Evidential clustering of large dissimilarity data

Thierry Denœux*[1], Songsak Sriboonchitta† and
Orakanya Kanjanatarakul‡

* Sorbonne Universités
Université de Technologie de Compiègne, CNRS,
UMR 7253 Heudiasyc, France
† Faculty of Economics, Chiang Mai University, Thailand
‡ Faculty of Management Sciences,
Chiang Mai Rajabhat University, Thailand

May 24, 2016

[1]Corresponding author. Phone: +33 344 234 496, fax: +33 344234477, email: tdenoeux@utc.fr.

Abstract

In evidential clustering, the membership of objects to clusters is considered to be uncertain and is represented by Dempster-Shafer mass functions, forming a credal partition. The EVCLUS algorithm constructs a credal partition in such a way that larger dissimilarities between objects correspond to higher degrees of conflict between the associated mass functions. In this paper, we present several improvements to EVCLUS, making it applicable to very large dissimilarity data. First, the gradient-based optimization procedure in the original EVCLUS algorithm is replaced by a much faster iterative row-wise quadratic programming method. Secondly, we show that EVCLUS can be provided with only a random sample of the dissimilarities, reducing the time and space complexity from quadratic to roughly linear. Finally, we introduce a two-step approach to construct credal partitions assigning masses to selected pairs of clusters, making the algorithm outputs more informative than those of the original EVCLUS, while remaining manageable for large numbers of clusters.

Keywords: Dempster-Shafer theory, evidence theory, belief functions, unsupervised learning, credal partition, relational data, proximity data, pairwise data.

1 Introduction

Clustering data into groups is one of the fundamental tasks in data mining and machine learning. Clustering algorithms can be distinguished according to the input data they can process, and according to the outputs they produce.

Typically, two categories of input data are considered: attribute (vectorial) data and dissimilarity (proximity, relational, pairwise) data. In the former case, each object is described by a vector of numerical or categorical attributes. In the latter, the data takes the form of a matrix of dissimilarities between objects. Attribute data can be easily transformed into dissimilarity data by choosing a suitable distance. The inverse transformation (from dissimilarity to attribute data) is generally more difficult, except in the special case of metric dissimilarities, i.e., dissimilarities that are exact Euclidean distances between vectors in a latent space, a case not so frequent in practice. Finding an attribute representation of a set of objects, such that distances between objects approximate a given dissimilarity matrix is often a difficult task (referred to as multidimensional scaling – MDS), which requires to solve a large scale nonlinear optimization problem [3, 4]. Most clustering algorithms, such as the c-means algorithms and its numerous variants, are designed to handle attribute data. A smaller number of algorithms, referred to as *relational clustering* methods, can directly handle dissimilarity data [9–11].

As for the clustering outputs, we can 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 [2] and possibilistic [16] clustering, and more recently, rough [20, 27] and evidential [7, 25] clustering. Contrary to classical (hard) partitional clustering, in which each object is assigned unambiguously and with full certainty to a single cluster, these variants allow for ambiguity, uncertainty or doubt in the assignment of objects to clusters. For this reason, they are referred to as "soft" clustering methods [28], in contrast with classical, "hard" clustering. Among soft clustering paradigms, *evidential clustering* describes the uncertainty in the membership of each object to clusters using a Dempster-Shafer mass function [30], which assigns a mass to each subset of clusters. This is a rich and informative description of the clustering structure of a data set, which can be shown to include hard, fuzzy and rough partitions as special cases. Recently, evidential clustering has been successfully applied in various domains such as machine prognosis [29], medical

image processing [17, 24] and analysis of social networks [34]. Similar ideas have also been exploited in supervised classification (see, e.g., [18, 21, 22]).

In [7], one of us (the first author) introduced EVCLUS, an evidential clustering algorithm that handles (non necessarily metric) dissimilarity data. EVCLUS is based on the natural assumption that the plausibility of two objects belonging to the same cluster is higher when the two objects are more similar. This assumption translates into the search for a credal partition minimizing a cost function. A variant of EVCLUS allowing one to use prior knowledge in the form of pairwise constraints was later introduced in [1].

The EVCLUS algorithm has several advantages. It is conceptually simple and it can handle non metric dissimilarity data (even expressed on an ordinal scale). It was also shown to outperform some of the state-of-the-art relational clustering techniques on a number of datasets [7]. On the minus side, the main drawback of EVCLUS is its computational complexity. As other relational clustering algorithms, it requires to store the whole dissimilarity matrix; the space complexity is thus $O(n^2)$, where n is the number of objects, which precludes application to datasets containing more than a few thousand objects. Furthermore, each iteration of the gradient-based optimization procedure implemented in the EVCLUS algorithm requires $O(f^3 n^2)$ operations, where f is the number of focal sets of the mass functions, i.e., the number of subsets of clusters being considered. In the worst case, $f = 2^c$, where c is the number of clusters. To make the method usable even for moderate values of c, we need to restrict the form of the mass functions so that masses are only assigned to focal sets of size 0, 1 or c, which prevents us from fully exploiting the potential generality of the method.

In this paper, we propose some improvements to the EVCLUS algorithm, making it applicable to very large datasets. These improvements are threefold. First, the gradient-based optimization procedure in the original EVCLUS algorithm is replaced by an adaptation of the much faster iterative row-wise quadratic programming method proposed in [31]. Secondly, we show that EVCLUS does not need to be provided with the whole dissimilarity matrix, reducing the time and space complexity from quadratic to roughly linear. Finally, we introduce a two-step approach to construct credal partitions assigning masses to selected pairs of clusters, making the algorithm outputs more informative than those of the original EVCLUS, while remaining manageable for large numbers of clusters.

The rest of the paper is organized as follows. The background on belief functions, evidential clustering and the EVCUS algorithm will first be recalled in Section 2. The new optimization procedure will be described and

evaluated in Section 3. Improvements of EVCLUS making it applicable to problems with large numbers of objects and large numbers of clusters will then be described, respectively, in Sections 4 and 5. Finally, Section 6 will conclude the paper.

2 Background

In this section, a brief reminder on Dempster-Shafer theory will first be provided in Section 2.1. Credal partitions and related necessary notions will then be recalled in Section 2.2, and the EVCLUS algorithm will be presented in Section 2.3.

2.1 Mass functions

Let $\Omega = \{\omega_1, \ldots, \omega_c\}$ be a finite set representing the possible answers to some question Q, one and only one of which is true. The true answer is denoted by $\boldsymbol{\omega}$. A *mass function* Ω is a mapping from the power set 2^Ω to $[0, 1]$ such that

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

Each number $m(A)$ represents the degree of support attached to the proposition $\boldsymbol{\omega} \in A$, and to no more specific proposition [30]. The subsets A of Ω such that $m_i(A) > 0$ are called the *focal sets* of m. A mass function m is said to be

- *normalized* if $\emptyset$ is not a focal set;
- *logical* if it has only one focal set;
- *Bayesian* if its focal sets are singletons;
- *certain* if it is both logical and Bayesian, i.e., if it has only one focal set, and this focal set is a singleton;
- *consonant* if its focal sets are nested.

To each mass function m, we may associate belief and plausibility functions from 2^Ω to $[0, 1]$ defined, respectively, as follows,

$$Bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B) \tag{2a}$$

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \tag{2b}$$

for all $A \subseteq \Omega$. These two functions are linked by the relation $Pl(A) = Bel(\Omega) - Bel(\overline{A})$, for all $A \subseteq \Omega$. The quantity $Bel(A)$ is a measure of how much the proposition "$\boldsymbol{\omega} \in A$" is supported by the available evidence. In contrast $Bel(\Omega) - Pl(A) = Bel(\overline{A})$ is a measure of how much the complementary hypothesis $\overline{A}$ is supported, so that $Pl(A)$ can be seen as a measure of lack of support for $\overline{A}$. The function $pl : \Omega \to [0, 1]$ that maps each element ω of Ω to its plausibility $pl(\omega) = Pl(\{\omega\})$ is called the *contour function* associated to m.

If m is Bayesian, then $Bel = Pl$, and this function is a probability measure; the contour function is thus the usual probability mass function, i.e., $Bel(A) = Pl(A) = \sum_{\omega \in A} pl(\omega)$ for all $A \subseteq \Omega$. If m is consonant, then Pl is a possibility measure, i.e., we have $Pl(A \cup B) = \max(Pl(A), Pl(B))$ for all $A, B \subseteq \Omega$, and Bel is the dual necessity measure; pl is then the corresponding possibility distribution, i.e., $Pl(A) = \max_{\omega \in A} pl(\omega)$ for all $A \subseteq \Omega$. A consonant mass function can be uniquely recovered from its contour function.

Let m_1 and m_2 be two mass functions defined on the same set Ω. Their *degree of conflict* [30] is defined as

$$\kappa = \sum_{A \cap B = \emptyset} m_1(A) m_2(B). \tag{3}$$

It is comprised between 0 and 1. When m_1 and m_2 are two mass functions representing two independent pieces of evidence about the same question, κ is interpreted as a measure of conflict between these two pieces of evidence. A different interpretation of κ was provided in [7], for the case where m_1 and m_2 represent independent pieces of evidence about two different questions Q_1 and Q_2, with the same set of possible answers Ω: in that case, $1 - \kappa$ is the plausibility that the true answers to Q_1 and Q_2 are identical.

Example 1 *Let us assume that the questions of interest concern the nationalities of Ann and Henri. Let* $\Omega = \{\text{Singapore}, \text{Thailand}, \text{France}, \text{Canada}\}$ *be the set of possible answers to both questions. We receive some evidence that Ann comes from an Asian country, with probability 0.8, and independent evidence that Henri originates from a country where French is an official language, with probability 0.5. What is the plausibility that Ann and Henri have the same nationality? The two pieces of evidence translate into the following mass functions*

$$m_1(\{\text{Singapore}, \text{Thailand}\}) = 0.8, \quad m_1(\Omega) = 0.2, \tag{4a}$$

$$m_2(\{\text{France}, \text{Canada}\}) = 0.5, \quad m_2(\Omega) = 0.5. \tag{4b}$$

The degree of conflict between m_1 and m_2 is

$$\kappa = m_1(\{\text{Singapore}, \text{Thailand}\})m_2(\{\text{France}, \text{Canada}\}) \tag{5a}$$
$$= 0.8 \times 0.5 = 0.4. \tag{5b}$$

The requested plausibility is thus $1 - 0.4 = 0.6$. □

Assume now that our state of knowledge about $\boldsymbol{\omega}$ is described by a mass function m, and we need to pick one or several elements of Ω as our best guess about $\boldsymbol{\omega}$. This is the *decision* problem, to which several solutions have been proposed. A simple solution is to pick the element ω^* with the highest plausibility,

$$\omega^* = \arg\max_{\omega \in \Omega} pl(\omega). \tag{6}$$

This rule yields a precise decision, i.e., it picks a single element in Ω. However, as a mass function assigns masses to subsets of ω, and is thus less precise than a probability distribution, it can be argued that it cannot always provide the basis for a precise decision. Another decision rule allowing for ambiguity is the following. We say that hypothesis ω is strictly preferable to, or dominates, ω', iff $Bel(\{\omega\}) > Pl(\{\omega'\})$, which we denote by $\omega \succ \omega'$. Relation $\succ$ is a partial order on Ω. We then consider the set of its maximal (non dominated) elements,

$$\Omega^* = \{\omega \in \Omega | \forall \omega' \in \Omega, Bel(\{\omega'\}) \leq Pl(\{\omega\})\}. \tag{7}$$

In this approach, we no longer achieve a single decision, but a set of potential decisions. This ambiguity in the decision is a consequence of the ambiguity of assigning masses to sets, and not to elements of Ω. This decision rule will be referred to as the *interval dominance rule*.

Example 2 *Let* $\Omega = \{\omega_1, \omega_2, \omega_3\}$ *and consider the following mass function,*

$$m(\{\omega_1\}) = 0.3, m(\{\omega_2\}) = 0.4, m(\{\omega_1, \omega_3\}) = 0.3. \tag{8}$$

The contour function is $pl(\omega_1) = 0.6$, $pl(\omega_2) = 0.4$, $pl(\omega_3) = 0.3$, *hence* $\omega^* = \omega_1$. *Furthermore, we have* $Bel(\{\omega_1\}) = 0.3$, $Bel(\{\omega_2\}) = 0.4$ *and* $Bel(\{\omega_3\}) = 0$. *As* $Bel(\{\omega_2\}) > pl(\omega_3)$, ω_2 *dominates* ω_3, *but* ω_1 *and* ω_2 *are not dominated. Consequently,* $\Omega^* = \{\omega_1, \omega_2\}$. □

2.2 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 can provide a 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 $u_{ik} = 1$ otherwise. If objects cannot be assigned to clusters with certainty, then it is natural to quantify cluster-membership uncertainty by mass functions $m_1, \ldots, m_n$, where each mass function m_i is defined on Ω and describes the uncertainty about the cluster of object i. The n-tuple $\mathcal{M} = (m_1, \ldots, m_n)$ is called a *credal partition* [7].

