was already used in [9] to evaluate EVCLUS. The proximity values are measured on an ordinal scale and range from 0 (self-connection), to 4 (absent or unreported connection) with intermediate values: 1 (dense connection), 2 (intermediate connection) and 3 (weak connection). The representation of the eigenvalues of matrix W in Figure 3 shows that the dissimilarity matrix is not Euclidean. From functional considerations, the cortex can be divided into four regions: auditory (A), visual (V), somatosensory (S), and frontolimbic (F). The clustering task is to find a four-class partition of the 65 cortical areas, based on the dissimilarity data, which is consistent with the functional regions. As reported in [9], only three points out of 65 were misclassified using EVCLUS-1. This error rate was consistent with the leave-one-out rates obtained by Graepel *et al.* [13] in a supervised setting. Correct solutions were provided by FNM, RFCM, and NRFCM for small values of the fuzzification parameter h ($h < 1.4$). The best solution, 3 errors among 65, was obtained by RFCM with $h = 1.2$.

To apply RECM on this dataset, the parameter δ^2 was fixed to a value greater than the maximum of dissimilarities ($\delta^2 = 5$) because the detection of outliers was not the aim of the study. As the fuzzy algorithms, RECM converges toward a useless solution with an equal mass on all the focal elements when β is set to 2. Setting β to 1.1 gives interesting solutions presented in Figures 5, 6 and 7 for different values of α . A 2-D map of the data was obtained from a multidimensional scaling algorithm. Hard partitions were computed by assigning each object to the class with highest pignistic probability (A.3). They are presented using different symbols with size proportional to the maximum pignistic probability. By this way, a hard and a fuzzy partition are represented on the same graph. For $\alpha = 0.5$, three wrong classifications were observed, a result equivalent to that of EVCLUS². For smaller values of α , slightly higher error rates were observed (from 4 to 7 errors).

We can gain more insight into the data by studying the lower and upper approximations of each cluster computed from the credal partition. For $\alpha = 0.5$, the lower and upper approximations are equal since the mass is in totality allocated to the singletons of Ω (see Figure 5). When using smaller values of

² It should be noted that classification errors are just given here as a means to check that RECM and EVCLUS yield comparable results in terms of hard partitions, and that these results are consistent with a physical description of the problem. However, these algorithms were *not* designed to minimize a classification error, as the goal of cluster analysis is *not* to minimize the discrepancy with a known partition.

α , we can see that the lower approximations concentrate on the cores of the clusters, whereas the upper approximations become more and more imprecise until including almost all the data points (see Figures 6 and 7).

In [9], only the results using EVCLUS-1 were reported. When trying to apply EVCLUS-2 to this data to obtain a more general credal partition, it turns out that correct classifications rates could be obtained only when the penalization applied to non singletons element was very strong. In this case, the mass is allocated in priority to the singletons of Ω : consequently, EVCLUS-1 and EVCLUS-2 give similar results. Although corresponding to good classification rates, the structures found are not general credal partitions taking full advantage of the belief functions framework. When λ is lowered, EVCLUS-2 is much more unstable, many local minima are found and the algorithm fails to provide interesting solutions. In contrast, RECM is fast and it is easy to set parameter α once β and δ^2 have been fixed to obtain a desired level of description of the data, as illustrated by figures 5 to 7. These remarks are supported by the experimental results reported in Table 5. It may be seen that, despite the non Euclidean nature of the data, RECM is very efficient and produces very stable results. In contrast, EVCLUS, in the full version, is slow and produces variable results. Note that, although the results are strongly dependent on the values of λ and ϵ , the main tendencies observed are maintained whatever their values.

INSERT FIGURES 5 TO 7

INSERT TABLE 5

4.3 “Protein” dataset

This real dataset consists of a dissimilarity matrix measuring the structural proximity of 213 proteins sequences [16,13]. The eigenvalues of corresponding inner products matrix W being all positive, the dissimilarity matrix is Euclidean. The proteins are divided into 4 classes of globins: hemoglobins- α (HA), hemoglobins- β (HB), myoglobins (M) and heterogeneous globins (G). The study consists in checking whether these four natural clusters could be recovered from the dissimilarities.

As reported in [9], the best classification result with the fuzzy algorithms was obtained by RFCM with a fuzzification constant $h = 1.05$: five proteins were misclassified. EVCLUS-1 was run with $\lambda = 0.005$. The hard partition computed from the maximum pignistic probabilities leads to only one misclassification out of 213. As with the cat cortex dataset, we found out in these new experiments that EVCLUS-2 is trapped in local minima and fails to provide good solutions in terms of correct classification rate.

To apply RECM, the maximum of the dissimilarities being 13.64, we fixed $\delta^2 = 20$. As in the fuzzy algorithms, we chose a low value for β ($=1.1$). We set $\alpha = 0$ so as to obtain masses on non-singleton subsets of Ω . A comparison of the hard and fuzzy partitions obtained using EVCLUS-1 and RECM is presented in Figure 8. It can be seen that they are very similar. Three points are misclassified with RECM. The lower and upper approximations of the clusters are shown in Figure 9; they can be considered as good summaries of the data. Additionally, a comparison between the mass allocated to the empty set by both algorithms is also provided in Figure 10 (the size of the symbols is proportional to the mass of the empty set). The G -class, although situated in the middle of the MDS configuration, was found in [9] to receive the highest mass on the empty set. A reasonable explanation was that the members of this class are characterized by a high within-class dissimilarity value and, at the same time, very small differences between within and between-class dissimilarities. It may be seen from Figure 10 that the peculiarity of the G -class is also detected by RECM. The experimental results concerning the execution times and the variability of the solutions are given in Table 6. The same conclusions as before can be drawn: RECM is faster and more stable.

