EK-NNclus: A clustering procedure based on the evidential K-nearest neighbor rule

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

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

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

Abstract

We propose a new clustering algorithm based on the evidential K nearest-neighbor (EK-NN) rule. Starting from an initial partition, the algorithm, called EK-NNclus, iteratively reassigns objects to clusters using the EK-NN rule, until a stable partition is obtained. After convergence, the cluster membership of each object is described by a Dempster-Shafer mass function assigning a mass to each cluster and to the whole set of clusters. The mass assigned to the set of clusters can be used to identify outliers. The method can be implemented in a competitive Hopfield neural network, whose energy function is related to the plausibility of the partition. The procedure can thus be seen as searching for the most plausible partition of the data. The EK-NNclus algorithm can be set up to depend on two parameters, the number K of neighbors and a scale parameter, which can be fixed using simple heuristics. The number of clusters does not need to be determined in advance. Numerical experiments with a variety of datasets show that the method generally performs better than density-based and model-based procedures for finding a partition with an unknown number of clusters.

Keywords: Dempster-Shafer Theory, Evidence Theory, Hopfield neural networks, Unsupervised learning, Credal partition.

1 Introduction

Clustering may be defined as the search for groups in data, in an unsupervised way. Over the years, the initial concepts of hierarchical and partitional clustering have been extended to the search for more complex data structures, leading to the notions of fuzzy [2], possibilistic [19], rough [22], or credal clustering [7, 27]. In spite of the huge amount of work done in this area, the design of computationally efficient algorithms able to reveal an informative structure in data remains a topic of considerable interest.

In the so-called "decision-directed" approach to clustering [page 536] [10], prior knowledge is used to design a classifier, which is used to label the samples. The classifier is then updated, and the process is repeated until no changes occur in the labels. For instance, the well-known c-means algorithm is based on this principle: here, the nearest-prototype classifier is used to label the samples, and it is updated by taking as prototypes the centers of each cluster. The classification EM algorithm [4] is also based on the same principle, with an arbitrary parametric classifier and maximum likelihood estimation.

In recent years, new approaches to classification and clustering using the Dempster-Shafer theory of belief functions [5, 31] have been developed. For supervised classification, one of the most widely used method is the evidential K-nearest neighbor (EK-NN) rule [6,42]. Variants of this method have been proposed in [21,23,24,30,41]. This approach has been successfully applied in many domains including bioinformatics [32, 33, 38, 39], medical image processing [3], remote sensing [41], machine diagnosis [37], process control [34], among others. In unsupervised learning, the notion of credal partition has been introduced in [7]. In a credal partition, the member of an object to clusters is described by a Dempster-Shafer mass function. The ECM algorithm, a c-means-like algorithm that generates credal partitions, was introduced in [27]. Variants of this algorithm were proposed in [20, 26, 40].

In this paper, we introduce a new decision-directed evidential clustering algorithm based on the EK-NN rule. Starting from an initial random partition, the label of each sample is updated in turn using the EK-NN rule. We prove that this algorithm, (called Ek-NNclus) converges to a fixed point that corresponds, under some assumptions, to the most plausible partition of the data. We also show that the algorithm can be implemented in a competitive Hopfield neural network [15, 16], which allows its possible parallelization. The method is simple and depends on a small number of easily tunable parameters. In particular, it does not require to fix the number of clusters in advance. After convergence, one obtains a credal partition, which is more informative than a fuzzy partition and allows us to easily detect outliers.

The rest of this paper is organized as follows. The background on belief functions, the EKNN rule and credal clustering will first be recalled in Section 2. The new clustering algorithm will then be introduced and its theoretical properties will be studied in Section 3. Finally, experiments will be presented in Section 4 and Section 5 will conclude the paper.

2 Background

This section is intended to make the paper self-contained, by recalling recalling the necessary concepts related to belief functions (Section 2.1), the EK-NN rule (Section 2.2) and credal clustering (Section 2.3).