Example 3 *Consider, for instance, the "Butterfly" dataset shown in Figure 1. This dataset is adapted from the classical example by Windham [33], with an added outlier (point 12). Figure 2 shows the credal partition with $c = 2$ clusters produced by the EVCLUS algorithm (see Section 2.3 below). 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 $m_3(\{\omega_2\}) \approx 0.8$, which means that object o_3 most probably belongs to cluster ω_2. Similarly, $m_9(\{\omega_1\}) \approx 0.8$, indicating strong support in the assignment of object o_9 to cluster ω_1. 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 quite different for object o_{12}: here, we have $m_{12}(\emptyset) \approx 1$, indicating that this object does not seem to belong to any of the two clusters.* □

The notion of credal partition is very general, in the sense that it boils down to several alternative clustering structures when the mass functions composing the credal partition have some special forms:

- If all mass functions m_i are certain, then we have a *hard partition*, in which object o_i is unambiguously assigned to cluster ω_k if $m_i(\{\omega_k\}) = 1$.
- If the m_i are Bayesian, then the credal partition is equivalent to a *fuzzy partition* [2]; the degree of membership of object i to cluster k is then $u_{ik} = m_i(\{\omega_k\})$, with $\sum_{k=1}^{c} m_i(\{\omega_k\}) = 1$ for $i = 1, \ldots, n$.
- If the mass function m_i are consonant, then they are uniquely described by their contour functions $pl_i(\omega_k) = \sum_{A \subseteq \Omega, \omega_k \in A} m_i(A)$, which

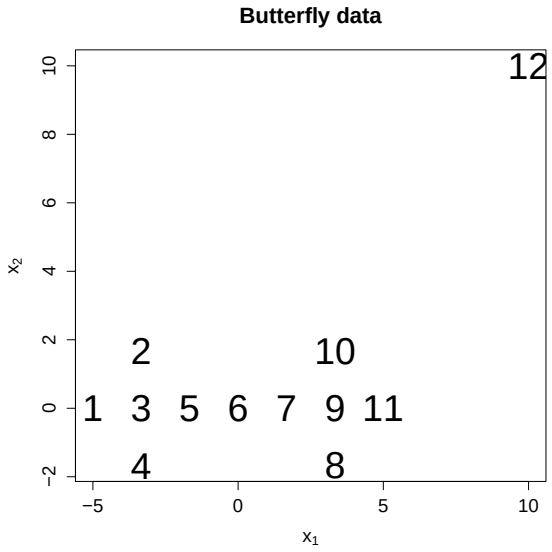


Figure 1: Butterfly dataset.

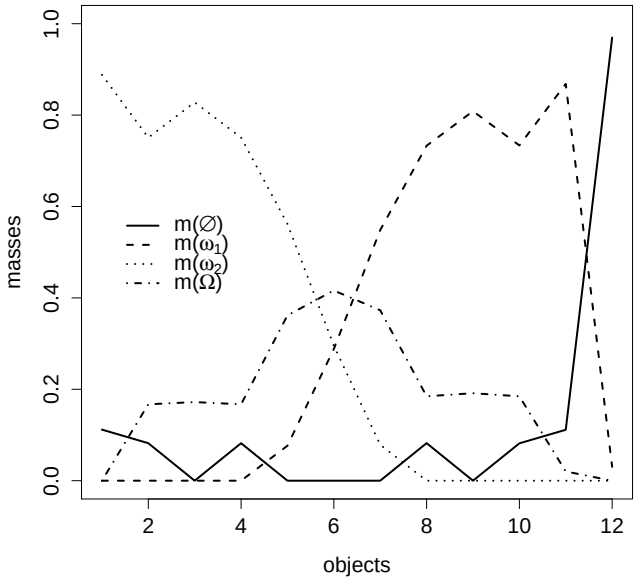


Figure 2: Credal partition of the Butterfly dataset.

are possibility distributions. We then have a *possibilistic partition* [16], with $u_{ik} = pl_i(\omega_k)$ for all i and k. We note that $\max_k pl_i(\omega_k) = 1 - m_i(\emptyset)$.

- Finally, if each m_i is logical, with $m_i(A_i) = 1$ for some $A_i \subseteq \Omega$, we can define *lower and upper approximations* of each cluster, as in rough clustering [20, 27]. The lower approximation of cluster ω_k is the set of objects that *surely* belong to ω_k,

$$\omega_k^L = \{o_i \in \mathcal{O} | A_i = \{\omega_k\}\} = \{o_i \in \mathcal{O} | Bel_i(\{\omega_k\}) = 1\}, \tag{9}$$

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

$$\omega_k^U = \{o_i \in \mathcal{O} | \omega_k \in A_i\} = \{o_i \in \mathcal{O} | Pl_i(\{\omega_k\}) = 1\}. \tag{10}$$

The membership values to the lower and upper approximations of cluster ω_k are then, respectively, $\underline{u}_{ik} = Bel_i(\{\omega_k\})$ and $\overline{u}_{ik} = Pl_i(\{\omega_k\})$.

Hard, fuzzy, possibilistic and rough partitions may also be computed from a credal partition as by-products, by applying some of the operations described in Section 2.1. Specifically, let $m_1, \ldots, m_n$ be a credal partition, and let $pl_1, \ldots, pl_n$ be the corresponding contour functions. Then $u_{ik} = pl_i(\omega_k)$ can be interpreted as the possibility that object i belongs to cluster ω_k, and the matrix $\boldsymbol{U} = (u_{ik})$ defines a possibilistic partition. By normalizing the u_{ik} as $u_{ik}^* = u_{ik} / \sum_\ell u_{i\ell}$, we get a fuzzy partition. Selecting, for each object i, the cluster ω_k with the highest plausibility gives us a hard partition. Finally, selecting for each m_i a set of clusters using the interval dominance decision rule (7) yields a rough partition.

2.3 EVCLUS algorithm

The first evidential clustering procedure, called EVCLUS, was introduced in [7]. It applies some ideas from Multidimensional Scaling (MDS) [3] to clustering. Let $\boldsymbol{D} = (d_{ij})$ be an $n \times n$ dissimilarity matrix, where d_{ij} denotes the dissimilarity between objects o_i and o_j. Dissimilarities may be distances computed from attribute data, or they may be provided directly, in which case they need not satisfy the axioms of a distance function.

To derive a credal partition $\mathcal{M} = (m_1, \ldots, m_n)$ from $\boldsymbol{D}$, 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,

as recalled in Section 2.1, 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 have mass functions m_i and m_j with low degree of conflict, whereas highly dissimilar objects are assigned highly conflicting mass functions.

This problem is similar to the one addressed by MDS, which aims to represent objects in some Euclidean space, in such a way that the distances in that space match the observed dissimilarities [3,4]. Here, we want to find a credal partition that minimizes the discrepancy between the pairwise degrees of conflict and the dissimilarities, up to some increasing transformation. In [7], we proposed to minimize the following stress function,

$$S(\mathcal{M}, a, b) = \sum_{i<j} \frac{(a\kappa_{ij} + b - d_{ij})^2}{d_{ij}} \ , \tag{11}$$

where a and b are two coefficients that make the solution invariant under any affine transformation of the dissimilarities. The division of each term in the sum by d_{ij} gives more weight to smaller dissimilarities[1].

New stress function. A simpler stress function, which will be used in the rest of this paper, is

$$J(\mathcal{M}) = \eta \sum_{i<j} (\kappa_{ij} - \delta_{ij})^2, \tag{12}$$

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 $[0, 1]$. How to choose function φ? If we could guess the value of some threshold d_0 such that any objects o_i and o_j probably belong to the same cluster whenever $d_{ij} \leq d_0$, and to different clusters otherwise, then we could define φ as

$$\varphi(d) = \begin{cases} 0 & \text{if } d \leq d_0, \\ 1 & \text{otherwise.} \end{cases} \tag{13}$$

However, such a step function is not differentiable, which would make the minimization of $J(\mathcal{M})$ difficult. We thus replace it by a soft threshold

[1] In [7], we actually proposed to minimize the sum of $S(\mathcal{M}, a, b)$ and an entropy term, aimed at penalizing more complex mass functions. In later investigations, we found that this entropy term adds unnecessary complexity to the method. It will not be used in the new version of EVCLUS described in this paper.

function, such as

$$\varphi(d) = 1 - \exp(-\gamma d^2), \tag{14}$$

where γ is a user-defined parameter. Parameter γ in (14) can be fixed as follows. For $\alpha \in (0,1)$, let d_0 be the distance such that

$$\varphi(d_0) = 1 - \exp(-\gamma d_0^2) = 1 - \alpha. \tag{15}$$

Solving Equation (15) for γ, we find $\gamma = -\log \alpha / d_0^2$. Parameter d_0 has a simple interpretation: two objects o_i and o_j such that $d_{ij} \geq d_0$ have a plausibility at least $1-\alpha$ of belonging to different clusters. In the simulations presented in this paper, we used $\alpha = 0.05$, leaving d_0 as the only parameter to be adjusted. Our results suggest that the results of EVCLUS are not very sensitive to the choice of d_0. Typically, d_0 can be set to some quantile of the dissimilarities d_{ij}. We suggest to start with the 0.9-quantile, but finding a suitable value of d_0 may sometimes require a trial and error process.

Remark: We can remark here an important difference between EVCLUS and MDS: in (11) and (12), the degrees of conflict κ_{ij} are not distances. They are not even dissimilarities, because the degree of conflict between a mass function and itself is nonnull, in general. However, we do have $\kappa_{ii} \approx 0$ whenever $m_i(\{\omega\}) \approx 1$ for some $\omega \in \Omega$, and $\kappa_{ij} \approx 1$ whenever $m_i(\{\omega\}) \approx 1$ and $m_j(\{\omega'\}) \approx 1$, for some $\omega \neq \omega'$. Consequently, criteria (11) and (12) are minimized when mass functions of similar (respectively, dissimilar) objects are focussed on the same cluster (respectively, on different clusters).

Example 4 *For the Butterfly data of Figure 1, let $\alpha = 0.05$ and $d_0 = 11$. Figure 3 shows the transformed dissimilarities $\varphi(d_{ij})$ as a function of the Euclidean distances d_{ij}. Figure 4 is a plot, called a "Shepard diagram" in MDS [3], showing the degrees of conflict κ_{ij} as a function of the transformed dissimilarities δ_{ij}.* □

In [7], we proposed to minimize stress function (11) or (12) using an iterative gradient-based optimization procedure. The constraints $\sum_{k=1}^{f} m_i(F_k) = 1$ and $m_i(F_k) \geq 0$, where $F_1, \ldots, F_f$ are the focal sets of the mass functions m_i, are implicitly taken into account by the following reparametrization,

$$m_i(F_k) = \frac{\exp(\alpha_{ik})}{\sum_{\ell=1}^{f} \exp(\alpha_{i\ell})}, \tag{16}$$

where the α_{ik} for $i = 1, \ldots, n$ and $k = 1, \ldots, f$ are nf real parameters representing the credal partition.

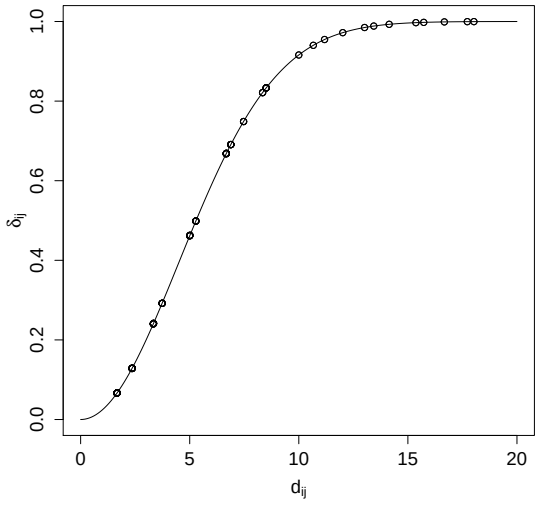


Figure 3: Transformed dissimilarities $\delta_{ij} = \varphi(d_{ij})$ vs. Euclidean distances d_{ij} for the Butterfly dataset.