INSERT FIGURES 8 TO 10

INSERT TABLE 6

4.4 Discussion

For which application is RECM designed? In problems where vectorial data are available, it is in general more efficient to apply clustering directly to the object-attribute matrix X rather than applying a relational clustering algorithm to a dissimilarity matrix derived from X . The first obvious application of relational clustering is when relational data is the only available data. This situation often occurs in domains like psychology, bioinformatics, economics, psychophysics... Another interest of relational algorithms is when the dissimilarity between objects cannot be properly measured using the standard Euclidean norm. In web mining applications, e.g., some authors have proposed dissimilarity measures mixing numerical and non numerical attributes so as to take into account the structure of web sites [18,24]. RECM belongs to the same family of algorithms as RFCM, AP or NERFCM. As underlined by Bezdeck [1, page 181], due to their computational complexity, these algorithms are not well-adapted to handle very large data sets, which is not the issue addressed in this paper. As RFCM and similar algorithms, RECM is best suited to handle a few hundreds of objects.

Pros and cons with respect to fuzzy relational algorithms. The complexity of RECM and EVCLUS is higher than that of conventional fuzzy relational clusterings. If, as a final result, only a hard partition is needed, evidential algorithms cannot be recommended. However, we have shown that, as a counterpart of a higher complexity, these algorithms are able to provide, in the case of small or moderate size datasets, a very rich description of the structure of the data. Moreover, the possibility of combining several partitions in the evidential framework has been shown to be an interesting feature of these algorithms [9,21].

RECM or EVCLUS? The three experiments above have shown the strengths and weaknesses of both algorithms. As EVCLUS makes no assumption about the nature of the dissimilarities, it is *a priori* more suited to non Euclidean dissimilarities. It also seems to be simpler to use, as only one parameter (λ) has to be set. The counterpart of this simplicity is a lack of flexibility, as the outlier detection rate and the imprecision of the partition (mass allocated to non singletons) are controlled by a single parameter. Experiments have shown that the gradient-based optimization procedure is not efficient when the full version with 2^c focal elements is used. The limited version of EVCLUS seems to be more advisable, although not providing a general credal partition. In contrast, the optimization algorithm in RECM has been shown to be computationally much more efficient. The method makes the explicit assumption that the input dissimilarities are computed as squared Euclidean distances in a vector space. However, reasonable solutions, very close to those of EVCLUS, were obtained in our experiments even when the Euclidean assumption was not verified. Moreover, RECM allows us to better exploit the expressive power of the belief function framework by a proper adjustment of the parameters.

5 Conclusion

The concept of credal partition is a recently introduced generalization of hard, fuzzy and possibilistic partitions, which makes use of the expressive power of the Dempster-Shafer theory of belief functions. A credal partition is more general than a fuzzy partition in that masses (similar to membership degrees) are assigned to sets of clusters with any cardinality. As shown in previous work [9,20,21], this formalism can be used to generate meaningful representations of the data. In particular, the lower (respectively, upper) approximation of a cluster can be computed as the set of objects certainly (respectively, possibly) belonging to that cluster.

Until now, only two algorithms were available for automatically constructing a credal partition from learning data. The EVCLUS algorithm [9] is based on

gradient descent of a stress function measuring the discrepancy between object dissimilarities and degrees of conflict of the associated bbas. This algorithm can be applied to proximity data, but it is rather slow and, practically, it can only be used to generate special kinds of credal partitions in which the masses are assigned to focal elements of cardinality 0, 1 and c , where c is the number of clusters. The other algorithm, called ECM [21], is an evidential counterpart of the fuzzy c -means and is based on alternative optimization of cluster centers and belief masses. This algorithm is computationally much more efficient than EVCLUS, but it can only be applied to vectorial data.

In this paper, a relational version of ECM, called RECM, has been introduced. This new algorithm can be seen as an evidential counterpart of relational fuzzy clustering algorithm such as RFCM. Although based on the assumption that the input dissimilarities are squared Euclidean distances, this algorithm has been shown to yield results comparable to those provided by EVCLUS even in case of non Euclidean data. The advantages of RECM over EVCLUS are twofold: first, RECM is faster and more stable; secondly, it allows the construction of general credal partition in which belief masses are assigned to focal sets of any cardinality, thus exploiting the full expressive power of belief functions.

Although the application of belief function theory to supervised or partially supervised classification has been well developed (see, e.g., [7,10,22]), the application of this theory to unsupervised classification has been until now very limited, partly due to the lack of efficient algorithms. Potential applications of this approach include the fusion of clustering results (see, e.g., [11]) and the integration of prior knowledge in clustering [30]. We believe that the availability of efficient algorithms such as the one introduced in this paper will allow further progress in this direction.

References

- [1] J. C. Bezdek. *Pattern Recognition with fuzzy objective function algorithms..* Plenum Press, New-York, 1981.
- [2] J. C. Bezdek, J. Keller, R. Krishnapuram, and N. R. Pal. *Fuzzy models and algorithms for pattern recognition and image processing.* Kluwer Academic Publishers, Boston, 1999.
- [3] I. Borg and P. Groenen. *Modern multidimensional scaling.* Springer, New-York, 1997.
- [4] T. F. Cox and M. A.A. Cox. *Multidimensional scaling.* Chapman and Hall, London, 1994.

