Evidential Clustering: A review*

Thierry Denœux[1] and Orakanya Kanjanatarakul[2]

[1] Sorbonne Universités, Université de Technologie de Compiègne, CNRS, UMR 7253 Heudiasyc, France, email: Thierry.Denoeux@utc.fr

[2] Faculty of Management Sciences, Chiang Mai Rajabhat University, Thailand, email: orakanyaa@gmail.com

Abstract. In evidential clustering, uncertainty about the assignment of objects to clusters is represented by Dempster-Shafer mass functions. The resulting clustering structure, called a credal partition, is shown to be more general than hard, fuzzy, possibilistic and rough partitions, which are recovered as special cases. Three algorithms to generate a credal partition are reviewed. Each of these algorithms is shown to implement a decision-directed clustering strategy. Their relative merits are discussed.

1 Introduction

Clustering is one of the most important tasks in data analysis and machine learning. It aims at revealing some structure in a dataset, so as to highlight groups (clusters) of objects that are similar among themselves, and dissimilar to objects of other groups. Traditionally, we distinguish between *partitional* clustering, which aims at finding a partition of the objects, and *hierarchical* clustering, which finds a sequence of nested partitions.

Over the years, the notion of partitional clustering has been extended to several important variants, including fuzzy [3], possibilistic [12], rough [17] and evidential clustering [8, 9, 15]. Contrary to classical (hard) partitional clustering, in which each object is assigned unambiguously and with full certainty to one and only one cluster, these variants allow ambiguity, uncertainty or doubt in the assignment of objects to clusters. For this reason, they are referred to as *soft* clustering methods, in contrast with classical, *hard* clustering [18].

Among soft clustering paradigms, evidential clustering describes the uncertainty in the membership of objects to clusters using a *Dempster-Shafer mass functions* [20]. Roughly speaking, a mass function can be seen as a collection of sets with corresponding masses. A collection of such mass functions for n objects is called a credal partition. Evidential clustering consists in constructing such a credal partition automatically from the data, by minimizing a cost function.

*This research was supported by the Labex MS2T, which was funded by the French Government, through the program “Investments for the future” by the National Agency for Research (reference ANR-11-IDEX-0004-02).

In this paper, we provide a comprehensive review of evidential clustering algorithms, implemented in the R package `evclust`[3] [7]. Each of the main algorithms to date can be seen as implementing a decision-directed clustering strategy: starting from an initial credal partition and an evidential classifier, the classifier and the partition are updated in turn, until the algorithm has converged to a stable state.

The rest of this paper is structured as follows. In Section 2, the notion of credal partition is first recalled, and some relationships with other clustering paradigms are described. The main evidential clustering algorithms are then reviewed in Section 3. Finally, Section 4 concludes the paper.

2 Credal partition

We first recall the notion of credal partition in Section 2.1. The relation with other clustering paradigms is analyzed in Section 2.2, and the problem of summarizing a credal partition is addressed in Section 2.3.

2.1 Credal partition

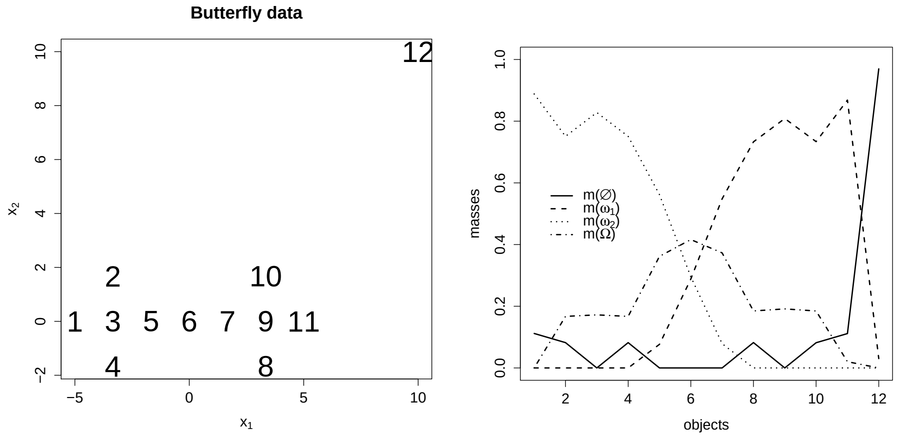
Assume that we have a set $\mathcal{O} = \{o_1, \ldots, o_n\}$ of n objects, each one belonging to one and only one of c groups or clusters. Let $\Omega = \{\omega_1, \ldots, \omega_c\}$ denote the set of clusters. If we know for sure which cluster each object belongs to, we have a (hard) partition of the n objects. Such a partition may be represented by binary variables u_{ik} such that $u_{ik} = 1$ if object o_i belongs to cluster ω_k, and $u_{ik} = 0$ otherwise.