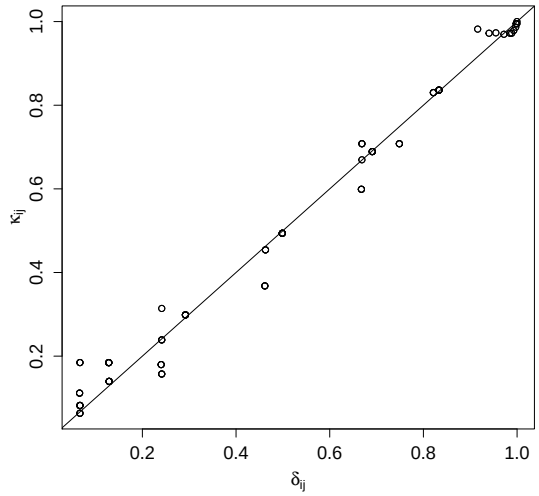


Figure 4: Shepard diagram for the Butterfly dataset: degrees of conflict κ_{ij} (y-axis) vs transformed dissimilarities δ_{ij} (x-axis).

2.4 Discussion

In [7], the EVCLUS algorithm was shown to perform very well in discovering meaningful clusters in several non-Euclidean datasets, a notoriously difficult problem (see, e.g., [9, 15]). In particular, its performances compared favorably with those of state-of-the-art techniques such as the Non Euclidean Relational Fuzzy c-Means (NRFCM) algorithm [9]. In addition, the very general concept of credal partition results in greater expressive power and in improved robustness with respect to atypical data.

Yet, the original EVCLUS also has some limitations. First, as most clustering algorithms for proximity data, it requires storing the whole dissimilarity matrix, which has space complexity $O(n^2)$, where n is the number of objects. Consequently, the algorithm is not suitable for datasets containing more than a few thousand objects. Secondly, each gradient calculation needed in the non linear optimization procedure of EVCLUS requires $O(f^3n^2)$ operations, where f is the number of focal sets of the mass functions in the credal partition. Except for very small numbers of clusters, we thus need to restrict the form of the mass functions, in such a way that the number of focal sets remains proportional to c. In [7], we proposed to limit the focal sets to the singletons $\{\omega_k\}$, the empty set $\emptyset$, and the whole set of clusters Ω. Whereas this restriction makes EVCLUS potentially applicable to datasets with a large number of clusters, it also severely limits the expressive power of the generated credal partitions.

After EVCLUS, other evidential clustering algorithms have been proposed. The Evidential c-Means (ECM) algorithm, introduced in [25], is an alternating optimization algorithm akin to the fuzzy c-means (FCM) algorithm [2], which alternatively searches for the best credal partition given a set of prototypes, and then for best prototypes given the credal partition. The main difference with FCM is that prototypes are defined not only for clusters, but also for sets of clusters (or "meta-clusters"). In [23], a variant of the ECM algorithm (called CCM) was proposed, based on an alternative definition of the distance between a vector and the prototype of a meta-cluster. This modification sometimes produces more sensible results in situations where the prototype of a meta-cluster is close to that of singleton cluster. The ECM and CCM algorithms work only with attribute data, but a version of ECM for dissimilarity data, the Relational Evidential c-Means (RECM) was proposed in [26]. The RECM algorithm was shown to yields results comparable to those of EVCLUS for some datasets, while being significantly faster. One iteration of RECM involves $O(nfc^2 + n^2c)$ operations: it thus takes time proportional to the number f of focal sets.

This lower complexity makes it possible to generate credal partitions with general mass functions for moderate values of c. However, RECM assumes the dissimilarities to be Euclidean distances. If this assumption is not verified, it may produce poor results, or even fail to converge. Also, RECM needs to store the whole dissimilarity matrix in memory, and is thus not suitable for very large datasets.

In [34], Zhou et al. introduce another variant of ECM, called the Median Evidential c-means (MECM), which is an evidential counterpart to the median c-means and median fuzzy c-means algorithms. The MECM can be used with dissimilarity data, and it does not require the dissimilarities between objects to verify the axioms of distances. Yet, it still requires to store the whole dissimilarity matrix. Recently, we introduced another evidential clustering procedure based on the evidential k-nearest neighbor rule, called Ek-NNclus [6]. The Ek-NNclus uses only the k nearest neighbors of each object: consequently, it has lower storage requirements than EVCLUS, RECM or MECM, which makes it suitable for clustering very large datasets. However, Ek-NNclus generates only very simple credal partitions, in which masses are assigned only to singletons $\{\omega_k\}$ and to the set Ω of clusters. Its outputs thus do not have as much expressive power as those of EVCLUS and RECM.

From this general overview of evidential clustering algorithms[2], we can conclude that EVCLUS has some distinctive advantages over competing algorithms in terms of applicability to non-metric dissimilarities and expressive power. However, it suffers from a relatively high complexity, which limits its application to datasets of a few thousand objects with a small number of clusters. In the rest of this paper, we will see how EVCLUS can be modified to overcome these limitations.

3 Fast optimization

In this section, we show experimentally that the Iterative Row-wise Quadratic Programming (IRQP) algorithm introduced in [31] can, by exploiting the particular form of stress function (12), drastically speed up the EVCLUS procedure. The method will be described in Section 3.1, and experimental results will be presented in Section 3.2.

[2]ECM, RECM, Ek-NNclus, and k-EVCLUS introduced in this paper, have been implemented in the R package `evclust` [5] available on the first author's web page at `https://www.hds.utc.fr/~tdenoeux`.

3.1 Algorithm

To simplify the presentation of the proposed optimization algorithm, let us rewrite (12) using matrix notations. Let us assume that the f focal sets $F_1, \ldots, F_f$ of mass functions $m_1, \ldots, m_n$ have been ordered in some way. We can then represent each mass function m_i by a vector $\boldsymbol{m}_i = (m_1(F_1), \ldots, m_i(F_f))^T$ of length f. The credal partition $\mathcal{M} = (m_1, \ldots, m_n)$ can then be represented by a matrix $\boldsymbol{M} = (\boldsymbol{m}_1^T, \ldots, \boldsymbol{m}_n^T)^T$ of size $n \times f$.

The degree of conflict (3) between two mass functions m_i and m_j can be written as

$$\kappa_{ij} = \boldsymbol{m}_i^T \boldsymbol{C} \boldsymbol{m}_j, \tag{17}$$

where $\boldsymbol{C}$ is the square matrix of size f, with general term

$$C_{k\ell} = \begin{cases} 1 & \text{if } F_k \cap F_\ell = \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

With these notations, the stress function (12) can be written as

$$J(\boldsymbol{M}) = \eta \sum_{i<j} (\boldsymbol{m}_i^T \boldsymbol{C} \boldsymbol{m}_j - \delta_{ij})^2. \tag{19}$$

The idea behind the IRQP algorithm is to minimize $J(\boldsymbol{M})$ with respect to each row of $\boldsymbol{M}$ at a time, keeping the other rows constant [31]. Minimizing $J(\boldsymbol{M})$ with respect to $\boldsymbol{m}_i$ is equivalent to minimizing

$$g(\boldsymbol{m}_i) = \|\boldsymbol{M}_{-i} \boldsymbol{C} \boldsymbol{m}_i - \boldsymbol{\delta}_i\|^2, \tag{20}$$

where $\boldsymbol{M}_{-i}$ is the matrix obtained from $\boldsymbol{M}$ by deleting row i, and $\boldsymbol{\delta}_i$ is the vector of transformed dissimilarities δ_{ij} between object o_i and all other objects o_j, $j \neq i$. Minimizing $g(\boldsymbol{m}_i)$ under the contraints $\boldsymbol{m}_i^T \boldsymbol{1} = 1$ and $\boldsymbol{m}_i \geq \boldsymbol{0}$ is a linearly constrained positive least-squares problem, which can be solved using efficient algorithms (see, e.g., [14]).

By iteratively updating each row of $\boldsymbol{M}$ as described above, as long as the overall function value decreases, the algorithm converges to a stable function value, which is at least a local minimum. To decide when to stop the algorithm, we compute a running mean of the relative error as follows,

$$e_0 = 1, \tag{21a}$$

$$e_t = \rho\, e_{t-1} + (1-\rho) \frac{|J_t - J_{t-1}|}{J_{t-1}}, \quad t = 1, 2, \ldots, \tag{21b}$$

where t is the iteration counter, J_t is the stress value at iteration t, and $\rho = 0.5$. The algorithm is then stopped when $e_t < \epsilon$, for some threshold ϵ. The whole procedure is summarized in Algorithm 1.

The complexity of Algorithm 1 depends on the complexity of the Quadratic Programming (QP) problem (20) solved at each iteration. Each instance of this problem has f variables and $f + 1$ constraints: it is thus much smaller than the initial nonlinear optimization problem, which has nf variables and $n(f + 1)$ constraints. Furthermore, the quadratic function (20) being minimized in convex. It is known [32] that convex QP problems can be solved in polynomial time. Worst case theoretical bounds exist [32], but they are of little use in practice, because the actual running time depends critically on the particular instance of the problem and the algorithm used. As we will see in Section 3.2 below, we found experimentally the IRQP algorithm to be much faster than the gradient algorithm for minimizing the stress function (12).

Algorithm 1 EVCLUS-IRQP algorithm.

Input: Dissimilarities $D = (d_{ij})$, d_0, α, ϵ

```
γ ← −log α/d_0^2
Compute δ_ij ← 1 − exp(−γ d_ij^2) for 1 ≤ i < j ≤ n
Compute η = ( Σ_{i<j} δ_ij^2 )^{−1}
Compute matrix C using (18)
Initialize credal partition matrix M randomly
t ← 0, e_0 ← 1
Compute J_0 using (19)
while e_t ≥ ε do
    t ← t + 1
    J_t ← 0
    for i = 1 to n do
        Delete row i from current matrix M to get M_{−i}
        Find m_i^{(t)} by minimizing (20) subject to m_i^T 1 = 1 and m_i ≥ 0
        J_t ← J_t + η g(m_i^{(t)})
    end for
    e_t ← 0.5 e_{t−1} + 0.5 |J_t − J_{t−1}| / J_{t−1}
end while
```

Output: Credal partition $\boldsymbol{M}$

3.2 Simulation results

In this section, we first compare the Gradient-based and IRQP optimization algorithms on real data: the protein dataset, and then on simulated data, for which we can vary the number of objects. For all the experiments reported in this section, we used the version of EVCLUS with the empty set $\emptyset$, the singletons $\{\omega_k\}$, and Ω as focal sets.

Experiment 1: Protein dataset. The Protein dataset [7, 8, 12] consists of a dissimilarity matrix derived from the structural comparison of 213 protein sequences. Each of these proteins is known to belong to one of four classes of globins: hemoglobin-α (HA), hemoglobin-β (HB), myoglobin (M) and heterogeneous globins (G). Figure 5 displays a two-dimensional MDS configuration of the data with the true partition, as well as the clustering result obtained by EVCLUS, with $c = 4$ and $d_0 = \max_{i,j} d_{ij}$. We show both the hard partition obtained by assigning each object to the cluster with the highest plausibility, as well as the lower and upper approximations of each cluster, obtained using the interval-dominance rule. We can see that the clustering structure of the data is well recovered, with only two misclassified objects.

We ran the Gradient and IRQP algorithms on the Protein dataset with $c = 4$ and $\epsilon = 10^{-5}$. Both algorithms were run 20 times from random initial values. In each run, both algorithms were started from the same initial conditions. Figure 6 shows the evolution of stress as a function of time for the Gradient (left) and IRQP (right) algorithms. We can see that, on this data, the IRQP algorithm converges more than 10 times faster than the Gradient algorithm. We also see that the stress values at convergence for IRQP have lower variability and are consistently smaller than those obtained by the Gradient algorithm. This is also illustrated by Figure 7, which shows boxplots of the stress values at convergence and computing times, for both algorithms.