2.1 Belief functions

The theory of belief functions (also referred to as Dempster-Shafer, or evidence theory) is a framework for reasoning under uncertainty based on the explicit representation and combination of items evidence [5, 31]. Let us assume that we are interested in the value of some variable $\boldsymbol{\omega}$ taking values in a finite domain Ω, called the *frame of discernment*. Uncertain evidence about $\boldsymbol{\omega}$ may be represented by a *mass function* m on Ω, defined as a function from the powerset of Ω, denoted as 2^Ω, to the interval $[0, 1]$, such that $m(\emptyset) = 0$ and

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

Each number $m(A)$ is interpreted as the probability that the evidence supports exactly the assertion $\boldsymbol{\omega} \in A$ (and no more specific assertion), i.e., the probability of knowing that $\boldsymbol{\omega} \in A$, and nothing more. In particular, $m(\Omega)$ is the probability that the evidence tells us nothing about $\boldsymbol{\omega}$, i.e., it is the probability of knowing nothing. A subset A of Ω such that $m(A) \neq 0$ is called a *focal set* of m. The mass function for which Ω is the only focal set is said to be *vacuous*; it represent total ignorance.

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

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

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

for all $A \subseteq \Omega$. These two functions are linked by the relation $Pl(A) = 1 - Bel(\overline{A})$, for all $A \subseteq \Omega$. Each quantity $Bel(A)$ may be interpreted as the probability that the assertion $\boldsymbol{\omega} \in A$ is implied by the evidence, while $Pl(A)$ is the probability that this assertion is not contradicted by the evidence. 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.

A key idea in Dempster-Shafer theory is that beliefs are elaborated by aggregating independent items of evidence. Assume that we have two pieces of evidence represented by mass functions m_1 and m_2 on the same frame of discernment Ω. If one piece of evidence tells us that $\boldsymbol{\omega} \in A$ and the other source tells us that $\boldsymbol{\omega} \in B$ for some non-disjoint subsets A and B of Ω, and if both sources are reliable, then we know that $\boldsymbol{\omega} \in A \cap B$. Under the independence assumption, the probabilities $m_1(A)$ and $m_2(B)$ should be multiplied. If, however, A and B are disjoint, we can conclude that the interpretations "$\boldsymbol{\omega} \in A$" and '$\boldsymbol{\omega} \in B$" cannot hold jointly, and the probabilities must be conditioned to eliminate such pairs of interpretations. This line of reasoning leads to the following combination rule, referred to as Dempster's rule [31],

$$(m_1 \oplus m_2)(A) = \frac{1}{1-\kappa} \sum_{B \cap C = A} m_1(B) m_2(C) \tag{3a}$$

for all $A \subseteq \Omega$, $A \neq \emptyset$ and $(m_1 \oplus m_2)(\emptyset) = 0$, where

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

is the *degree of conflict* between m_1 and m_2. If $\kappa = 1$, there is a logical contradiction between the two pieces of evidence and they cannot be combined. Dempster's rule is commutative, associative, and it admits the vacuous mass function as neutral element.

Whereas the computation of the full combined mass function $m_1 \oplus m_2$ may be prohibitive in very large frames of discernment, the corresponding contour function can be computed in time proportional to the size of the frame, using the following property,

$$pl_1 \oplus pl_2 = \frac{pl_1 pl_2}{1-\kappa}, \tag{4}$$

where pl_1 and pl_2 are the contour functions of two mass functions m_1 and m_2, and the same symbol $\oplus$ is used for mass functions and contour functions.

2.2 EK-NN rule

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. If object o_j belongs to group $\omega_{k(j)}$, then 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,

$$m_j(\{\omega_{k(j)}\}) = \alpha_j, \tag{5a}$$

$$m_j(\Omega) = 1 - \alpha_j, \tag{5b}$$

with

$$\alpha_j = \varphi(d_j), \tag{5c}$$

where φ is a non-increasing mapping from $[0, +\infty)$ to $[0, 1]$, such that

$$\lim_{d \to +\infty} \varphi(d) = 0. \tag{6}$$

According to (5), the mass function m_j has two focal sets: the class $\omega_{k(j)}$ of o_j, and Ω. It becomes vacuous when d_j becomes infinitely large. In [6], it was proposed to choose φ as $\varphi(d_j) = \alpha_0 \exp(-\gamma_k d_j)$ for some constants α_0 and γ_k. Considering the distances to the n objects in the database as independent pieces of evidence, the n mass function m_j can then be combined by Dempster's rule to yield the combined mass function