If objects cannot be assigned to clusters with certainty, then we can quantify cluster-membership uncertainty by mass functions $m_1, \ldots, m_n$, where each mass function m_i is a mapping from 2^Ω to $[0,1]$, such that $\sum_{A \subseteq \Omega} m_i(A) = 1$. Each mass $m_i(A)$ is interpreted as a degree of support attached to the proposition "the true cluster of object o_i is in A", and to no more specific proposition. A subset A of Ω such that $m_i(A) > 0$ is called a *focal set* of m_i. The n-tuple $\mathcal{M} = (m_1, \ldots, m_n)$ is called a *credal partition* [9].

Example 1 *Consider, for instance, the "Butterfly" dataset shown in Figure 1(a). Figure 1(b) shows the credal partition with $c = 2$ clusters produced by the Evidential c-means (ECM) algorithm [15]. In this figure, the masses $m_i(\emptyset)$, $m_i(\{\omega_1\})$, $m_i(\{\omega_2\})$ and $m_i(\Omega)$ are plotted as a function of i, for $i = 1, \ldots, 12$. We can see that $m_3(\{\omega_1\}) \approx 1$, which means that object o_3 almost certainly belongs to cluster ω_1. Similarly, $m_9(\{\omega_2\}) \approx 1$, indicating almost certain assignment of object o_9 to cluster ω_2. In contrast, objects o_6 and o_{12} correspond to two different situations of maximum uncertainty. Object o_6 has a large mass assigned to Ω: this reflects ambiguity in the class membership of this object, which means that it might belong to ω_1 as well as to ω_2. The situation is completely different*

[3] This package can be downloaded from the CRAN web site at `https://cran.r-project.org/web/packages`.

for object o_{12}, for which the largest mass is assigned to the empty set, indicating that this object does not seem to belong to any of the two clusters.



(a) (b)

Fig. 1. Butterfly dataset (a) and a credal partition (b).

2.2 Relationships with other clustering paradigms

The notion of credal partition boils down to several alternative clustering structures when the mass functions composing the credal partition have some special forms (see Figure 2).

Hard partition: If all mass functions m_i are *certain* (i.e., have a single focal set, which is a singleton), then we have a hard partition, with $u_{ik} = 1$ if $m_i(\{\omega_k\}) = 1$, and $u_{ik} = 0$ otherwise.

Fuzzy partition: If the m_i are *Bayesian* (i.e., they assign masses only to singletons, in which case the corresponding belief function becomes additive), then the credal partition is equivalent to a fuzzy partition; the degree of membership of object i to cluster k is $u_{ik} = m_i(\{\omega_k\})$.

Fuzzy partition with a noise cluster: A mass function m such that each focal set is either a singleton, or the empty set may be called an *unnormalized Bayesian mass function*. If each mass function m_i is unnormalized Bayesian, then we can define, as before, the membership degree of object i to cluster k a $u_{ik} = m_i(\{\omega_k\})$, but we now have $\sum_{k=1}^{c} u_{ik} \leq 1$, for $i = 1, \ldots, n$. We then have $m_i(\emptyset) = u_{i*} = 1 - \sum_{k=1}^{c} u_{ik}$, which can be interpreted as the degree of membership to a "noise cluster" [5].