- [5] R.N. Davé. Characterization and detection of noise in clustering. *Pattern Recognition Letters*, 12, 657-664, 1991.
- [6] R.N. Davé. Clustering of relational data containing noise and outliers. In *FUZZ'IEEE 98*, vol. 2, pages 1411–1416, 1998.
- [7] T. Dencœux. A k -nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Trans. on Systems, Man and Cybernetics*, 25(05):804–813, 1995.
- [8] T. Dencœux and M.-H. Masson. Clustering of proximity data using belief functions. B. Bouchon-Meunier, L. Foulloy and R. R. Yager, Eds. *Intelligent systems for information processing from representation to application*. 291-302, Elsevier, Amsterdam, 2003.
- [9] T. Dencœux and M.-H. Masson. EVCLUS: EVidential CLUStering of proximity data. *IEEE Transactions on Systems, Man and Cybernetics Part B*, 34(1), 95-109, 2004.
- [10] T. Dencœux and P. Smets. Classification using belief functions: the relationship between the case-based and model-based approaches. *IEEE Transactions on Systems, Man and Cybernetics B*, 36(6):1395–1406, 2006.
- [11] A.L.N. Fred and A.K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6), 835-850, 2005.
- [12] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Trans. on Neural Networks*, 13(3), 780-784, 2002.
- [13] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. in *Advances in Neural Information Processing Systems 11*, M. Kearns, S. Solla, and D. Kohn, eds., MIT Press, Cambridge, MA, 438–444, 1999.
- [14] R.J. Hathaway, J.W. Davenport, and J.C. Bezdek. Relational duals of the c-means clustering algorithms. *Pattern Recognition*, 22, 205-212, 1989.
- [15] R.J. Hathaway and J.C. Bezdek. Nerf c-means : Non-euclidean relational fuzzy clustering. *Pattern Recognition*, 27:429–437, 1994.
- [16] T. Hofmann and J. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, 1997.
- [17] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(2), 98-110, 1993.
- [18] R. Krishnapuram, A. Joshi, O. Nasraoui and L. Yi. Low-complexity fuzzy relational clustering algorithms for web mining. *IEEE Transactions on Fuzzy Systems*, 9, 595-608, 2001.
- [19] J. Laub and K.-R. Muller. Feature discovery in non-metric pairwise data. *Journal of Machine Learning Research*, 5, 801-818, 2004.

- [20] M.-H. Masson and T. Denœux. Clustering interval-valued data using belief functions. *Pattern Recognition Letters*, 25(2), 163-171, 2004.
- [21] M.-H. Masson and T. Denœux. ECM: An evidential version of the fuzzy *c*-means algorithm. *Pattern Recognition*, 41, 1384-1397, 2008.
- [22] B. Quost, T. Denoeux and M.-H. Masson. Pairwise classifier combination using belief functions. *Pattern Recognition Letters*, 28(5) , 644-653, 2007.
- [23] M. Roubens. Pattern classification problems and fuzzy sets. *Fuzzy sets and systems*, 1:239–253, 1978.
- [24] T.A. Runkler and J.C. Bezdek. Web mining with relational clustering. *International Journal of Approximate Reasoning*, 32(2-3), 217-236, 2003.
- [25] J. W. Sammon. A non-linear mapping for data structure analysis. *IEEE Transactions on Computers*, 18:401–409, 1969.
- [26] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, Princeton, N.J., 1976.
- [27] P. Smets. The combination of evidence in the Transferable Belief Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5), 447-458, 1990.
- [28] P. Smets and R. Kennes. The Transferable Belief Model. *Artificial Intelligence*, 66, 191-243, 1994.
- [29] P. Smets. Decision making in the TBM: the necessity of the pignistic transformation. *International Journal of Approximate Reasoning*, 38(5), 133-147, 2005.
- [30] L. Tari, C. Baral and S. Kim Fuzzy c-means clustering with prior biological knowledge. *Journal of Biomedical Informatics*, in press, 2008.
- [31] M.P. Windham. Numerical classification of proximity data with assignment measures. *Journal of classification*, 2:157–172, 1985.
- [32] G. Young and A.S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3:19–22, 1938.

Table 1

Credal partition of Appendix B example.

F	$m_1(F)$	$m_2(F)$	$m_3(F)$	$m_4(F)$	$m_5(F)$
$\emptyset$	0	0	0	0	0
$\{\omega_1\}$	0	0	0	0.2	0
$\{\omega_2\}$	0	1	0	0.35	0
$\{\omega_1, \omega_2\}$	0.7	0	0	0	0
$\{\omega_3\}$	0	0	0.2	0.45	0
$\{\omega_1, \omega_3\}$	0	0	0.5	0	0
$\{\omega_2, \omega_3\}$	0	0	0	0	0
Ω	0.3	0	0.3	0	1

Table 2

Pignistic probabilities for the credal partition of Appendix B example.

i	$\text{BetP}_1(\{\omega_i\})$	$\text{BetP}_2(\{\omega_i\})$	$\text{BetP}_3(\{\omega_i\})$	$\text{BetP}_4(\{\omega_i\})$	$\text{BetP}_5(\{\omega_i\})$
1	0.45	0	0.35	0.2	1/3
2	0.45	1	0.1	0.35	1/3
3	0.1	0	0.55	0.45	1/3

Table 3

Hard credal partition of Appendix B example.