$$m = m_1 \oplus m_2 \oplus \ldots \oplus m_n. \tag{7}$$

For computational reasons, the mass functions m_j for objects o_j that are very dissimilar to object o are nearly vacuous and can be neglected. A useful heuristic is to consider only the K *nearest neighbors* of object o in the database. Denoting by N_K the set of indices of these nearest neighbors, the combined mass function thus becomes

$$m = \bigoplus_{j \in N_K} m_j. \tag{8}$$

If a decision has to be made, one can then assign 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 (5) is

$$pl_j(\omega_\ell) = \begin{cases} 1 & \text{if } \ell = k(j), \\ 1 - \alpha_j & \text{otherwise,} \end{cases} \tag{9a}$$

$$= (1 - \alpha_j)^{1 - s_{j\ell}} \tag{9b}$$

for $\ell = 1, \ldots, c$, where $s_{i\ell} = 1$ if object o_i belongs to class ℓ, and $s_{i\ell} = 0$ otherwise. From (4), the combined contour function is thus

$$pl(\omega_\ell) \propto \prod_{j \in N_K} (1 - \alpha_j)^{1 - s_{j\ell}}, \tag{10}$$

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

$$\ln pl(\omega_\ell) = \sum_{j \in N_K} (1 - s_{j\ell}) \ln(1 - \alpha_j) + C \tag{11a}$$

$$= - \sum_{j \in N_K} s_{j\ell} \ln(1 - \alpha_j) + C' \tag{11b}$$

$$= \sum_{j=1}^{n} v_j s_{j\ell} + C', \tag{11c}$$

where C and C' are constants, and $v_j = -\ln(1 - \alpha_j)$ if $j \in N_K$, and $v_j = 0$ otherwise.

Equations (5)-(11) define the EKNN rule [6]. In [8], it was shown that the coefficient α_j in (5) can be interpreted as a mass of belief on the hypothesis that objects o and o_j belong to the same class. The EK-NN rule can then be derived as Dempster's rule applied to suitable mass functions.

Some modifications of the EKNN rule have been proposed recently. For instance, in [23], Liu et al. propose to replace Dempster's rule in (8) by a two-step combination process: with each class, mass functions are first averaged; then, the average class-conditional mass functions are combined by the Dubois-Prade (DP) rule, a variant of Dempster's rule that assigns the conflicting mass to unions of focal sets [9]. In [24], the same authors propose a more complex scheme, in which the mass functions m_i corresponding to each neighbor are constructed by aggregating two mass functions focused, respectively, on a class and on its complement. Within each class, the resulting mass functions are then combined by Dempster's rule. Finally, mass functions across classes are combined using a variant of the DP rule, in which masses are assigned only to some preselected subsets of classes. In [25], the authors propose to use the K nearest neighbors in each class; the mass functions are averaged within each class, while a modified DP rule is used to combine the class-conditional mass functions. Generally, these alternative algorithms are designed to assign masses to sets of classes (also called "meta-classes"). While this objective may be appealing, and the resulting methods may have good performances for some datasets, these modifications to the basic algorithm also make it more complex. In particular, it is no longer possible, using these modified algorithms, to compute the combined contour function using such a simple formula as (10). For this reason, we will stick to the original version of the EKNN rule in this paper.

2.3 Credal clustering

Assume that we have n objects, each one belonging to one of c groups or clusters. Let $\Omega = \{\omega_1, \ldots, \omega_c\}$ be the set of clusters. If we know for sure which cluster each object belongs to, we can give a partition of the n objects. Such a partition may be represented by binary variables u_{ik} such that $u_{ik} = 1$ if object i belongs to cluster 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 function $m_1, \ldots, m_n$, where each mass function m_i is defined on Ω and describes the uncertainty about the cluster of object i. Such a n-tuple of mass functions is called a credal partition [7].