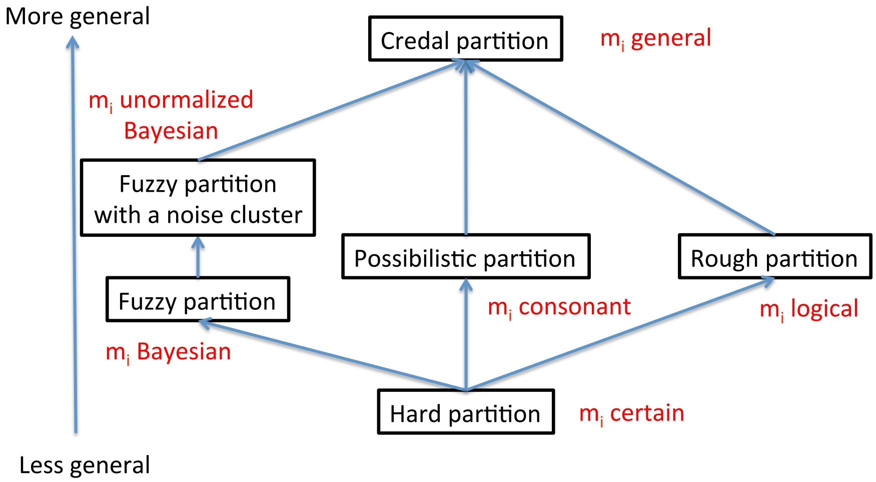


Fig. 2. Relationship between credal partitions and other clustering structures.

Possibilistic partition: If the mass functions m_i are *consonant* (i.e., if their focal sets are nested), then they are uniquely described by their contour functions

$$pl_i(\omega_k) = \sum_{A \subseteq \Omega, \omega_k \in A} m_i(A), \tag{1}$$

which are possibility distributions. We then have a possibilistic partition, with $u_{ik} = pl_i(\omega_k)$ for all i and k. We note that $\max_k pl_i(\omega_k) = 1 - m_i(\emptyset)$.

Rough partition: Assume that each m_i is *logical*, i.e., we have $m_i(A_i) = 1$ for some $A_i \subseteq \Omega$, $A_i \neq \emptyset$. We can then define the *lower approximation* of cluster ω_k as the set of objects that *surely* belong to ω_k,

$$\omega_k^L = \{o_i \in \mathcal{O} | A_i = \{\omega_k\}\}, \tag{2}$$

and the *upper approximation* of cluster ω_k as the set of objects that *possibly* belong to ω_k,

$$\omega_k^U = \{o_i \in \mathcal{O} | \omega_k \in A_i\}. \tag{3}$$

The membership values to the lower and upper approximations of cluster ω_k are then, respectively, $\underline{u}_{ik} = Bel_i(\{\omega_k\})$ and $\overline{u}_{ik} = Pl_i(\{\omega_k\})$. If we allow $A_i = \emptyset$ for some i, then we have $\overline{u}_{ik} = 0$ for all k, which means that object o_i does not belong to the upper approximation of any cluster.

2.3 Summarization of a credal partition

A credal partition is a quite complex clustering structure, which often needs to be summarized in some way to become interpretable by the user. This can be achieved by transforming each of the mass functions in the credal partition into a simpler representation. Depending on the representation used, each of clustering structures mentioned in Section 2.2 can be recovered as different partial views of a credal partition. Some of the relevant transformations are discussed below.

Fuzzy and hard partitions: A fuzzy partition can be obtained by transforming each mass function m_i into a probability distribution p_i using the plausibility-probability transformation defined as

$$p_i(\omega_k) = \frac{pl_i(\omega_k)}{\sum_{\ell=1}^{c} pl_i(\omega_\ell)}, \quad k = 1, \ldots, c, \tag{4}$$

where pl_i is the contour function associated to m_i, given by (1). By selecting, for each object, the cluster with maximum probability, we then get a hard partition.

Fuzzy partition with noise cluster: In the plausibility-probability transformation (4), the information contained in the masses $m_i(\emptyset)$ assigned to the empty set is lost. However, this information may be important if the dataset contains outliers. To keep track of it, we can define an unnormalized plausibility transformation as $\pi_i(\omega_k) = (1 - m_i(\emptyset))p_i(\omega_k)$, for $k = 1, \ldots, c$. The degree of membership of each object i to cluster k can then be defined as $u_{ik} = \pi_i(\omega_k)$ and the degree of membership to the noise cluster as $u_{i*} = m_i(\emptyset)$.

Possibilistic partition: A possibilistic partition can be obtained from a credal partition by computing a consonant approximation of each of the mass functions m_i [11]. The simplest approach is to approximate m_i by the consonant mass function with the same contour function, in which case the degree of possibility of object o_i belonging to cluster ω_k is $u_{ik} = pl_i(\omega_k)$.