Experiment 2: Simulated dataset To study the influence of the number of objects on the computing time of both algorithms, we generated artificial datasets with four clusters of $n/4$ two-dimensional vectors, generated from a multivariate t distribution with five degrees of freedom and centered, respectively, on $[0, 0]$, $[0, 5]$, $[5, 0]$ and $[5, 5]$. The dissimilarities were computed as the Euclidean distances between the data points. A typical dataset with $n = 200$ is shown in Figure 8, together with the result of EVCLUS with $c = 4$ and d_0 equal to the 0.8-quantile of the Euclidean dis-

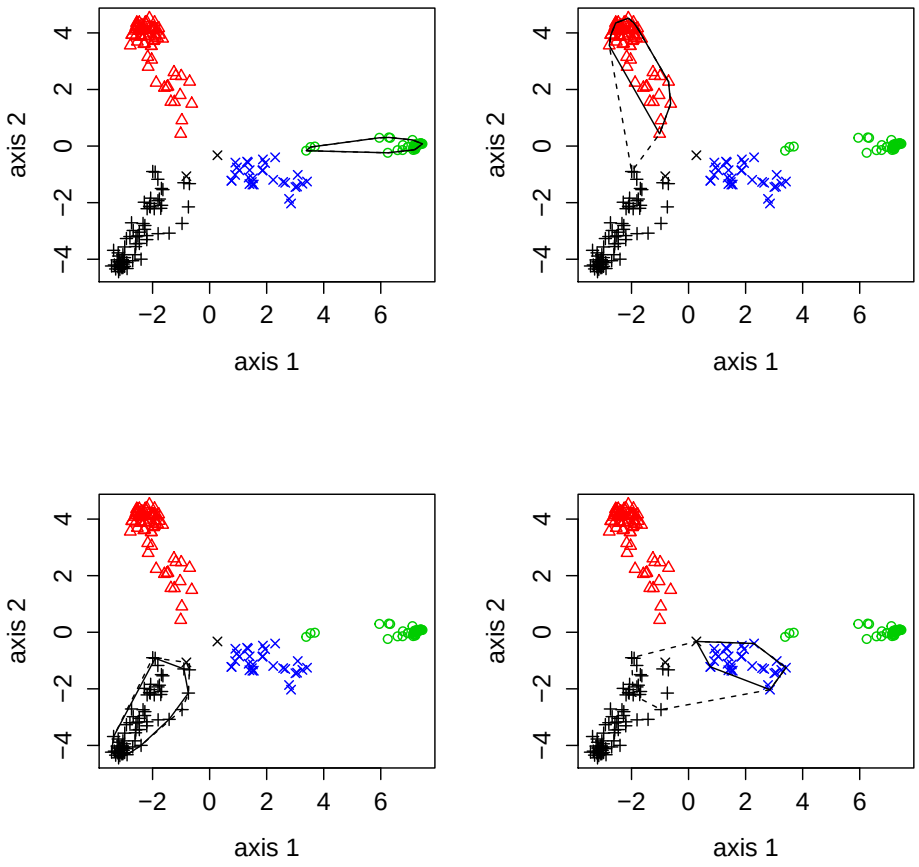
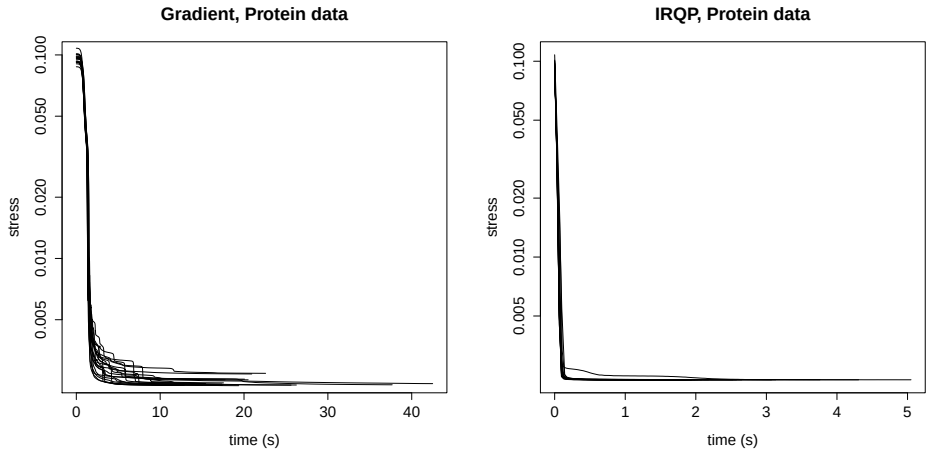
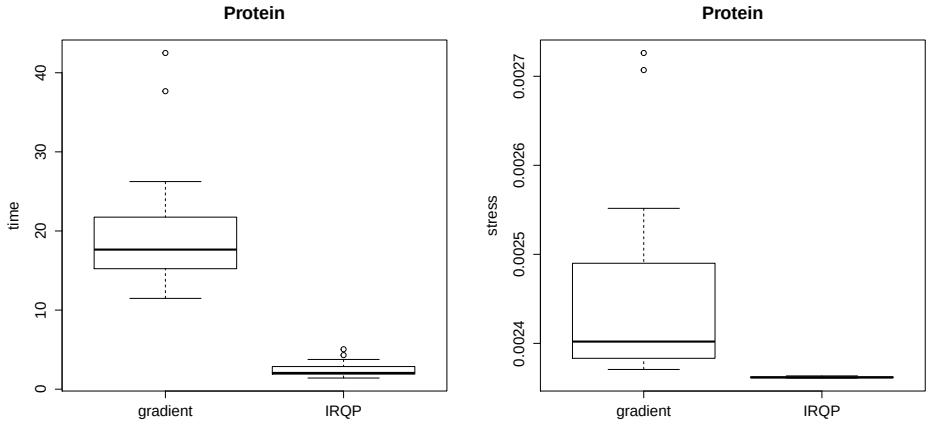


Figure 5: Lower and upper approximations of four clusters for the Protein dataset. The true classes are HA (black), HB (red), M (green) and G (blue). The clusters found by EVCLUS are plotted with different symbols. The convex hulls of the cluster lower and upper approximations are displayed using solid and interrupted lines, respectively.



(a) (b)

Figure 6: Stress vs. time (in seconds) for 20 runs of the Gradient (a) and IRQP (b) algorithms on the Protein data. Note the different scales on the x-axes.



(a) (b)

Figure 7: Boxplots of computing time (a) and stress value at convergence (b) for 20 runs of the Gradient and IRQP algorithms on the Protein data.

tances. In Figure 8, the outliers, shown as circles, are defined as points such that $m_i(\emptyset) > m_i(A)$, for all non empty subset A of Ω. The lower and upper approximations of each cluster are computed using the interval dominance rule, but the outliers are excluded from the lower approximations.

Figure 9 shows stress as a function of time for both optimization algorithms applied to a single dataset of $n = 200$ objects, with 20 different random initial conditions. Figure 10 shows boxplots of computing times and stress for the 20 runs of both algorithms. From Figure 9, we can see that the Gradient algorithm was trapped three times in a local minimum, while the IRQP algorithm was trapped only once. Overall, the results for this dataset are similar to the previous ones with the Protein data: the IRQP algorithm converges more than 10 times faster than the Gradient algorithm, and it converges to lower values of the stress function.

To study the influence of the number n of objects on computing time, we varied n from 100 to 600. For each value of n, we generated 20 different datasets, which we clustered using EVCLUS, with the two optimization algorithms. As a comparison, we also applied the RECM algorithm [26] to the same data. For RECM, the parameters were set to $c = 4$, $\alpha = 1$, $\beta = 1.5$, $\delta = d_0$ and $\epsilon = 10^{-5}$. As for EVCLUS, the focal sets were restricted to the empty set $\emptyset$, singletons $\{\omega_k\}$, and Ω. The results are shown in Figure 11. These results show that the computing time of the Gradient algorithm increases much faster as a function of n, than those of the IRQP and RECM algorithms (Figure 11(a)).

When comparing IRQP with RECM (Figure 11(b)), we can see that the latter is still faster, and its computing time increases more slowly with n than that of EVCLUS with IRQP. The difference, however, is much smaller than that reported in [26], where the Gradient algorithm was used. For this dataset, where dissimilarities are metric, EVCLUS and RECM yield similar results. It must be recalled, however, that RECM may yield poor results or even fail to converge when applied to non metric dissimilarities, in contrast with EVCLUS (see Section 4.2 below).

We also applied the MECM algorithm [34], using R code provided by the authors. We found this implementation of MECM much slower than both EVCLUS and RECM. For the simulated data studied in this section, the average computing time was 46.0 s for $n = 100$ and 141.0 s for $n = 200$. It is clear that this algorithm (at least, in this implementation) is not suitable for the clustering of large data sets. For this reason, we did not consider it for further analysis. We will come back in Section 4 to the comparison between evidential clustering algorithms.

In this section, we have shown that the IRQP optimization algorithms

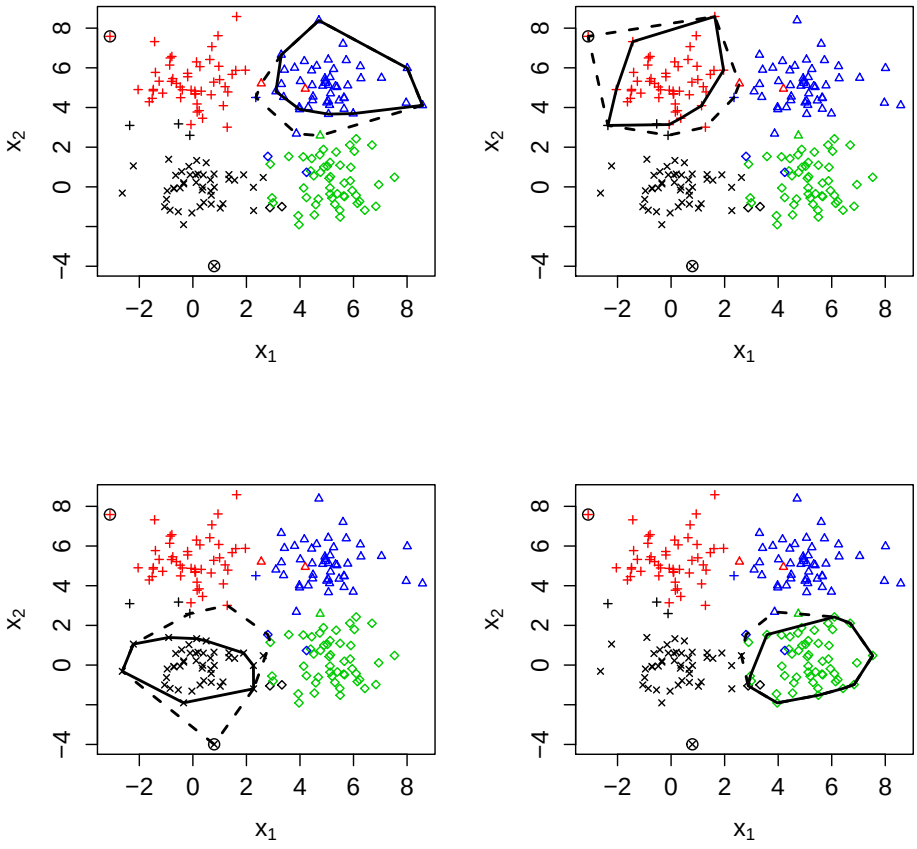
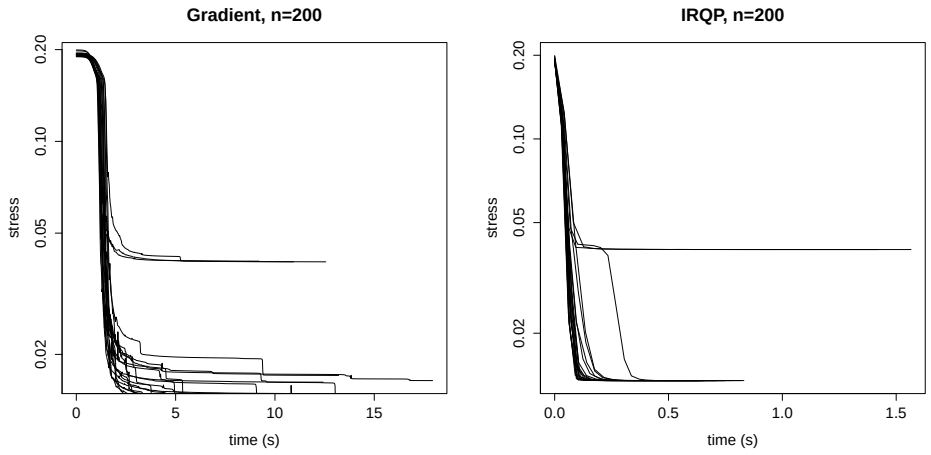
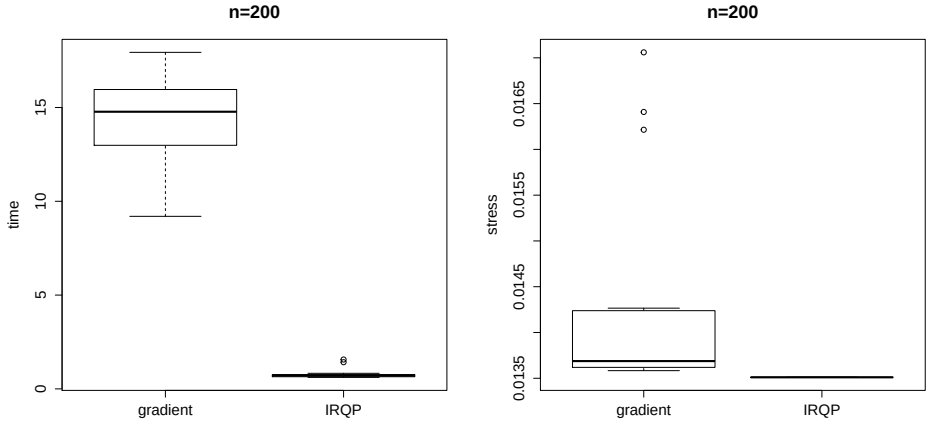


Figure 8: Lower and upper approximations of the four clusters for one generated dataset with $n = 200$. The true classes are displayed with different colors. The clusters found by EVCLUS are plotted with different symbols. The convex hulls of the cluster lower and upper approximations are displayed using solid and interrupted lines, respectively. The two outliers are indicated by circles.