It is clear that the concept of credal partitions subsumes the most common clustering structures. If each mass function has only focal set, and this focal set is a singleton, then we are back to the full certainty case and we have a hard partition. If all focal sets are singletons, then we have

$$\sum_{k=1}^{c} m_i(\{\omega_k\}), \quad i = 1, \ldots, n, \tag{12}$$

and the credal partition becomes a fuzzy partition [2]. If each mass function m_i has only one focal set A_i, then we can define the inner approximation of cluster k as the set of objects i such that $A_i = \{\omega_k\}$, and outer approximation of cluster k as the set of objects i such that $\omega_k \in A_i$. We thus recover concepts of rough clustering [22].

The first algorithm that was proposed for computing a credal partition is the EVCLUS algorithm [7]. This algorithm works with dissimilarity data and does not require the dissimilarities to be Euclidean. The Evidential c-Means (ECM) algorithm was then introduced in [27]. It 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"). The ECM algorithm works with attribute data, but a relational data version was proposed in [28]. In [26], a variant of the ECM algorithm (called CCM) is proposed, based on an alternative definition of the distance between a vector and the prototype of a meta-cluster. This modification produces more sensible results in situations where the prototype of a meta-cluster is close to that of singleton cluster. In [40], Zhou et al. introduce another variant of ECM, called Median Evidential c-means (MECM), which is an evidential counterpart to the median c-means and median fuzzy c-means algorithms. An advantage of this approach is that it does not require the dissimilarities between objects to verify the axioms of distances.

Whereas a credal partition provides a rich description of the data structure, it implies, in the general case, the storage of $O(2^c)$ numbers, which can become prohibitive for large n. To make this notion operational for datasets with a large number of clusters, we need to restrict the form of the mass functions m_i. For moderate values of c, we can restrict the focal sets to have size at most two, or to be equal to Ω. For large values of c, we can be even more drastic and restrict the focal sets to singletons, and Ω. This is the kind of credal partition generated by the algorithm described in the next section.

3 EK-NNclus algorithm

In this section, we will show how the EK-NN rule recalled in Section 2.2 can be used for clustering. The algorithm will first be described in Section 3.1. A neural implementation and a proof of convergence will be presented in Section 3.2. Finally, an interpretation of the criterion optimized by the algorithm will be discussed in Section 3.3.

3.1 Description of the algorithm

As mentioned in the introduction, the decision-directed approach to clustering is to use a classifier to label the objects, and then to use the labeled objects to train the classifier. These two steps are iterated until the object labels are no longer modified. The EK-NNclus algorithm introduced in this section is based on this principle, using the EK-NN rule as a classifier. As this classification rule is non parametric, there is no training step, and the clustering procedure becomes very simple. It can be described as follows.

Preparation Let $D = (d_{ij})$ be a symmetric $n \times n$ matrix of distances or dissimilarities between the n objects. They can be computed from attributes, or directly available. The numbers d_{ij} need not be Euclidean distances, nor do they need not be distances at all. Given K, we compute the set $N_K(i)$ of indices of the K nearest neighbors of object i. We then compute

$$\alpha_{ij} = \begin{cases} \varphi(d_{ij}) & \text{if } j \in N_K(i) \\ 0 & \text{otherwise,} \end{cases} \tag{13a}$$

$$v_{ij} = -\ln(1 - \alpha_{ij}), \tag{13b}$$

for all $(i, j) \in \{1, \ldots, n\}^2$. We assume $\alpha_{ij} < 1$ for all i and j. If computing time is not an issue, K can be chosen very large, even equal to $n - 1$.

Initialization 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 s_{ik} as $s_{ik} = 1$ is object o_i belongs to cluster k, and $s_{ik} = 0$ otherwise.