Rough partition: As explained in Section 2.2, a credal partition becomes equivalent to a rough partition when all mass functions m_i are logical. A general credal partition can thus be transformed into a rough partition by deriving a set A_i of clusters from each mass function m_i. This can be done by selecting a focal set A_i such that $m_i(A_i) \geq m_i(A)$ for any subset A of Ω, as suggested in [15]. Alternatively, we can use the following *interval dominance decision rule*, and select the set A_i^* of clusters whose plausibility exceeds the degree of belief of any other cluster,

$$A_i^* = \{\omega \in \Omega | \forall \omega' \in \Omega, pl_i^*(\omega) \geq m_i^*(\{\omega'\})\}, \tag{5}$$

where pl_i^* and m_i^* are the normalized contour and mass functions defined, respectively, by $pl_i^* = pl_i/(1 - m_i(\emptyset))$ and $m_i^* = m_i/(1 - m_i(\emptyset))$. If the interval dominance rule is used, we may account for the mass assigned to the empty set by defining A_i as follows,

$$A_i = \begin{cases} \emptyset & \text{if } m_i(\emptyset) = \max_{A \subseteq \Omega} m_i(A) \\ A_i^* & \text{otherwise.} \end{cases} \tag{6}$$

3 Review of evidential clustering algorithms

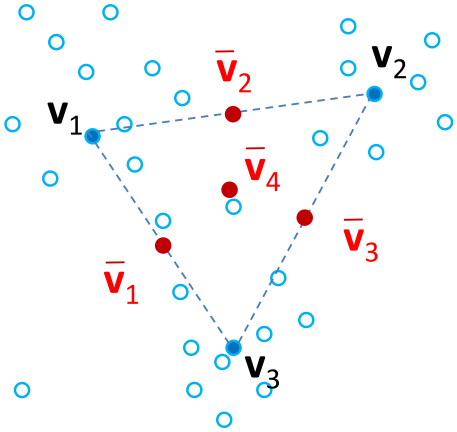
Three main algorithms have been proposed to generate credal partitions: the Evidential c-means (ECM) [15,16], EK-NNclus [8], and EVCLUS [9,10]. These algorithms are described in the next sections.

3.1 Evidential c-means

In contrast to EVCLUS, the Evidential c-means algorithm (ECM) [15] is a prototype-based clustering algorithm, which generalizes the hard and fuzzy c-means (FCM) algorithm. The method is suitable to cluster attribute data. As in FCM, each cluster ω_k is represented by a prototype $\boldsymbol{v}_k$ in the attribute space. However, in contrast with FCM, each non-empty set of clusters $A_j \subseteq \Omega$ is also represented by a prototype $\overline{\boldsymbol{v}}_j$, which is defined as the center of mass of all prototypes $\boldsymbol{v}_k$, for $\omega_k \in A_k$ (Figure 3). Formally,

$$\overline{\boldsymbol{v}}_j = \frac{1}{c_j} \sum_{k=1}^{c} s_{kj} \boldsymbol{v}_k, \tag{7}$$

where $c_j = |A_j|$ denotes the cardinality of A_j, and $s_{kj} = 1$ if $\omega_k \in A_j$, $s_{kj} = 0$ otherwise.



$\overline{\mathbf{v}}_2$
$\mathbf{v}_2$
$\mathbf{v}_1$
$\overline{\mathbf{v}}_4$
$\overline{\mathbf{v}}_3$
$\overline{\mathbf{v}}_1$
$\mathbf{v}_3$

Fig. 3. Representation of sets of clusters by prototypes in the ECM algorithm.

Let Δ_{ij} denote the distance between a vector $\boldsymbol{x}_i$ and prototype $\overline{\boldsymbol{v}}_j$, and let the distance between any vector $\boldsymbol{x}_i$ and the empty set by defined as a fixed value δ. The ECM algorithm is based on the idea that $m_{ij} = m_i(A_j)$ should be high if $\boldsymbol{x}_i$ is close to $\overline{\boldsymbol{v}}_j$, i.e. if Δ_{ij} is small. Furthermore, if $\boldsymbol{x}_i$ is far from all prototypes