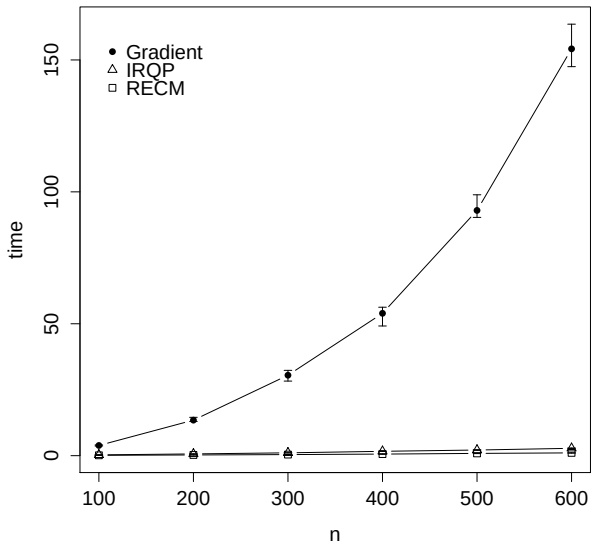
(a) (b)

Figure 9: Stress vs. time (in seconds) for 20 runs of the Gradient (a) and IRQP (b) algorithms on the synthetic data, with $n = 200$. Note the different scales on the x-axes.

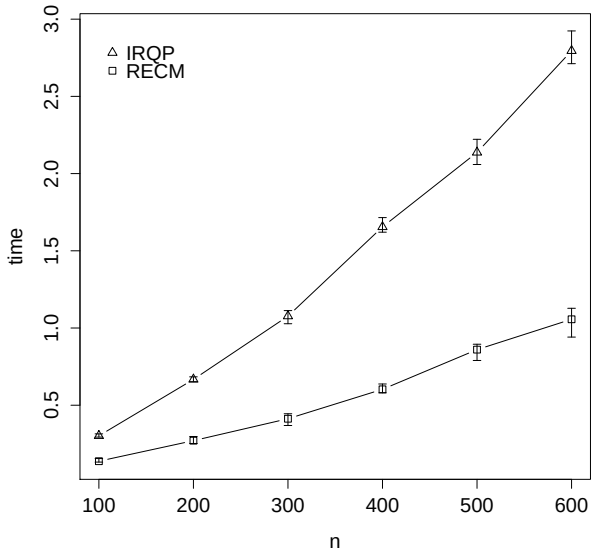


(a) (b)

Figure 10: Boxplots of computing time (a) and stress value at convergence (b) for 20 runs of the Gradient and IRQP algorithms on the synthetic data, with $n = 200$.



(a)



(b)

Figure 11: Computing time (in seconds) as a function of the number n of objects for EVCLUS with the Gradient and IRQP algorithms and for RECM (a), and zoom on the curves corresponding to IRQP and RECM (b).

allows us to gain several orders of magnitude in computing time, as compared to the Gradient algorithm, when using the EVCLUS algorithms with data sets of moderate size (from several hundred to a few thousand objects). However, because it uses the full dissimilarity matrix, the space and time complexity of EVCLUS remains proportional to n^2, which makes it inapplicable to very large datasets. In Section 4, we will see that this limitation can be overcome, making EVCLUS competitive with other evidential clustering algorithms for large datasets.

4 Handling large datasets

As mentioned above, the $O(n^2)$ complexity of EVCLUS, where n is the number of objects, makes it inapplicable to large dissimilarity data. The fundamental reason for this high complexity is the fact that the calculation of stress criterion (12) requires the full dissimilarity matrix. However, as is well-known, there is usually some redundancy in a dissimilarity matrix, even if the dissimilarity measure is not a distance. 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,

$$J_k(\mathcal{M}) = \eta \sum_{i=1}^{n} \sum_{r=1}^{k} (\kappa_{i,j_r(i)} - \delta_{i,j_r(i)})^2, \tag{22}$$

where, as before, η is a normalizing constant,

$$\eta = \left(\sum_{i=1}^{n} \sum_{r=1}^{k} \delta^2_{i,j_r(i)} \right)^{-1}. \tag{23}$$

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. In the experiments below, we show that this version of EVCLUS (hereafter referred to as k-EVCLUS) is more scalable than the original version, and applicable to large dissimilarity datasets. The results of these experiments will also provide guidelines for the choice of k.

4.1 Results with simulated data

First, we simulated data from the same distribution as that used in Experiment 2 of Section 3.2, with different numbers n of objects, and we used the Euclidean distances as dissimilarities. Algorithm k-EVCLUS was run with d_0 equal to the 0.9-quantile of distances, $c = 4$, and $\epsilon = 10^{-5}$.

Figures 12-14 shows the ARI, computing time[3] and average nonspecificity as functions of k for a simulated dataset with $n = 2000$. The values of k were chosen as 10, 20, 50, 100, 200, 500 and 1999. When $k = 1999 = n-1$, the whole distance matrix is used, and k-EVCLUS boils down to EVCLUS. The average nonspecificity, defined as

$$N^* = \frac{1}{n \log_2(c)} \sum_{i=1}^{n} \left[\sum_{A \in 2^\Omega \setminus \emptyset} m_i(A) \log_2 |A| + m_i(\emptyset) \log_2(c) \right], \qquad (24)$$

was shown in [25] to a good validity index for a credal partition. It is comprised between 0 and 1. Smaller values mean that masses are assigned to non empty focal sets with small cardinality, which is evidence for the adequacy of the credal partition to the data. For each k, the algorithm was run 10 times with different random initial conditions. The median as well as the lower and upper quartiles are reported.

As we can see, k-EVCLUS performs as well as EVCLUS ($k = 1999$) in terms of ARI and nonspecificity (Figures 12 and 14), as long as $k \geq 100$, with a significant gain in training time (Figure 13). We observe that the computing time is higher for $k = 10$ than it is for $k = 20$, which is due to the fact that the algorithm took more time to converge for $k = 10$. Figure 15 displays lower and upper approximations of the four clusters obtained by k-EVCLUS, with $k = 50$ (Figure 15(a)) and $k = 1000$ (Figure 15(b)). We can see that the credal partitions obtained for these two values of k are qualitatively very similar, which confirms that very little information is lost by taking k as small as 50, in which case only 5% of the distance matrix is actually used.

We can wonder if the necessary value of k increases in proportion of n, i.e., if n is multiplied by some positive number, do we need to also multiply k by the same amount? To answer this question, we repeated the above experiments with $n = 10000$. The results are shown in Figures 16-18. We can see that three curves in this figure are very similar to those of Figures

[3]All simulations reported in this paper were performed on an Apple MacBook Pro computer with a 2.5 GHz Intel Core i7 processor.

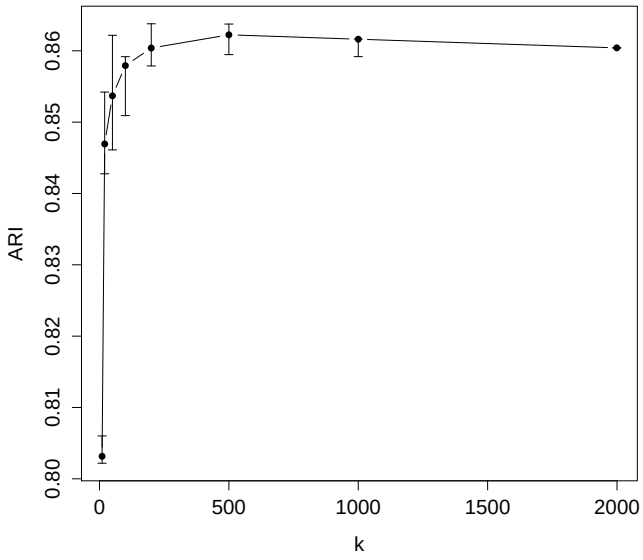


Figure 12: Adjusted Rand Index between the true partition and the maximum plausibility partition found by k-EVCLUS as a function of k for the simulated data with $n = 2000$. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.

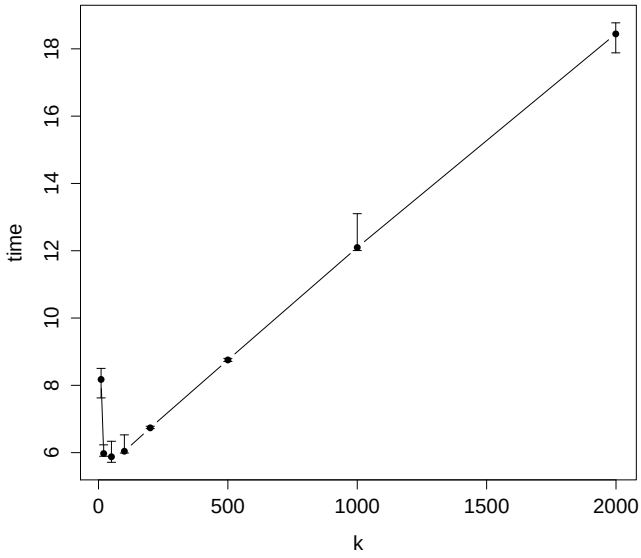


Figure 13: Computing time of k-EVCLUS as a function of k for the simulated data with $n = 2000$. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.

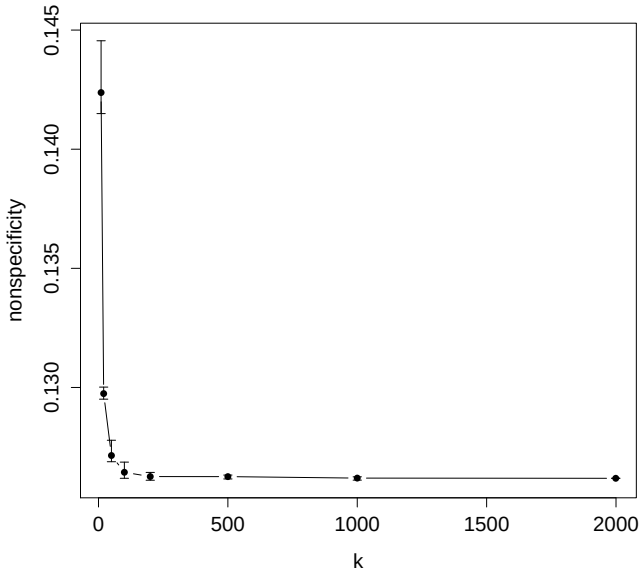
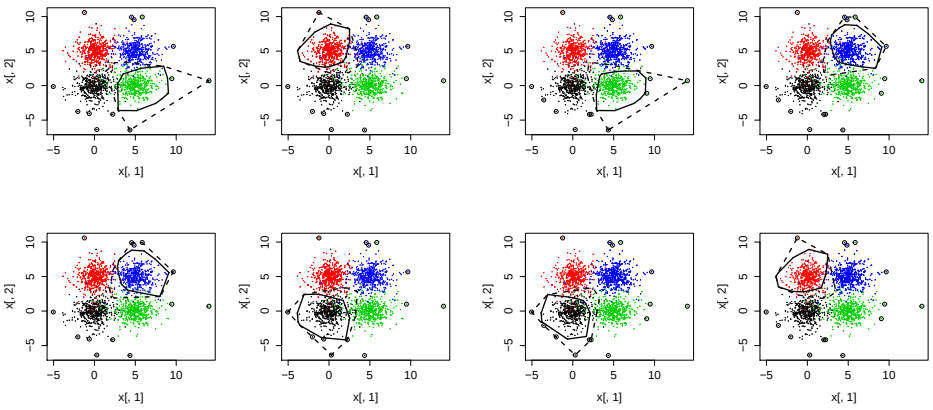


Figure 14: Nonspecificity of the maximum plausibility partition found by k-EVCLUS as a function of k for the simulated data with $n = 2000$. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.



(a) (b)

Figure 15: Lower and upper approximations of the four clusters for the simulated dataset of size $n = 2000$, obtained by k-EVCLUS with $k = 50$ (a) and $k = 1000$ (b). The outliers are indicated by circles.

12-14, and k-EVCLUS again performs similarly to EVCLUS for $k \geq 100$. We note that, for $k = 100$, k-EVCLUS uses only 2% of the distances.