F	$m_1(F)$	$m_2(F)$	$m_3(F)$	$m_4(F)$	$m_5(F)$
$\emptyset$	0	0	0	0	0
$\{\omega_1\}$	0	0	0	0	0
$\{\omega_2\}$	0	1	0	0	0
$\{\omega_1, \omega_2\}$	1	0	0	0	0
$\{\omega_3\}$	0	0	0	1	0
$\{\omega_1, \omega_3\}$	0	0	1	0	0
$\{\omega_2, \omega_3\}$	0	0	0	0	0
Ω	0	0	0	0	1

Table 4

Experimental comparison between RECM (with $\alpha = 1$, $\beta = 1.5$, $\delta^2 = 25$) and EVCLUS (with $\lambda = 0.1$) on the Diamond dataset. The coefficient of variation is the ratio of the standard deviation to the mean of the final value of the stress function, over 50 runs of both algorithms.

	RECM	EVCLUS
Without inlier (object 1)		
CPU time	0.44 ± 0.05	3.70 ± 0.28
Nb iterations	14 ± 3	1626 ± 122
coefficient of variation	0%	12%
With inlier (object 1)		
CPU time	1.61 ± 0.20	3.82 ± 0.45
Nb iterations	66 ± 8	1599 ± 182
coefficient of variation	0%	8.03%

Table 5

Experimental comparison between RECM (with $\alpha = 0.5$, $\beta = 1.1$, $\delta^2 = 5$) and EVCLUS (with $\lambda = 10^{-3}$) on the cat cortex dataset. The coefficient of variation is the ratio of the standard deviation to the mean of the final value of the stress function, over 50 runs of both algorithms.

	RECM	EVCLUS-1	EVCLUS-2
CPU time	2.23 ± 0.74	18.51 ± 3.87	27.03 ± 4.06
Nb iterations	19 ± 7	1652 ± 333	1830 ± 274
coefficient of variation	0.6%	1.34%	5.07%

Table 6

Experimental comparison between RECM (with $\alpha = 0$, $\beta = 1.1$, $\delta^2 = 20$) and EVCLUS (with $\lambda = 0.005$) on the protein dataset. The coefficient of variation is the ratio of the standard deviation to the mean of the final value of the stress function, over 50 runs of both algorithms.

	RECM	EVCLUS-1	EVCLUS-2
CPU time	4.75 ± 0.86	91.21 ± 11.56	88.07 ± 22.73
Nb iterations	16 ± 3	672 ± 60	948 ± 237
coefficient of variation	$10^{-5}\%$	4.06%	6.5%

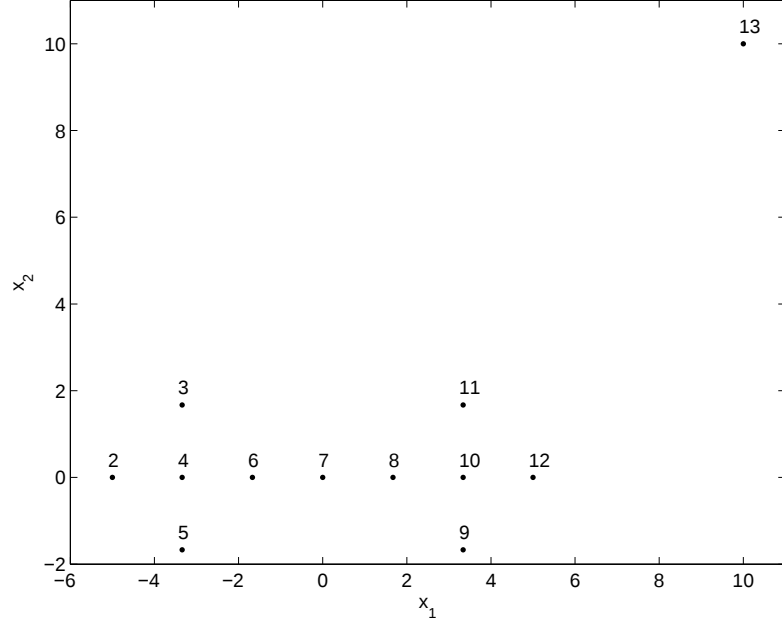


Fig. 1. Diamond dataset (without inlier).