$\overline{\boldsymbol{v}}_j$, then $m_{i\emptyset} = m_i(\emptyset)$ should be large. Such a configuration of mass functions and prototypes can be achieved by minimizing the following cost function,

$$J_{\text{ECM}}(\mathcal{M}, V) = \sum_{i=1}^{n} \sum_{\{j \mid A_j \neq \emptyset, A_j \subseteq \Omega\}} c_j^{\alpha} m_{ij}^{\beta} \Delta_{ij}^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^{\beta}, \tag{8}$$

subject to

$$\sum_{\{j \mid A_j \subseteq \Omega, A_j \neq \emptyset\}} m_{ij} + m_{i\emptyset} = 1 \quad \forall i = 1, n, \tag{9}$$

where $\mathcal{M}$ is the credal partition and $V = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_c)$ is the matrix of prototypes. This cost function depends on three coefficients: β controls the hardness of the evidential partition as in the FCM algorithm; δ controls the amount of data considered as outliers, as in the Davé's Noise Clustering algorithm [5]; finally parameter α controls the specificity of the evidential partition, larger values of α penalizing subsets of clusters with large cardinality.

As in FCM, the minimization of the cost function J_{ECM} can be achieved by alternating two steps: (1) minimize $J_{\text{ECM}}(\mathcal{M}, V)$ with respect to $\mathcal{M}$ for fixed V, and (2) minimize $J_{\text{ECM}}(\mathcal{M}, V)$ with respect to V for fixed $\mathcal{M}$. The first step is achieved by the following update equations,

$$m_{ij} = \frac{c_j^{-\alpha/(\beta-1)} \Delta_{ij}^{-2/(\beta-1)}}{\sum_{A_k \neq \emptyset} c_k^{-\alpha/(\beta-1)} \Delta_{ik}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}}, \tag{10}$$

for $i = 1, \ldots, n$ and for all j such that $A_j \neq \emptyset$, and

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij}, \tag{11}$$

for $i = 1, \ldots, n$. The second step implies solving a system of the form $HV = B$, where B is the matrix of size $c \times p$ with general term

$$B_{lq} = \sum_{i=1}^{n} x_{iq} \sum_{A_j \ni \omega_l} c_j^{\alpha-1} m_{ij}^{\beta} \tag{12}$$

and H the matrix of size $c \times c$ given by:

$$H_{lk} = \sum_{i} \sum_{A_j \supseteq \{\omega_k, \omega_l\}} c_j^{\alpha-2} m_{ij}^{\beta}. \tag{13}$$

We can observe that Eqs (10)-(11) define an evidential classifier: given the matrix V of prototypes, they make it possible to compute a mass function for any new instance. The prototype-updating step can then be seen as a training phase, where the classifier is fitted to the data. ECM can thus be seen as a decision-directed clustering algorithm.

The Relational Evidential c-Means (RECM), a version of ECM for dissimilarity data, was introduced in [16]. In this version, we assume that the data consist in a square matrix $D = (d_{ij})$ of dissimilarities between n objects, so that ECM cannot be used directly. However, if we assume the dissimilarities d_{ij} to be *metric*, i.e., to be squared Euclidean distances in some attribute space, we can still compute the distances Δ_{ij} in (8) without explicitly computing the vectors $\boldsymbol{x}_i$ and $\boldsymbol{v}_k$, which allows us to find a credal partition $\mathcal{M}$ minimizing (8). Although the convergence of RECM is not guaranteed when the dissimilarities are not metric, the algorithm has been shown to be quite robust to violations of this assumption.

3.2 EK-NNclus

The EK-NNclus algorithm [8] is another decision-directed clustering procedure based on the evidential k-nearest neighbor (EK-NN) rule [6]. The EK-NN rule works as follows. Consider a classification problem in which an object o has to be classified in one of c groups, based on its distances to n objects in a dataset. Let $\Omega = \{\omega_1, \ldots, \omega_c\}$ be the set of groups, and d_j the distance between the object to be classified and object o_j in the dataset. The knowledge that object o is at a distance d_j from o_j is a piece of evidence that can be represented by the following mass function on Ω,

$$m_j(\{\omega_k\}) = u_{jk}\varphi(d_j), \quad k = 1, \ldots, c \tag{14a}$$

$$m_j(\Omega) = 1 - \varphi(d_j), \tag{14b}$$

where φ is a non-increasing mapping from $[0, +\infty)$ to $[0, 1]$, and $u_{jk} = 1$ if o_j belongs to class ω_k, $u_{jk} = 0$ otherwise. In [6], it was proposed to choose φ as $\varphi(d_j) = \alpha_0 \exp(-\gamma d_j)$ for some constants α_0 and γ. Denoting by N_K the set of indices of the K nearest neighbors of object o is the learning set, the K mass function m_j, $j \in N_K$ are then combined by Dempster's rule [20] to yield the combined mass function