It is also interesting to find out how k-EVCLUS compares with RECM and EK-NNclus in terms of quality of the results and computing time:

- The RECM algorithm was run 10 times with random initialization and the following default parameter values: $\alpha = 1$, $\beta = 1.5$ and $\delta^2 = 0.95$ quantile of dissimilarities. For $n = 2000$, the mean computing time of RECM was 12.99 s, and the mean ARI was 0.86. For $n = 10000$, it was not possible to run RECM because of memory limitations.

- EK-NNclus was also run 10 times with random initialization (with 1000 initial clusters) and $q = 0.95$. For $n = 2000$, we set $K = 100$; the mean computing time was 3.46 and the mean ARI was 0.74 (maximum: 0.83). For $n = 10000$ with $K = 300$, the mean computing time was 72.06 s, and the mean ARI was 0.73 (maximum: 0.86).

Comparing these results with those of Figures 12-13 and 16-17, we can see that RECM performs comparably with k-EVCLUS for $n = 2000$, but it does not scale to significantly larger datasets. As far as EK-NNclus is concerned, it is slightly faster than k-EVCLUS for $n = 2000$, but it becomes slower for $n = 10000$; it is also less robust than k-EVCLUS to initial conditions.

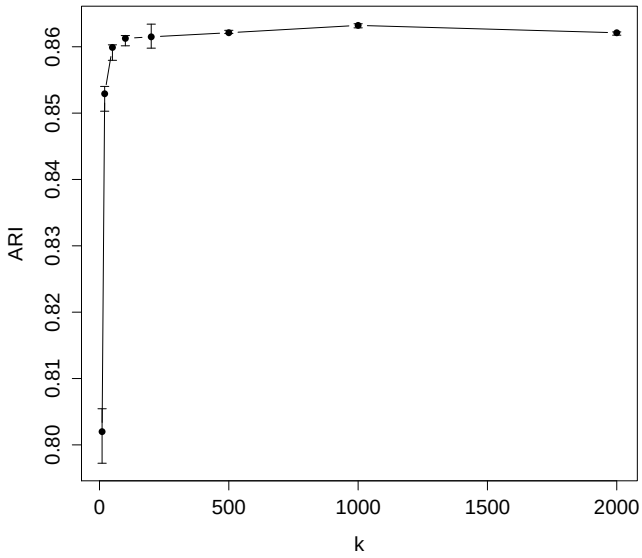


Figure 16: Adjusted Rand Index between the true partition and the maximum plausibility partition found by k-EVCLUS as a function of k for the simulated data with $n = 10000$. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.

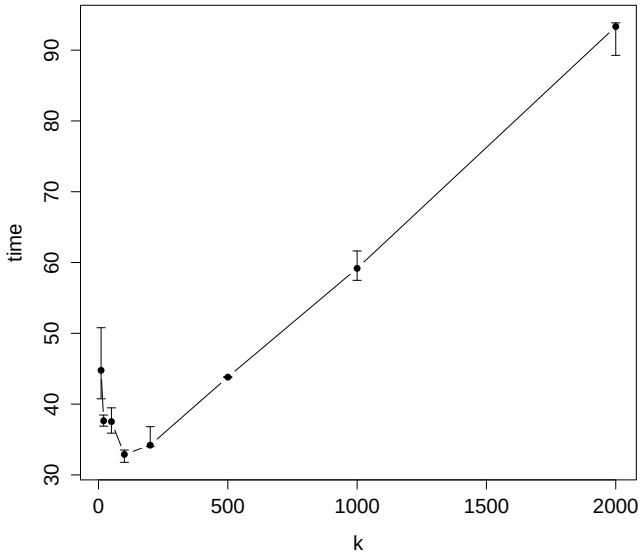


Figure 17: Computing time of k-EVCLUS as a function of k for the simulated data with $n = 10000$. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.

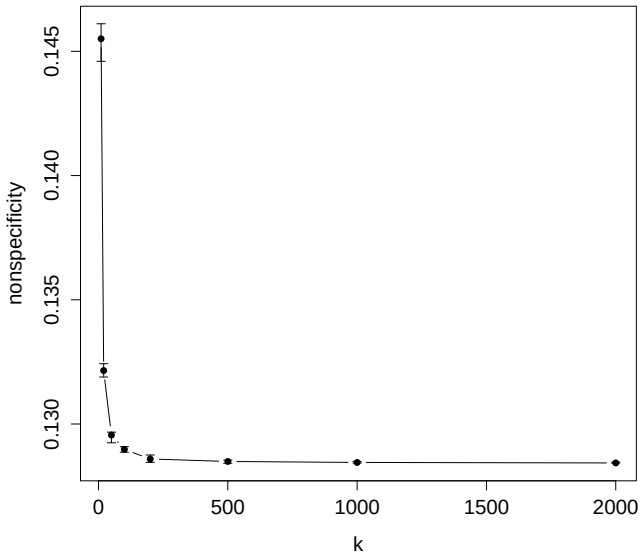


Figure 18: Nonspecificity of the maximum plausibility partition found by k-EVCLUS as a function of k, as a function of k, for the simulated data with $n = 10000$. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.

To compare these three algorithms in a more systematic way, we let n vary in from 1000 to 5000 (by 1000 increments), and we generated 10 datasets of each size, from the same distribution. We then recorded the computing times and ARI values for k-EVCLUS (with $k = 100$ and d_0 equal to the 0.9-quantile of the distances), RECM (with the same parameters as above), and EK-NNclus with $K = 3\sqrt{n}$ and $q = 0.95$. The results are reported in Figure 19. From Figure 19(a), we can see that k-EVCLUS and EK-NNclus are comparable in terms of computing time for different values of n, whereas the time complexity of RECM seems to be considerably higher. On the other hand, k-EVCLUS and RECM yield comparable results in terms of ARI (see Figure 19(b)), whereas the partitions obtained by EK-NNclus have higher variability. It must be noticed that the number c of clusters is specified for k-EVCLUS and RECM, but it is not for EK-NNclus. Overall, k-EVCLUS seems to provide the best results (for correctly specified c) in the least amount of time.

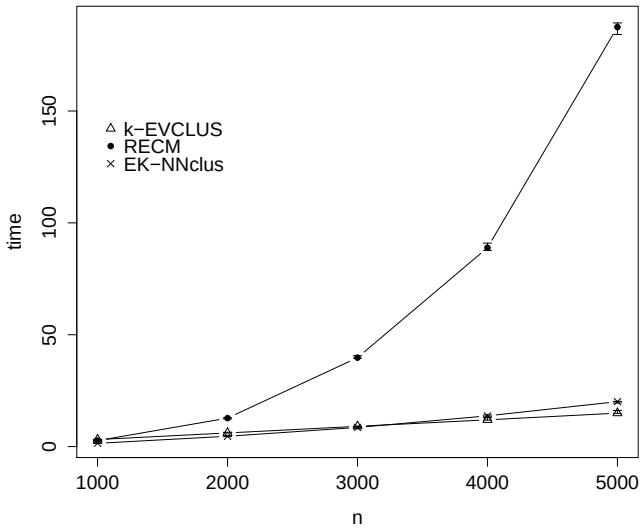
4.2 Results with real data

In this section, we consider two real dissimilarity datasets, both available from `http://prtools.org/disdatasets/index.html`. In these two datasets, the dissimilarities are non metric, i.e., they are not Euclidean distances.

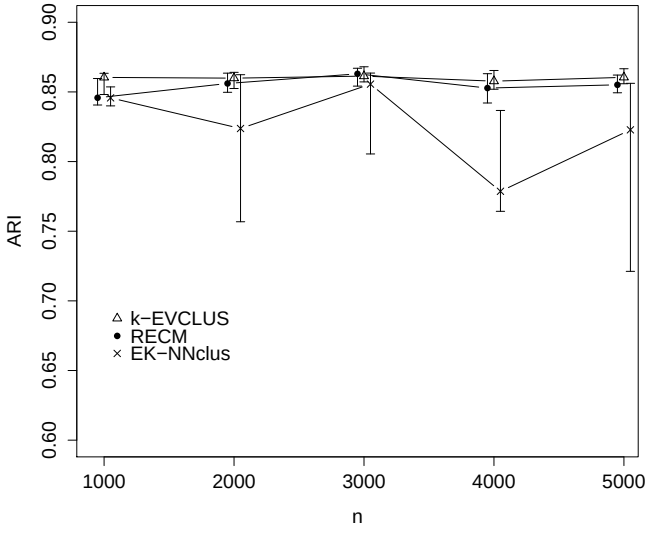
Zongker Digit dissimilarity data. This dataset contains similarities between 2000 handwritten digits in 10 classes, based on deformable template matching. The dissimilarity measure is the result of an iterative optimization of the non-linear deformation of the grid [13]. As the dissimilarity matrix was initially non symmetric, we symetrized it by the transformation $d_{ij} \leftarrow (d_{ij} + d_{ji})/2$.

The k-EVCLUS algorithm was run with $c = 10$ and the following values of k: 30, 50,100, 200, 300, 400, 500, 1000 and 1999. Parameter d_0 was fixed to the 0.3-quantile of the dissimilarities. For each value of k, k-EVCLUS was run 10 times with random initializations. The results are shown in Figures 20-22. We can see that optimal results (with an ARI roughly equal to 0.8) are reached for $k \geq 300$.

In contrast, RECM does not perform well on this data set, probably because of the non-metric nature of the dissimilarities. With the default values $\alpha = 1$, $\beta = 1.5$, and $\delta^2 =$ equal to the 0.95-quantile of dissimilarities, we obtained an average ARI over 10 trials equal to 0.25 (maximum: 0.36). The mean running time was 13.1s, and the mean nonspecificity was 0.01.



(a)



(b)

Figure 19: Computing time (a) and ARI (b) for k-EVCLUS, RECM and EKNNclus for simulated datasets with different values of n.

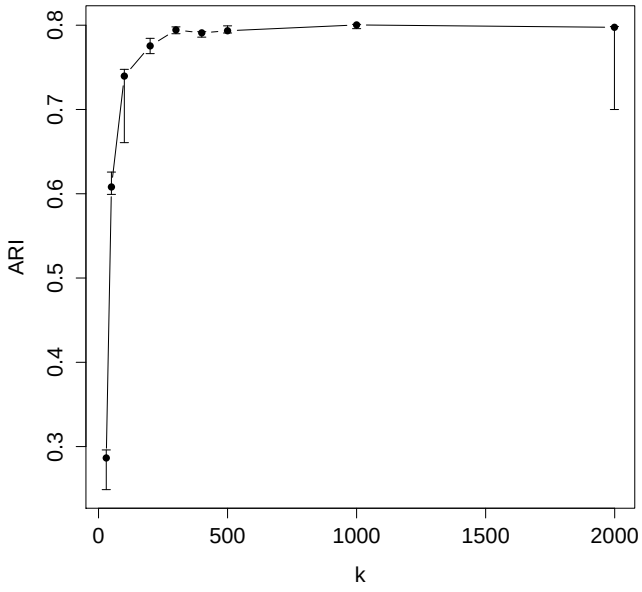


Figure 20: Adjusted Rand Index between the true partition and the maximum plausibility partition found by k-EVCLUS as a function of k for the Zongker digits data. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.

We also tested EK-NNclus on this dataset. For $K = 50$ and $q = 0.3$, EkNNclus found 9 clusters in 9.53 s on average (over 10 trials). The mean ARI was equal 0.55 (with a maximum of 0.58), which is significantly less that the values obtained by k-EVCLUS with the correct number of clusters.

Delft Gestures data This dataset consists of the dissimilarities computed from a set of gestures in a sign-language study [19]. They were measured by two video cameras observing the positions the two hands in 75 repetitions of creating 20 different signs. There are thus 1500 objects grouped in 20 clusters. The dissimilarities result from a dynamic time warping procedure.

Figures 23-25 shows the results obtained by k-EVCLUS with $c = 20$, d_0 fixed to the 0.2-quantile of the dissimilarities, and the following values of k: 30, 50, 100, 200, 300, 400, 500, 1000 and 1499. The three curves showing

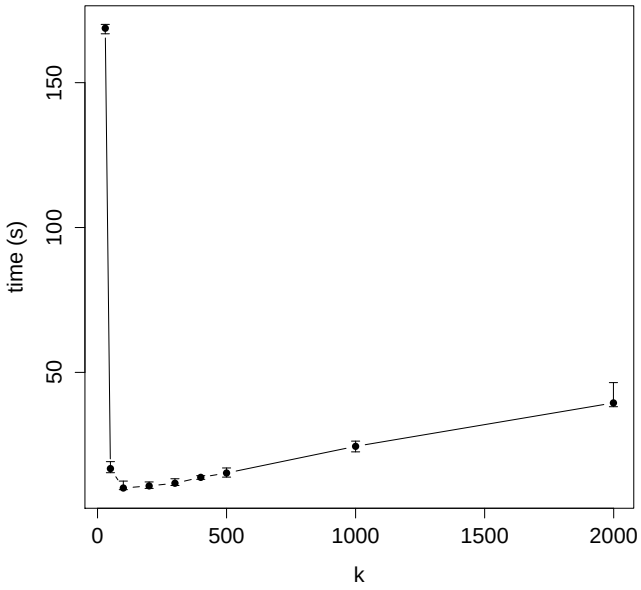


Figure 21: Computing time of k-EVCLUS as a function of k for the Zongker digits data. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.