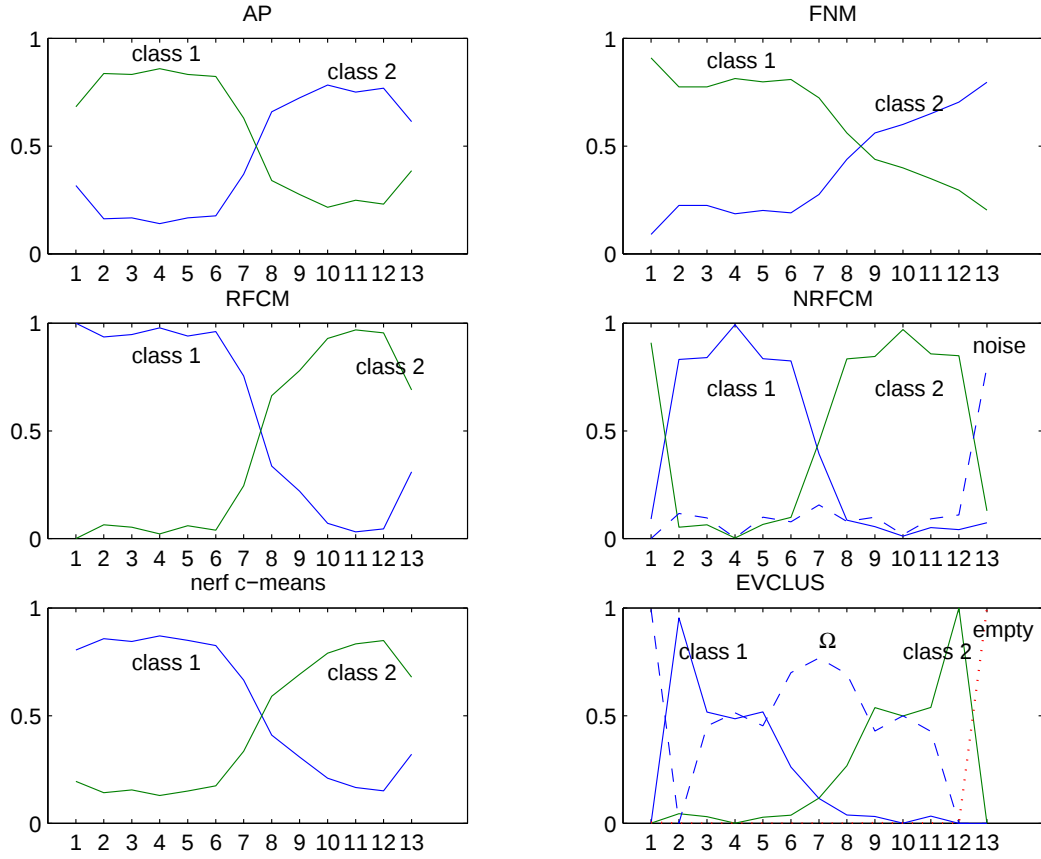


Fig. 2. Diamond dataset: results with 5 fuzzy algorithms and EVCLUS (reproduced from [9]).

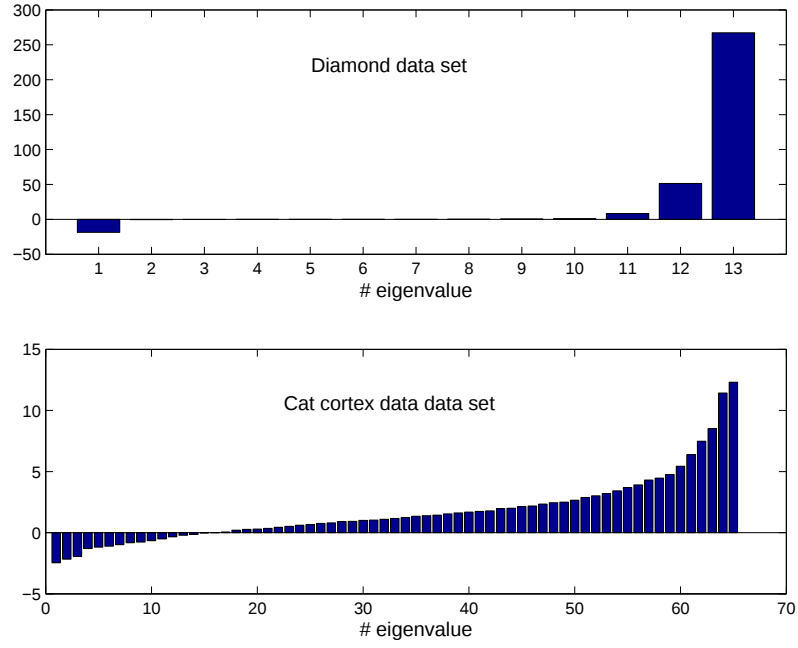


Fig. 3. Diamond and cat cortex data sets; Eigenvalues of W .

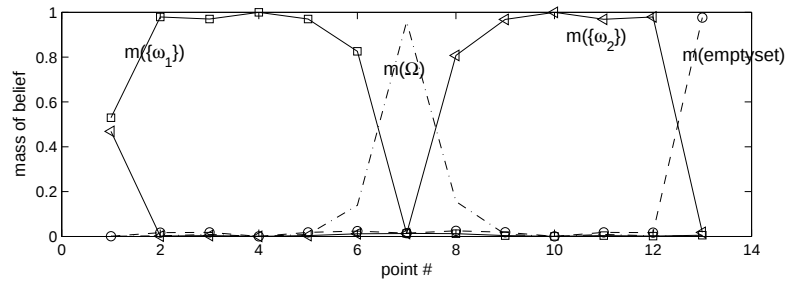


Fig. 4. Diamond dataset: results with RECM and $\alpha = 1$, $\beta = 1.5$ and $\delta^2 = 25$.