$$m = \bigoplus_{j \in N_K} m_j. \tag{15}$$

A decision can finally be made by assigning object o to the class ω_k with the highest plausibility. We can remark that, to make a decision, we need not compute the combined mass function m explicitly. The contour function pl_j corresponding to m_j in (14) is

$$pl_j(\omega_\ell) = (1 - \varphi(d_j))^{1-u_{j\ell}}, \tag{16}$$

for $\ell = 1, \ldots, c$. The combined contour function is thus

$$pl(\omega_\ell) \propto \prod_{j \in N_K} (1 - \varphi(d_j))^{1-u_{j\ell}}, \tag{17}$$

for $\ell = 1, \ldots, c$. Its logarithm can be written as

$$\ln pl(\omega_\ell) = \sum_{j=1}^{n} w_j u_{j\ell} + C, \tag{18}$$

where C is a constant, and $w_j = -\ln(1-\varphi(d_j))$ if $j \in N_K$, and $w_j = 0$ otherwise.

The EK-NNclus algorithm implements a decision-directed approach, using the above EK-NN rule as the base classifier. We start with a matrix $D = (d_{ij})$ of dissimilarities between n objects. To initialize the algorithm, the objects are labeled randomly (or using some prior knowledge if available). As the number of clusters is usually unknown, it can be set to $c = n$, i.e., we initially assume that there are as many clusters as objects and each cluster contains exactly one object. If n is very large, we can give c a large value, but smaller than n, and initialize the object labels randomly. As before, we define cluster-membership binary variables u_{ik} as $u_{ik} = 1$ is object o_i belongs to cluster k, and $u_{ik} = 0$ otherwise. An iteration of the algorithm then consists in updating the object labels in some random order, using the EKNN rule. For each object o_i, we compute the logarithms of the plausibilities of belonging to each cluster (up to an additive constant) using (18) as

$$s_{ik} = \sum_{j \in N_K(i)} w_{ij} u_{jk}, \quad k = 1, \ldots, c, \tag{19}$$

where $w_{ij} = -\ln(1 - \varphi(d_{ij}))$ and $N_K(i)$ is the set of indices of the K nearest neighbors of object o_i in the dataset. We then assign object o_i to the cluster with the highest plausibility, i.e., we update the variables u_{ik} as

$$u_{ik} = \begin{cases} 1 & \text{if } s_{ik} = \max_{k'} s_{ik'}, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

If the label of at least one object has been changed during the last iteration, then the objects are randomly re-ordered and a new iteration is started. Otherwise, we move to the last step described below, and the algorithm is stopped. We can remark that, after each iteration, some clusters may have disappeared. To save computation time and storage space, we can update the number c of clusters, renumber the clusters from 1 to c, and change the membership variables s_{ik} accordingly, after each iteration. After the algorithm has converged, we can compute the final mass functions

$$m_i = \bigoplus_{j \in N_K(i)} m_{ij}, \tag{21}$$

for $i = 1, \ldots, n$, where each m_{ij} is the following mass function,

$$m_{ij}(\{\omega_k\}) = u_{jk}\varphi(d_{ij}), \quad k = 1, \ldots, c \tag{22a}$$

$$m_{ij}(\Omega) = 1 - \varphi(d_{ij}). \tag{22b}$$

As compared to EVCLUS, EK-NNclus yields a credal partition with simpler mass functions, whose focal sets are the singletons and Ω. A major advantage of EK-NNclus is that it does not require the number of clusters to be fixed in advance. Heuristics for tuning the two parameters of the algorithm, K and γ, are described in [8]. Also, EK-NNclus is applicable to non-metric dissimilarity data.