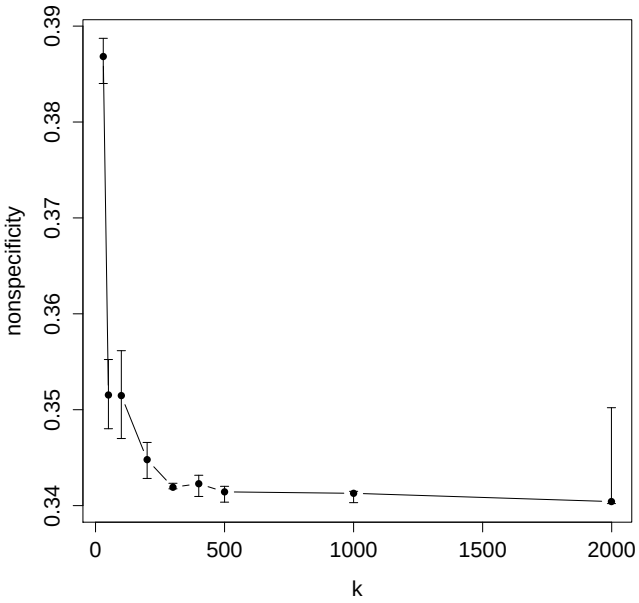


Figure 22: Nonspecificity of the maximum plausibility partition found by k-EVCLUS as a function of k for the Zongker digits data. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.

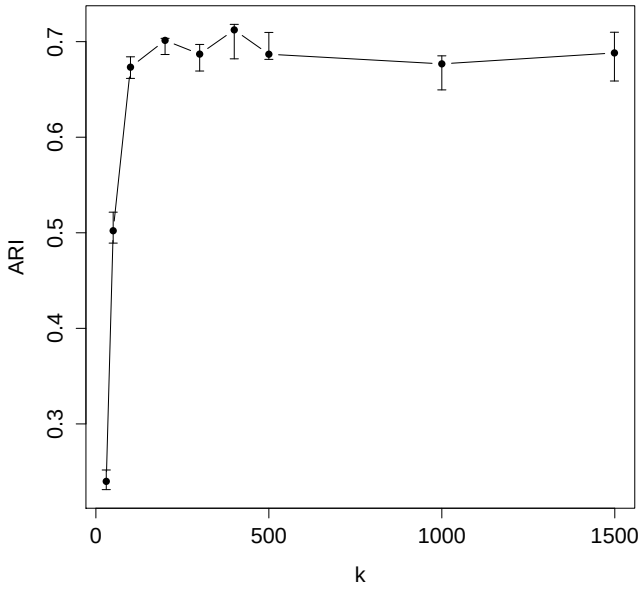


Figure 23: Adjusted Rand Index between the true partition and the maximum plausibility partition found by k-EVCLUS as a function of k, for the Gestures dataset. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.

the evolution of ARI, computing time and nonspecificity as a function of k follow the same patterns as in the previous examples: k-EVCLUS performs as well as EVCLUS for $k \geq 100$, with a significant reduction in training time.

Here again, RECM performed quite poorly on this dataset. With the same parameters as above, we obtained an average ARI (out of 10 trials) of 0.11 (with a maximum of 0.17). The mean computing time was 5.71s and the mean nonspecificity was 0.00621. For EK-NNclus with $K = 50$ and $q = 0.2$, the mean computing time over 10 trials was 4.16 s, the mean ARI was 0.59 (maximum: 0.66). The number of clusters ranged from 19 to 24, with a mean equal to 21.3.

In this section, we have shown that the principle underlying EVCLUS (i.e., constructing a credal partition by minimizing the discrepancy between degrees of conflict and dissimilarities) can be applied to large dissimilarity

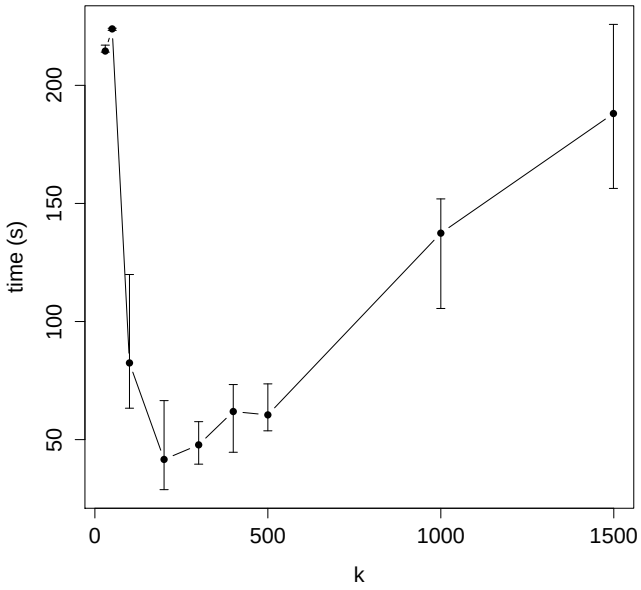


Figure 24: Computing time of k-EVCLUS as a function of k for the Gestures dataset. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.

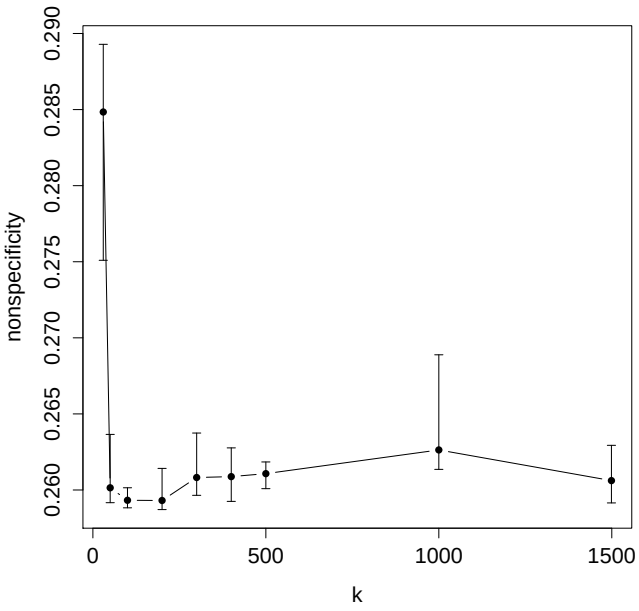


Figure 25: Nonspecificity of the maximum plausibility partition found by k-EVCLUS as a function of k, for the Gestures dataset. The error bars show the median as well as the lower and upper quartiles over 10 runs of the algorithm.

data sets, by randomly sampling the dissimilarities. The resulting method, called k-EVCLUS, is both faster than RECM (which is also limited by the requirement of storing the whole dissimilarity matrix), and more accurate than Ek-NNclus, provided that the number of clusters is correctly specified. In the next section, we address another issue with EVCLUS, which does not fully exploit the generality of credal partitions. We show how this limitation can be overcome by some simple post-processing operations.

5 Generating more informative credal partitions

As mentioned in the introduction, a limitation of the initial EVCLUS algorithm, as introduced in [7], is the fact that the complexity of the gradient calculation is proportional to f^3, where f is the number of focal sets in the credal partition. As a consequence, when using this algorithm, we need to drastically limit the number of the focal sets. The most stringent restriction that preserves the needed expressivity of the credal partition is to select as focal sets the empty set, the singletons, and the whole frame of discernent, in which case we have $f = c + 2$.

When using the IRQP algorithm introduced in Section 3.1, we no longer need to compute the gradient. However, there remains the problem that, if no restriction is imposed on the focal sets, the number of parameters in the optimization problem grows exponentially with the number of clusters. If we allow masses to be assigned to pairs of clusters, as suggested in [7] and [25], the number of focal sets becomes proportional to c^2, which is manageable for moderate values of c (say, until 10), but still makes the optimization of the stress function more difficult. It is clear, however, that only a few pairs of clusters will be assigned some mass during the learning process.

Example 5 *Consider, for instance, the S_2 dataset[4] shown in Figure 26. This dataset is composed on $n = 5000$ two-dimensional vectors grouped in 15 Gaussian clusters. We show the lower and upper approximations of the clusters obtained by EVCLUS with $c = 15$, $k = 100$, and d_0 fixed to the 0.2-quantile of the Euclidean distances. It is clear that, for instance, clusters 1 and 7 are not contiguous, and there can never be any ambiguity about assigning an object to one of these two clusters. Consequently, the mass assigned to the pair $\{\omega_1, \omega_7\}$ will always be null. In contrast, clusters 1 and 4, for instance, partially overlap: some objects are located at the boundary*

[4]This dataset is available at `https://cs.joensuu.fi/sipu/datasets`.

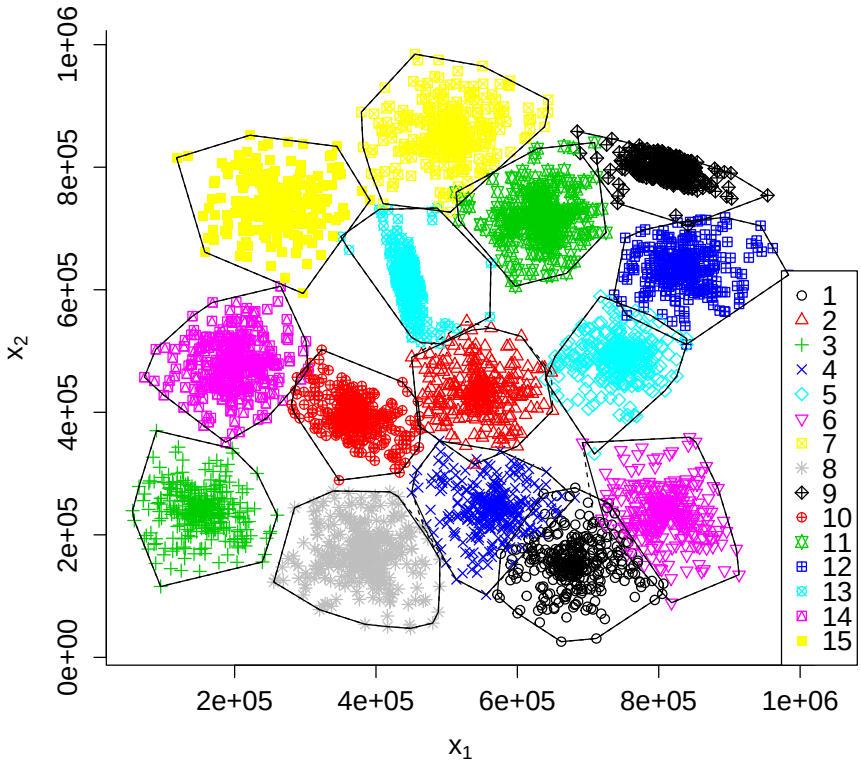


Figure 26: The S_2 dataset, and the 15 clusters found by k-EVCLUS with $k = 100$.

between these two clusters. For these objects, a positive mass should be assigned to the pair $\{\omega_1, \omega_4\}$. □

To determine which pairs of clusters can potentially become focal sets, we propose a two-step approach:

1. In the first step, k-EVCLUS is run in the basic configuration, with focal sets of cardinalities 0, 1 and c. A credal partition $\mathcal{M}_0$ is obtained. The similarity between each pair of clusters (ω_j, ω_ℓ) is measured by

$$S(j, \ell) = \sum_{i=1}^{n} pl_{ij} pl_{i\ell}, \tag{25}$$

where pl_{ij} and $pl_{i\ell}$ are the normalized plausibilities that object i belongs, respectively, to clusters j and ℓ. We then determine the set $\mathcal{P}_K$ of pairs $\{\omega_j, \omega_\ell\}$ that are mutual K nearest neighbors, according to the similarity measure S.

2. In the second step, k-EVCLUS is run again, starting from the previous credal partition $\mathcal{M}_0$, and adding as focal sets the pairs in $\mathcal{P}_K$.