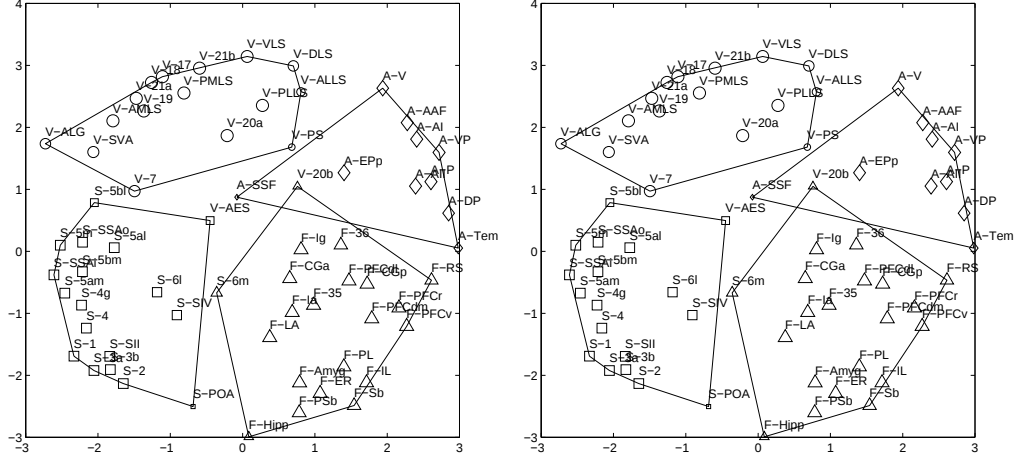


Fig. 5. Cat cortex dataset: lower (left) and upper (right) approximations of the clusters together with maximum pignistic assignments. A different symbol is used for each cluster found by RECM, the symbol size being proportional to the pignistic probability of the corresponding group. The first letter of each label (S,V,A,F) indicates the true class memberships. Settings : $\alpha = 0.5$, $\beta = 1.1$, $\delta^2 = 5$.

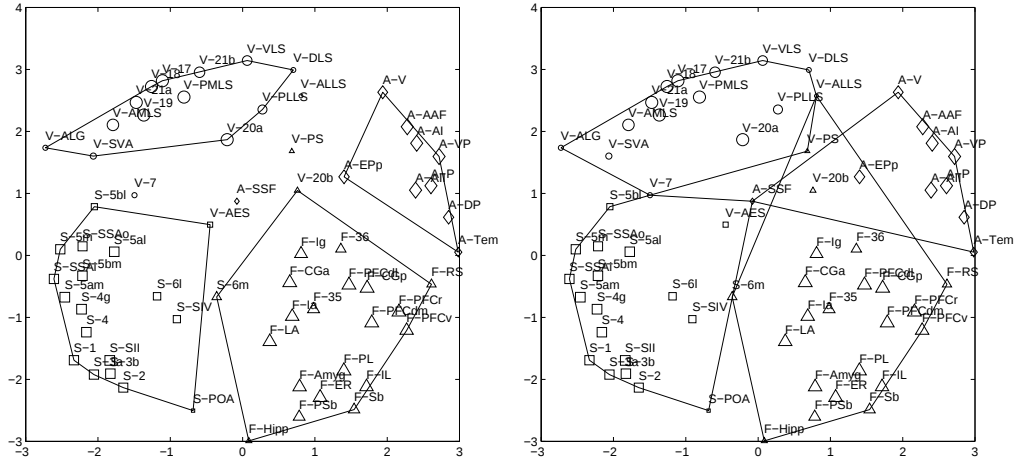


Fig. 6. Cat cortex dataset: lower (left) and upper (right) approximations of the clusters together with maximum pignistic assignments ($\alpha = 0.2$, $\beta = 1.1$, $\delta^2 = 5$).

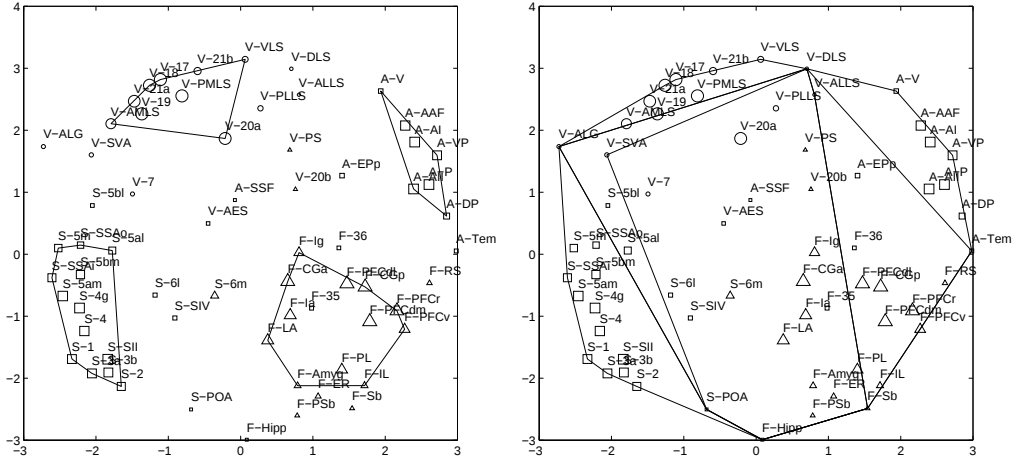


Fig. 7. Cat cortex: lower (left) and upper (right) approximations of the clusters together with maximum pignistic assignments ($\alpha = 0$, $\beta = 1.1$, $\delta^2 = 5$).