Iteration An iteration of the algorithm 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 (11) as

$$u_{ik} = \sum_{j \in N_K(i)} v_{ij} s_{jk}, \quad k = 1, \ldots, c. \tag{14}$$

We then assign object o_i to the cluster with the highest plausibility, i.e., we update the variables s_{ik} as

$$s_{ik} = \begin{cases} 1 & \text{if } u_{ik} = \max_{k'} u_{ik'}, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

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 (it will be the case, in particular, if the initial number of clusters is very large). 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.

Computation of the credal partition After the algorithm has converged, we can compute the final mass functions

$$m_i = \bigoplus_{j \in N_K(i)} m_{ij}, \tag{16}$$

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

$$m_{ij}(\{\omega_{k(j)}\}) = \alpha_{ij}, \tag{17a}$$

$$m_{ij}(\Omega) = 1 - \alpha_{ij}. \tag{17b}$$

The procedure is summarized in Algorithm 1.

The procedure described above iterative changes the object labels, starting from an initial configuration. In Section 3.2 below, we will show that this procedure can be implemented in a special kind of Hopfield neural network, and we present a proof of convergence of the procedure. But before that, let us present two simple examples to illustrate qualitatively the behavior of the algorithm.

Example 1 *Consider, as a first example, the toy "Butterfly" dataset displayed in Figure 1(a). It is composed of 12 objects described by two attributes. Point 12 is an outlier, while point 6 is situated half-way between the two clusters. We applied the EK-NNclus algorithm to this dataset, with the following settings. Initially, each point was assigned to a different cluster, so that the initial number of clusters was $c = 12$. We computed the*

Algorithm 1 EK-NNclus algorithm.

```
Require: Number of states $c$, distance matrix $D = (d_{ij})$, number of neighbors $K$
  Randomly initialize variables $s_{ik}$ for $i = 1, \ldots, n$; $k = 1, \ldots, c$.
  Compute $\alpha_{ij}$ and $v_{ij}$ for $i = 1, \ldots, n$; $j = 1, \ldots, n$
  $change \leftarrow$ true
  while $change$ do
    Select a random permutation $\sigma$ of $\{1, \ldots, n\}$
    $change \leftarrow$ false
    for $i = 1$ to $n$ do
      for $k = 1$ to $c$ do
        $u_{\sigma(i)k} \leftarrow \sum_{j \in N_K(\sigma(i))} v_{\sigma(i)j} s_{jk}$
      end for
      $k^* \leftarrow \arg\max_k u_{\sigma(i)k}$
      if $s_{\sigma(i)k^*} = 0$ then
        Set $s_{\sigma(i)k^*} \leftarrow 1$ and $s_{\sigma(i)k} \leftarrow 0$ for all $k \neq k^*$
        $change \leftarrow$ true
      end if
    end for
    Update $c$, renumber the clusters and change variables $s_{ik}$ accordingly
  end while
  for $i = 1$ to $n$ do
    Initialize $m_i$ as the vacuous mass function on $\Omega = \{\omega_1, \ldots, \omega_c\}$.
    for $j = 1$ to $K$ do
      $m_i \leftarrow m_i \oplus m_{ij}$
    end for
  end for
```

Euclidean distances d_{ij} between data points, and function φ was defined as $\varphi(d_{ij}) = \exp(-\gamma d_{ij}^2)$. The number of neighbors was set, successively, to $K = 11$, $K = 8$ and $K = 4$. Coefficient γ was fixed as the inverse of median of the square distances between each point and its K nearest neighbors,

$$\gamma = 1/\text{median}(\{d_{ij}^2, i \in \{1, \ldots, n\}, j \in N_K(i)\}) \tag{18}$$

The algorithm always converged to two clusters, in two to four iterations. The masses $m(\{\omega_1\})$, $m(\{\omega_2\})$ and $m(\Omega)$ for the 12 objects are shown in Figures 1(b), 1(c) and 1(d) for, respectively, $K = 11$, $K = 8$ and $K = 4$. We can see that the results remain similar for different values of K. The outlier (point 12) is characterized by a large mass assigned to Ω. For the *"borderline" object 6, the unit mass is shared equally between singletons ω_1 and ω_2.*