Example 6 *Consider again the S_2 dataset displayed in Figure 26. The similarities $S(1, \ell)$ between cluster ω_1 and each of the $c = 15$ clusters are*

$$275.5, 9.2, 4.0, \underline{61.4}, 6.5, 59.8, 4.4, 7.1, 5.5, 8.3, 7.4, 6.7, 6.5, 4.4, 4.0,$$

and the similarities $S(4, \ell)$ between cluster ω_4 and the 15 clusters are

$$\underline{61.4}, 30.2, 5.8, 219.6, 8.8, 9.1, 6.2, 34.2, 7.3, 16.7, 9.2, 8.5, 8.3, 6.2, 5.8,$$

where we have underlined the highest similarity of each object with other objects (excluding itself). We can see that ω_4 is the nearest neighbor of ω_1, and ω_1 is the nearest neighbor of ω_4. By definition, they are mutual nearest neighbors. For this dataset, there are four pairs of mutual neighbors, out of the 105 pairs of clusters: $\{\omega_1, \omega_4\}$, $\{\omega_7, \omega_{11}\}$, $\{\omega_9, \omega_{12}\}$, $\{\omega_{10}, \omega_{14}\}$. Setting $k = 2$, we get 12 pairs of 2-nearest neighbors: in addition to the four pairs above, we have $\{\omega_1, \omega_6\}$, $\{\omega_3, \omega_8\}$, $\{\omega_4, \omega_8\}$, $\{\omega_2, \omega_{10}\}$, $\{\omega_9, \omega_{11}\}$, $\{\omega_5, \omega_{12}\}$, $\{\omega_3, \omega_{14}\}$ and $\{\omega_{13}, \omega_{15}\}$. □

Figure 27 shows the lower approximations of the clusters, and the points assigned to pairs of clusters by the interval dominance rule, for the initial credal partition $\mathcal{M}_0$. Only 9 points out of 5000 have an ambiguous classification, which does not reflect the actual ambiguity of the classification for some points at the boundaries between clusters. Figure 28 displays the clustering result after integrating the four pairs of clusters in $\mathcal{P}_1$. The 139 ambiguous points shown in Figure 28 are objects at the boundary between neighboring clusters. The final credal partition, which allows us to identify these points, is clearly more informative than the initial one. For this dataset with 5000 objects, the first and step steps took, respectively, 29.5 s and 31.3 s.

6 Conclusions

Among evidential clustering algorithms, EVCLUS has the distinctive advantage of being applicable to general non metric dissimilarity data. However,

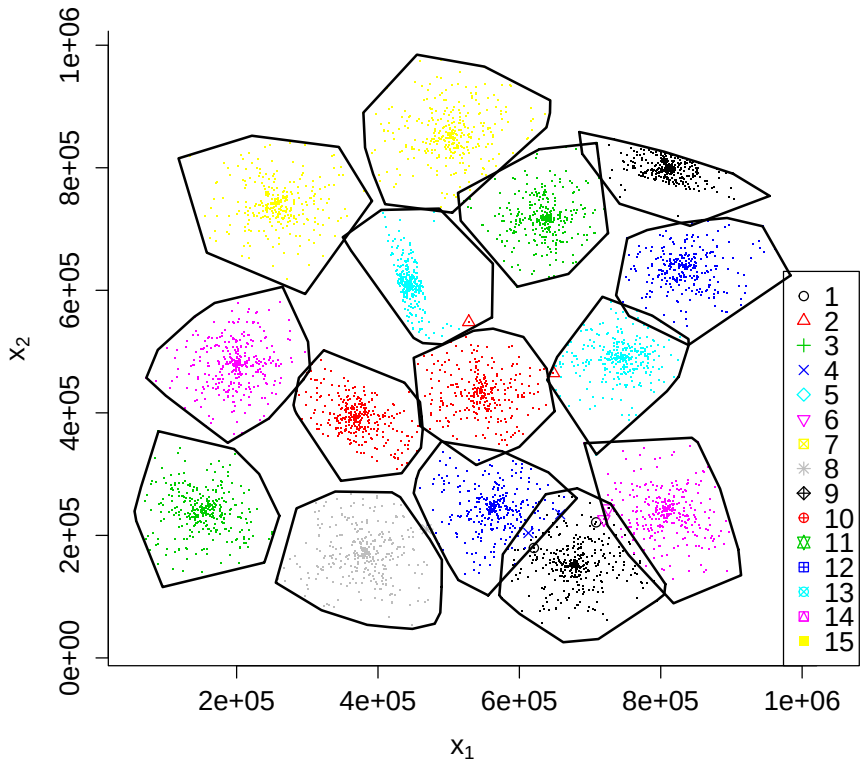


Figure 27: S_2 dataset: lower approximations and ambiguous objects for the initial credal partition $\mathcal{M}_0$ obtained by k-EVCLUS.

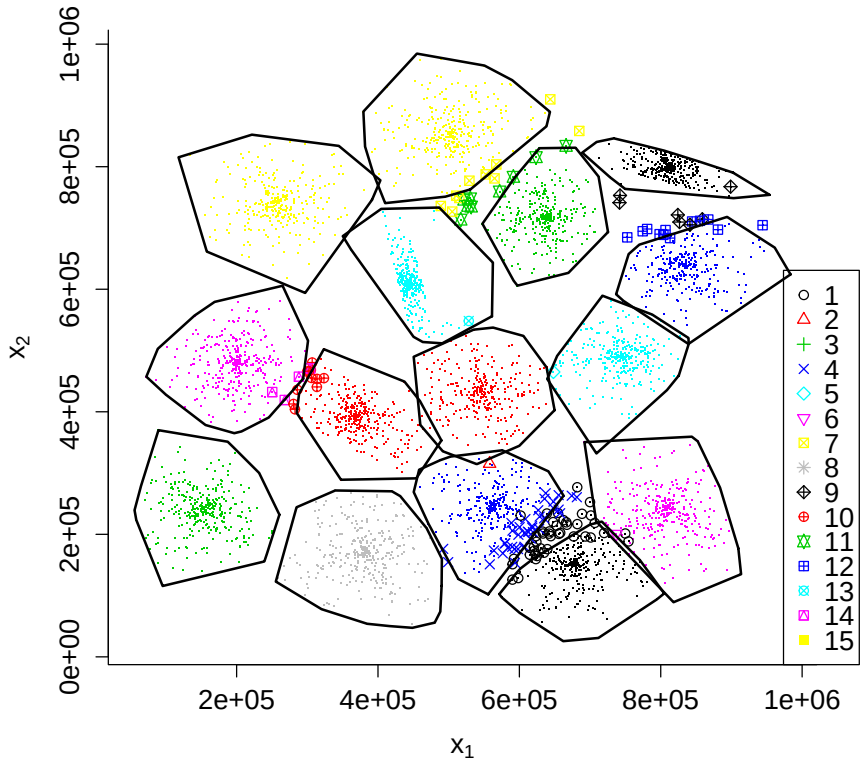


Figure 28: S_2 dataset: lower approximations and ambiguous objects for the final credal partition obtained by k-EVCLUS, after taking into account the four pairs of clusters that are mutual nearest neighbors.

in its original version, it also had a number of limitations. First, it was significantly slower than more recently introduced relational evidential clustering algorithms such as RECM and EK-NNclus. Because of its computational complexity, the expression power of belief functions could not be fully exploited, as the form of the focal sets had to be severely constrained. Finally, and most importantly, EVCLUS was limited to datasets of a few thousand objects, due to the necessity to store the whole dissimilarity matrix.

In this paper, we have been able to overcome these limitations, thanks to some major improvements to the original EVCLUS algorithm. First, the original gradient algorithm has been replaced by a much more efficient iterative row-wise quadratic programming procedure, which exploits the particular structure of the optimization problem. Secondly, we have shown that EVCLUS can only be provided with a randomly sampled subset of the dissimilarities. Specifically, we only need to supply the dissimilarities between each object and k randomly selected objects, reducing the space complexity from $O(n^2)$ to $O(kn)$. Our results suggest that, for a number n of objects between 1000 and 10,000, optimal results are obtained with k in the range 100-500. Finally, we have proposed a way to construct richer credal partitions, even with large numbers of classes, through a two-step procedure: in a first step, EVCLUS (or k-EVCLUS, the variant of EVCLUS with randomly sampled dissimilarities) is run with only the empty set, singletons and the whole set of clusters as focal sets; the similarity between clusters is computed, and pairs of neighboring clusters are identified. In a second step, the clustering algorithm is run again, starting from the previous solution, and adding to the focal sets the pairs of neighboring clusters found in the previous step. This simple procedure has been shown to provide more informative credal partitions, at the expense of a moderate increase on computing time, even for large numbers of clusters.

The improvements described in this paper make EVCLUS potentially applicable to large dissimilarity data, with of the order of 10^4 or even 10^5 objects. Analyzing even larger datasets (with millions of objects, as arising in social network studies, for instance), would probably require to sample the rows of the dissimilarity matrix. This issue obviously requires further investigation. Combining the ideas developed in this paper with the integration of instance-level constraints and active learning strategies, as introduced in [1], is also an interesting perspective.

Acknowledgements

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). It was also supported by the Center of Excellence in Econometrics at Chiang Mai University.

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] J. Bezdek. *Pattern Recognition with fuzzy objective function algorithm.* Plenum Press, New-York, 1981.

[3] I. Borg and P. Groenen. *Modern multidimensional scaling.* Springer, New-York, 1997.

[4] T. F. Cox and M. A. Cox. *Multidimensional scaling.* Chapman and Hall, London, 1994.

[5] T. Denœux. *evclust: Evidential Clustering*, 2016. R package version 1.0.0. url:`https://www.hds.utc.fr/~tdenoeux`

[6] 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.

[7] 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.

[8] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. In *Advances in Neural Information Processing Systems 11*, pages 438–444, Cambridge, MA, 1999. MIT Press.

[9] R. Hathaway and J. Bezdek. NERF c-means: Non-Euclidean relational fuzzy clustering. *Pattern Recognition*, 27:429–437, 1994.

[10] R. Hathaway, J. Bezdek, and J. Davenport. On relational data versions of c-means algorithms. *Pattern recognition Letters*, 17:607–612, 1996.

[11] R. Hathaway, J. Davenport, and J. Bezdek. Relational duals of the c-means clustering algorithms. *Pattern recognition*, 22(2):205–211, 1989.

[12] T. Hofmann and J. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, 1997.

[13] A. K. Jain and D. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1386–1391, 1997.

[14] M. A. Khalilia, J. Bezdek, M. Popescu, and J. M. Keller. An algorithm for linear least squares problems with equality and nonnegativity constraints. *Mathematical Programming*, 21:98–118, 1981.

[15] M. A. Khalilia, J. Bezdek, M. Popescu, and J. M. Keller. Improvements to the relational fuzzy c-means clustering algorithm. *Pattern Recognition*, 47(12):3920–3930, 2014.

[16] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Trans. on Fuzzy Systems*, 1:98–111, May 1993.

[17] 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.

[18] C. Lian, S. Ruan, and T. Denœux. An evidential classifier based on feature selection and two-step classification strategy. *Pattern Recognition*, 48:2318–2327, 2015.

[19] J. Lichtenauer, E. A. Hendriks, and M. J. T. Reinders. Sign language recognition by combining statistical DTW and independent classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:2040–2046, 2008.

[20] P. Lingras and G. Peters. Applying rough set concepts to clustering. In G. Peters, P. Lingras, D. Ślezak, and Y. Yao, editors, *Rough Sets: Selected Methods and Applications in Management and Engineering*, pages 23–37. Springer-Verlag, London, UK, 2012.

[21] Z.-G. Liu, Q. Pan, and J. Dezert. A new belief-based k-nearest neighbor classification method. *Pattern Recognition*, 46(3):834–844, 2013.

[22] Z.-G. Liu, Q. Pan, J. Dezert, and G. Mercier. Credal classification rule for uncertain data based on belief functions. *Pattern Recognition*, 47(7):2532–2541, 2014.

[23] Z.-G. Liu, Q. Pan, J. Dezert, and G. Mercier. Credal c-means clustering method based on belief functions. *Knowledge-Based Systems*, 74(0):119–132, 2015.

[24] 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.

[25] M.-H. Masson and T. Denoeux. ECM: an evidential version of the fuzzy c-means algorithm. *Pattern Recognition*, 41(4):1384–1397, 2008.

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

[27] G. Peters. Is there any need for rough clustering? *Pattern Recognition Letters*, 53:31–37, 2015.

[28] 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.

[29] 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.

[30] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, Princeton, N.J., 1976.

[31] 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.

[32] S. A. Vavasis. Complexity theory: quadratic programming. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization*, pages 304–307. Springer US, Boston, MA, 2001.

[33] M. Windham. Numerical classification of proximity data with assignment measures. *Journal of classification*, 2:157–172, 1985.

[34] 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.