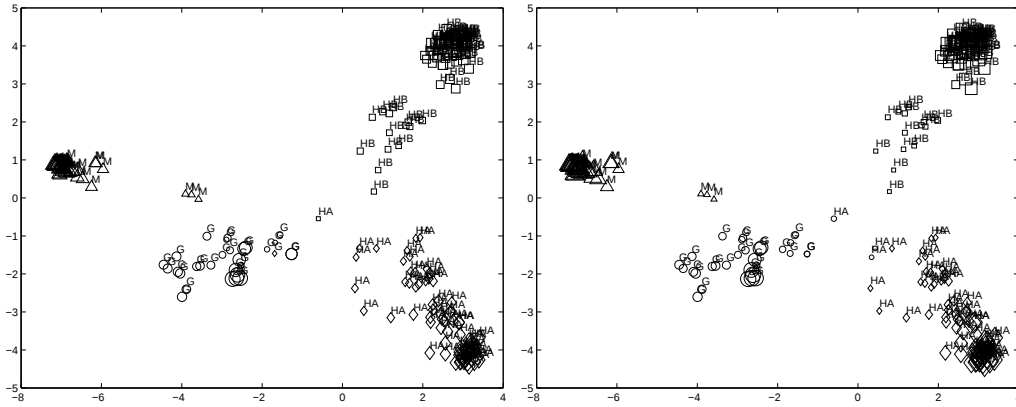


Fig. 8. MDS configuration of the Protein dataset. Each point, represented by a symbol, is assigned to the class with maximum pignistic probability; the size of the symbol is proportional to the maximum of pignistic probability; the labels (HA, HB, M, G) indicate the true class memberships; left: EVCLUS ($\lambda = 0.005$); right: RECM ($\alpha = 0$, $\beta = 1.1$, $\delta^2 = 20$).

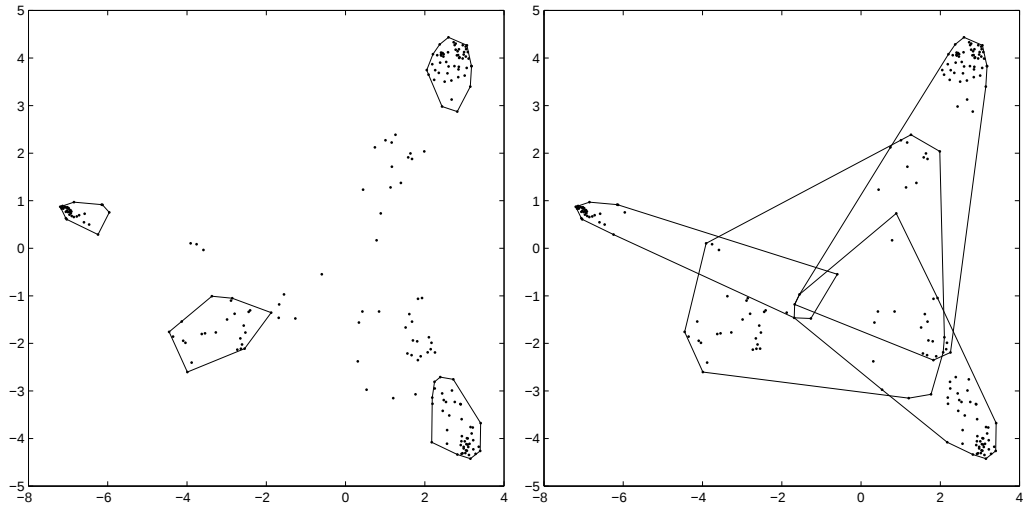


Fig. 9. Protein dataset: lower (left) and upper (right) approximations of the clusters by RECM ($\alpha = 0$, $\beta = 1.1$, $\delta^2 = 20$).

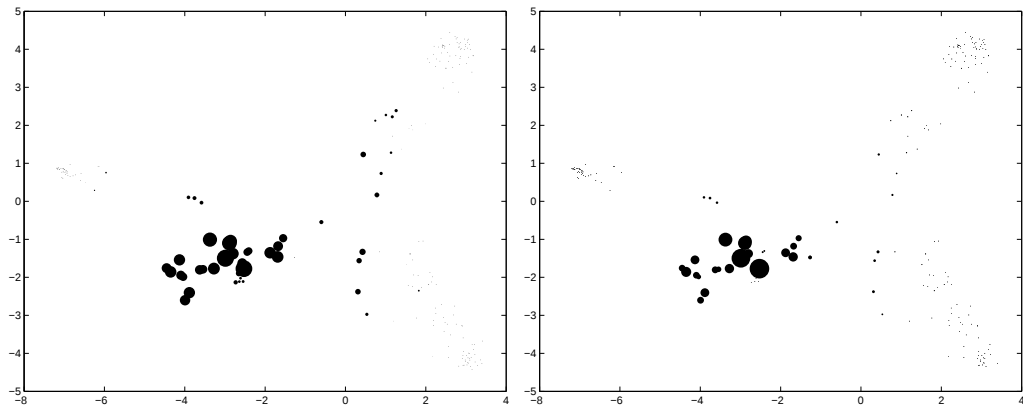


Fig. 10. Protein dataset: mass assigned to the empty set using EVCLUS (left) and RECM (right).

A Belief functions

The Dempster-Shafer theory of evidence (or belief function theory) is a theoretical framework for reasoning with partial and unreliable information. It encompasses different models of reasoning under uncertainty, including the probability and possibility theories as well as Smets' Transferable Belief Model [28]. A complete description can be found in Shafer's book [26].