Example 2 *As a second example, consider the dataset in Figure 2(a), consisting of three Gaussian clusters (with 20 points in each) and 10 points generated from the uniform distribution in $[-1, 2]^2$. The number K of neighbors was set to 30. Function φ and parameter γ were defined as in Example 1. As before, each data point was initially assigned to a separate cluster. The obtained partition in three clusters is shown in Figure 2(a). In Figure 2(b), the size of each point is proportional to the mass $m_i(\Omega)$ assigned to the set of clusters. We can see that outliers are easily identified as objects with large values of $m_i(\Omega)$.*

3.2 Neural network implementation

The algorithm described in the previous section can be implemented exactly in a competitive Hopfield model such as described in [15, 35]. This model is a variant of the discrete neural network model initially proposed by Hopfield [16]. Its architecture is composed of n groups of c neurons. Here, we have one group for each object to be classified, and, within each group, one neuron for each cluster. Let us denote by $s_{ik} \in \{0, 1\}$ the state of neuron k in group i and by v_{ij} the weight of the connection between groups i and j. The assignment of object i to cluster k corresponds to the activation of neuron k in group i, i.e., $s_{ik} = 1$. Using the terminology of neural networks, the term u_{ik} in (14) corresponds to the input of neuron k in group i. Within each group, only the neuron with the highest input fires, i.e., its state s_{ik} becomes equal to 1, according to (15). Neuron groups are updated one at a time, in serial mode.

In Hopfield networks, the weights are usually assumed to be positive and symmetric, i.e., it is assumed that $v_{ij} \geq 0$ and $v_{ij} = v_{ji}$, for all i and j. Here, the positivity condition is always met. To guarantee the symmetry condition, we have to assume that $K = n - 1$, or to change Eq. (13) to $\alpha_{ij} = \alpha_{ji} = \varphi(d_{ij})$ if $j \in N_K(i)$ and $i \in N_K(j)$, and $\alpha_{ij} = 0$, otherwise. When the weights are symmetric, it is easy to show that the neural network converges a stable state corresponding to a local minimum of the following energy function,

$$E(S) = -\frac{1}{2} \sum_{k=1}^{c} \sum_{i=1}^{n} \sum_{j \neq i} v_{ij} s_{ik} s_{jk}, \tag{19}$$

where $S = (s_{ik})$ denotes the $n \times c$ matrix of 0s and 1s encoding the neuron states. To see this, assume that the cluster label of object o_i is changed from