3.3 EVCLUS

The EVCLUS algorithm [9,10] applies some ideas from Multidimensional Scaling (MDS) [4] to clustering. Let $D = (d_{ij})$ be an $n \times n$ dissimilarity matrix, where d_{ij} denotes the dissimilarity between objects o_i and o_j. To derive a credal partition $\mathcal{M} = (m_1, \ldots, m_n)$ from D, we assume that the plausibility pl_{ij} that two objects o_i and o_j belong to the same class is a decreasing function of the dissimilarity d_{ij}: the more similar are two objects, the more plausible it is that they belong to the same cluster. Now, it can be shown [10] that the plausibility pl_{ij} is equal to $1 - \kappa_{ij}$, where κ_{ij} is the degree of conflict between m_i and m_j. The credal partition $\mathcal{M}$ should thus be determined in such a way that similar objects o_i and o_j have mass functions m_i and m_j with low degree of conflict, whereas highly dissimilar objects are assigned highly conflicting mass functions. This can be achieved by minimizing the discrepancy between the pairwise degrees of conflict and the dissimilarities, up to some increasing transformation. In [10], we proposed to minimize the following stress function,

$$J(\mathcal{M}) = \eta \sum_{i<j} (\kappa_{ij} - \delta_{ij})^2, \tag{23}$$

where $\eta = \left(\sum_{i<j} \delta_{ij}^2 \right)^{-1}$ is a normalizing constant, and the $\delta_{ij} = \varphi(d_{ij})$ are transformed dissimilarities, for some fixed increasing function φ from $[0, +\infty)$ to $[0, 1]$. A suitable choice for φ is a soft threshold function, such as $\varphi(d) = 1 - \exp(-\gamma d^2)$, where γ is a user-defined parameter. A heuristic for fixing γ is described in [10]. The stress function (23) by the Iterative Row-wise Quadratic Programming (IRQP) [10, 21]. The IRQP algorithm consists in minimizing (23) with respect to each mass function m_i at a time, leaving the other mass functions m_j fixed. At each iteration, we thus solve

$$\min_{m_i} \sum_{j \neq i} (\kappa_{ij} - \delta_{ij})^2, \tag{24}$$

such that $m_i(A) \geq 0$ for any $A \subseteq \Omega$ and $\sum_{A \subseteq \Omega} m(A) = 1$, which is a linearly constrained positive least-square problem that can be solved efficiently. We can remark that this algorithm can be seen as a decision-directed procedure, where each object o_i is classified at each step, using its distances to all the other objects. The IRQP algorithm has been shown to be much faster than the gradient procedure, and to reach lower values of the stress function.

A major drawback of the EVCLUS algorithm as originally proposed in [9] is that it requires to store the whole dissimilarity matrix D, which precludes its application to very large datasets. However, there is usually some redundancy in a dissimilarity matrix. In particular, if two objects o_1 and o_2 are very similar, then any object o_3 that is dissimilar from o_1 is usually also dissimilar from o_2. Because of such redundancies, it might be possible to compute the differences between degrees of conflict and dissimilarities, for *only a subset of randomly sampled dissimilarities.* More precisely, let $j_1(i), \ldots, j_k(i)$ be k integers sampled at random from the set $\{1, \ldots, i-1, i+1, \ldots, n\}$, for $i = 1, \ldots, n$. Let J_k the following stress criterion,

$$J_k(\mathcal{M}) = \eta \sum_{i=1}^{n} \sum_{r=1}^{k} (\kappa_{i,j_r(i)} - \delta_{i,j_r(i)})^2, \tag{25}$$

where, as before, η is a normalizing constant. Obviously, $J(\mathcal{M})$ is recovered as a special case when $k = n-1$. However, in the general case, the calculation of $J_k(\mathcal{M})$ requires only $O(nk)$ operations. If k can be kept constant as n increases, or, at least, if k increases slower than linearly with n, then significant gains in computing time and storage requirement could be achieved [10].

4 Conclusions

The notion of credal partition, as well as its relationships with alternative clustering paradigms have been reviewed. Basically, each of the alternative partitional clustering structures (i.e., hard, fuzzy, possibilistic and rough partitions) correspond to a special form of the mass functions within a credal partition. A credal partition can be transformed into a simpler clustering structure for easier interpretation. Recently, evidential clustering has been successfully applied in various domains such as machine prognosis [19], medical image processing [13, 14] and analysis of social networks [22]. Three main algorithms for generating credal partitionsand implemented in the R package `evclust` have been reviewed. Each of these three algorithms have their strengths and limitations, and the choice of an algorithm depends on the problem at hand. Both ECM and EK-NN are very efficient for handling attribute data. EK-NN has the additional advantage that it can determine the number of clusters automatically, while EVCLUS and ECM produce more informative outputs (with masses assigned to any subsets of clusters). EVCLUS was shown to be very effective for dealing with non metric dissimilarity data, and the recent improvements reported in [10] make it suitable to handle very large datasets. Methods for exploiting additional knowledge in the form of pairwise constraints have been studied in [1, 2], and the problem of handling a large number of clusters has been addressed in [10].