Let us consider a variable ω taking values in a finite and unordered set Ω called the frame of discernment. Partial knowledge regarding the actual value taken by ω can be represented by a *basic belief assignment* (bba) [26,28], defined as a function m from 2^Ω to $[0, 1]$, verifying:

$$\sum_{A \subseteq \Omega} m(A) = 1. \quad (\text{A.1})$$

The subsets A of Ω such that $m(A) > 0$ are the *focal sets* of m . Each focal set A is a set of possible values for ω , and the number $m(A)$ can be interpreted as a fraction of a unit mass of belief, which is allocated to A on the basis of a given evidential corpus. Complete ignorance corresponds to $m(\Omega) = 1$, whereas perfect knowledge of the value of ω is represented by the allocation of the whole mass of belief to a unique singleton of Ω (m is then called a *certain* bba). When all focal sets of m are singletons, m is equivalent to a probability function, and is called a *Bayesian* bba.

A bba m such that $m(\emptyset) = 0$ is said to be normal. This condition was originally imposed by Shafer [26], but it may be relaxed if one accepts the *open-world assumption* stating that the set Ω might not be complete, and ω might take its value outside Ω [27]. The quantity $m(\emptyset)$ is then interpreted as a mass of belief given to the hypothesis that ω might not lie in Ω .

A bba m can be equivalently represented by a plausibility function $\text{pl} : 2^\Omega \mapsto [0, 1]$, defined as

$$\text{pl}(A) \triangleq \sum_{B \cap A \neq \emptyset} m(B) \quad \forall A \subseteq \Omega. \quad (\text{A.2})$$

The plausibility $\text{pl}(A)$ represents the *potential* amount of support given to A .

The available evidence being modeled in the form of a basic belief assignment, it is often desirable or necessary to make a decision regarding the selection of one single hypothesis in Ω . Smets [28,29] has proposed and justified the use of a probability function for decision making. He has shown that the only transformation of a belief function into a probability function satisfying elementary rationality requirements is the pignistic transformation, in which each mass of belief $m(A)$ is equally distributed among the elements of A . This leads to the concept of pignistic probability BetP defined, for a normal bba,

by:

$$\text{BetP}(\omega) \triangleq \sum_{\{A \subseteq \Omega / \omega \in A\}} \frac{m(A)}{|A|} \quad \forall \omega \in \Omega, \quad (\text{A.3})$$

where $|A|$ denotes the cardinality of $A \subseteq \Omega$. If the bba is subnormal ($m(\emptyset) \neq 0$), then a preliminary normalization step has to be performed. Dempster's normalization consists in dividing all the masses given to nonempty sets by $1 - m(\emptyset)$.

B EVCLUS algorithm

EVCLUS (Evidential Clustering) was the first algorithm suggested to infer a credal partition from *proximity data* [8,9]. It is assumed that the available data consists of a $n \times n$ dissimilarity matrix $\Delta = (\delta_{ii'})$ where $\delta_{ii'}$ represents the dissimilarity between object o_i and object $o_{i'}$. Matrix Δ is only supposed to be symmetric with null diagonal elements. The basic and very intuitive idea of EVCLUS is that, the more similar two objects, the more *plausible* it is that they belong to the same cluster. Let us consider two objects o_i and $o_{i'}$ and two bbas m_i and $m_{i'}$ quantifying our belief in their class membership. It turns out (see [8] for the proof) that the plausibility that these two objects belong to the same class, given m_i and $m_{i'}$, can be computed as one minus the degree of conflict $K_{ii'}$ between m_i and $m_{i'}$ defined by:

$$K_{ii'} = \sum_{A \cap B = \emptyset} m_i(A) m_{i'}(B). \quad (\text{B.1})$$

It is thus possible to define a compatibility criterion between a credal partition M and a proximity matrix which is: the more dissimilar the objects, the higher should be the conflict between their bbas. To derive a credal partition *compatible* with a given dissimilarity matrix, by analogy with multidimensional scaling methods, EVCLUS minimizes an error function inspired from Sammon's stress function [25] defined as

$$J_{\text{EVCLUS}}(M, a, b) \triangleq \frac{1}{C} \sum_{i < i'} \frac{(a K_{ii'} + b - \delta_{ii'})^2}{\delta_{ii'}}, \quad (\text{B.2})$$

where a and b are two coefficients and C is a normalizing constant. This criterion can be minimized with respect to M , a and b using an iterative procedure. To control the model complexity, it was thus suggested to add to the stress function a penalization term that favors "simple", "informative" bbas. The informativeness of each bba m_i is measured through the following

entropy measure:

$$E(m_i) = \sum_{A \in \mathcal{F}(m_i) \setminus \{\emptyset\}} m_i(A) \log_2 \left(\frac{|A|}{m_i(A)} \right) + m_i(\emptyset) \log_2 \left(\frac{|\Omega|}{m_i(\emptyset)} \right), \quad (\text{B.3})$$

where the last term is equal to 0 if $m_i(\emptyset) = 0$. This measure tends to be small when the mass is assigned to few focal sets with small cardinality. Finally, the objective (or stress) function to be minimized is:

$$J'_{\text{EVCLUS}}(M, a, b) \triangleq J_{\text{EVCLUS}}(M, a, b) + \lambda \sum_{i=1}^n E(m_i), \quad (\text{B.4})$$

where λ is the penalization coefficient that controls the extent to which the entropy term influences the solution.