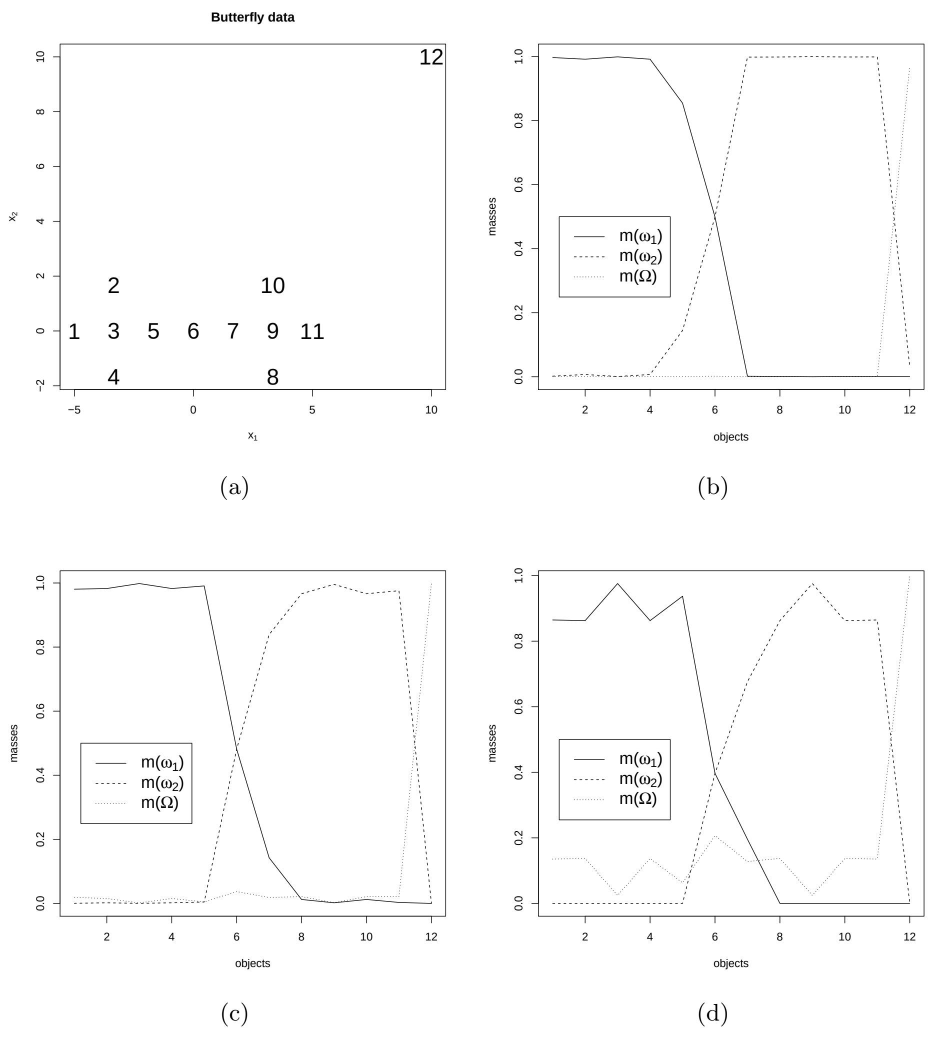
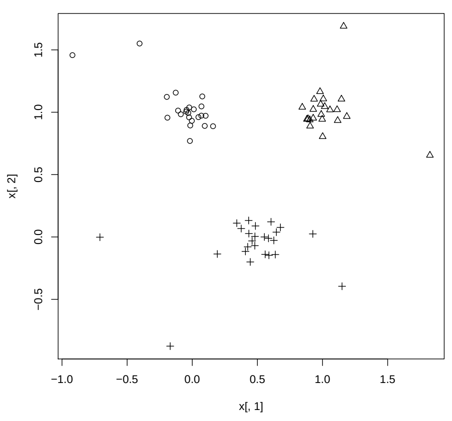
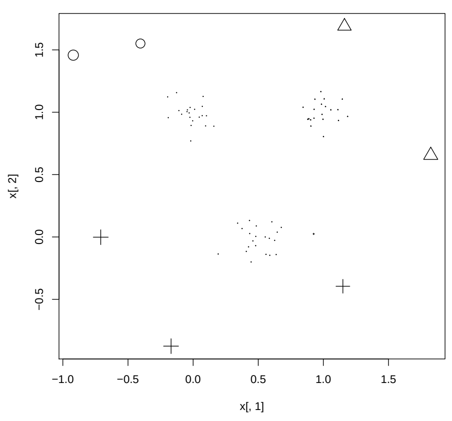


Figure 1: Butterfly dataset (a) and results of the EK-NNclus algorithm with $K = 11$ (b), $K = 8$ (c) and $K = 4$ (d).



(a)



(b)

Figure 2: Dataset of example 2: partition obtained by the E-K–NNclus algorithm (a) and masses $m_i(\Omega)$ (b).

k to k^*. The corresponding change of $E(S)$ is

$$\Delta E = +\sum_{j\neq i} v_{ij}s_{jk} - \sum_{j\neq i} v_{ij}s_{jk^*} = u_{ik} - u_{ik^*}, \tag{20}$$

which is strictly negative, since, by definition, $u_{ik^*} > u_{ik}$. As there is a finite number of states, the sequence of states produced by the algorithm is stationary, and the algorithm converges to a local minimum of the energy function E.