In future work, it will be interesting to performed detailed comparative experiments with these algorithms using a wide range of attribute and dissimilarity datasets. Such a study will require the definition of performance indices to measure the fit between a credal partition and a hard partition, or between two credal partition. This approach should provide guidelines for choosing a suitable algorithm, depending on the characteristics of the clustering problem.

References

1. V. Antoine, B. Quost, M.-H. Masson, and T. Denoeux. CEVCLUS: evidential clustering with instance-level constraints for relational data. *Soft Computing*, 18(7):1321–1335, 2014.

2. V. Antoine, B. Quost, M.-H. Masson, and T. Denoeux. CECM: Constrained evidential c-means algorithm. *Computational Statistics & Data Analysis*, 56(4):894–914, 2012.
3. J. Bezdek. *Pattern Recognition with fuzzy objective function algorithm.* Plenum Press, New-York, 1981.
4. I. Borg and P. Groenen. *Modern multidimensional scaling.* Springer, New-York, 1997.
5. R. Davé. Characterization and detection of noise in clustering. *Pattern Recognition Letters*, 12:657–664, 1991.
6. T. Denœux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Trans. on Systems, Man and Cybernetics*, 25(05):804–813, 1995.
7. T. Denœux. *evclust: Evidential Clustering*, 2016. R package version 1.0.2.
8. T. Denœux, O. Kanjanatarakul, and S. Sriboonchitta. EK-NNclus: a clustering procedure based on the evidential k-nearest neighbor rule. *Knowledge-based Systems*, 88:57–69, 2015.
9. T. Denœux and M.-H. Masson. EVCLUS: Evidential clustering of proximity data. *IEEE Trans. on Systems, Man and Cybernetics B*, 34(1):95–109, 2004.
10. T. Denœux, S. Sriboonchitta, and O. Kanjanatarakul. Evidential clustering of large dissimilarity data. *Knowledge-based Systems*, 106:179–195, 2016.
11. D. Dubois and H. Prade. Consonant approximations of belief measures. *International Journal of Approximate Reasoning*, 4:419–449, 1990.
12. R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Trans. on Fuzzy Systems*, 1:98–111, May 1993.
13. B. Lelandais, S. Ruan, T. Denœux, P. Vera, and I. Gardin. Fusion of multi-tracer PET images for dose painting. *Medical Image Analysis*, 18(7):1247–1259, 2014.
14. N. Makni, N. Betrouni, and O. Colot. Introducing spatial neighbourhood in evidential c-means for segmentation of multi-source images: Application to prostate multi-parametric MRI. *Information Fusion*, 19:61–72, 2014.
15. M.-H. Masson and T. Denoeux. ECM: an evidential version of the fuzzy c-means algorithm. *Pattern Recognition*, 41(4):1384–1397, 2008.
16. M.-H. Masson and T. Denœux. RECM: relational evidential c-means algorithm. *Pattern Recognition Letters*, 30:1015–1026, 2009.
17. G. Peters. Is there any need for rough clustering? *Pattern Recognition Letters*, 53:31–37, 2015.
18. G. Peters, F. Crespo, P. Lingras, and R. Weber. Soft clustering: fuzzy and rough approaches and their extensions and derivatives. *International Journal of Approximate Reasoning*, 54(2):307–322, 2013.
19. L. Serir, E. Ramasso, and N. Zerhouni. Evidential evolving Gustafson-Kessel algorithm for online data streams partitioning using belief function theory. *International Journal of Approximate Reasoning*, 53(5):747–768, 2012.
20. G. Shafer. *A mathematical theory of evidence.* Princeton University Press, Princeton, N.J., 1976.
21. C. J. ter Braak, Y. Kourmpetis, H. A. Kiers, and M. C. Bink. Approximating a similarity matrix by a latent class model: A reappraisal of additive fuzzy clustering. *Computational Statistics & Data Analysis*, 53(8):3183–3193, 2009.
22. K. Zhou, A. Martin, Q. Pan, and Z.-G. Liu. Median evidential c-means algorithm and its application to community detection. *Knowledge-Based Systems*, 74(0):69–88, 2015.