When the weights are not symmetric, some additional conditions should be met to insure convergence. In [36], Xu et al. present an analysis for the standard Hopfield model with asymmetric weights, but we are not aware of any similar results for the competitive model. In [16], Hopfield states, without proof, that the standard Hopfield network still converges when the following condition is met: if $v_{ij} > 0$ then $v_{ji} = 0$. Here, if the weights v_{ij} are fixed according to (13), we have a weaker form of asymmetry: if $v_{ij} > 0$ and $v_{ji} > 0$ then $v_{ij} = v_{ji}$. In simulations, the algorithm was always found to converge to a stable state, and it can be conjectured that the convergence property still holds in that case.

Example 3 *Figure 3 shows the decrease of the energy criterion, plotted as function of time, for five runs of the EK-NNclus algorithm, applied to the*

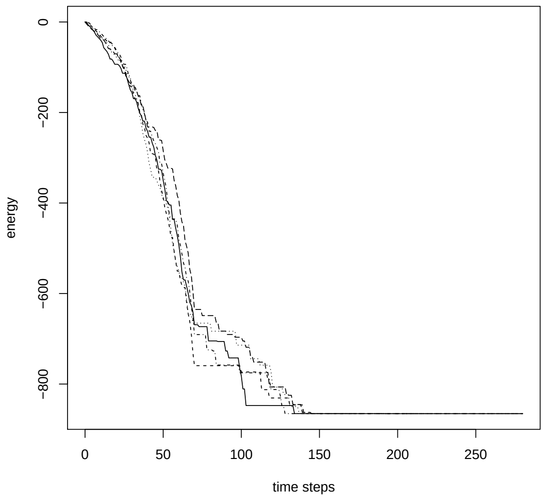


Figure 3: Decrease of the energy function for five runs of the EK-NNclus algorithm, applied to the data of Example 2.

data of Example 2. The algorithm always converged after two iterations (presentations of the whole dataset).

3.3 Interpretation of the energy function

We have just seen that the EK-NNclus algorithm converges to a local minimum of energy function (19). In this section, we show that the energy $E(S)$ is related to its plausibility of the partition encoded by S, lower-energy partitions being more plausible. The EK-NNclus algorithm can thus be seen as *searching for the most plausible partition*, among all the partitions of a dataset.

To see this, let us denote by $\mathcal{O} = \{o_1, \ldots, o_n\}$ the set of objects, $\mathcal{P}$ the set of partitions of $\mathcal{O}$ and $\mathcal{R}$ the set of equivalence relations on $\mathcal{O}$. Because the sets $\mathcal{P}$ and $\mathcal{R}$ are in one-to-one correspondence, we can reason equivalently using one set or another. In the following, we will place the emphasis on equivalence relations, for notational convenience. We will assume that there exists an unknown true relation R_0 in $\mathcal{R}$, about which we collect pieces of evidence in the form of pairwise distances d_{ij}.

As suggested in [7], we make the weak assumption that two objects have all the more chance to belong to the same cluster, that they are more similar.

Formally, let us denote by $\mathcal{R}_{ij}$ the set of equivalence relations R such that $(o_i, o_j) \in R$, i..e., such that objects o_i and o_j belong to the same cluster. The distance d_{ij} between objects o_i and o_j can be interpreted as a piece of evidence, which can be represented by the following mass function,

$$m_{ij}(\mathcal{R}_{ij}) = \alpha_{ij}, \tag{21a}$$

$$m_{ij}(\mathcal{R}) = 1 - \alpha_{ij} - \beta_{ij}, \tag{21b}$$

with $\alpha_{ij} = \varphi(d_{ij})$. Having collected $N = n(n-1)/2$ such independent mass functions (possibly vacuous), one for each pair of objects, we can combine by Dempster's rule. As the number of focal sets of the combined mass function m grows exponentially with N, the full computation of m will generally be intractable. However, thanks to property (4), the plausibility of any relation R in $\mathcal{R}$ can easily be computed up to a multiplicative constant, by reasoning as follows.

For any $R \in \mathcal{R}$, let $pl_{ij}(R)$ denote the plausibility of R induced by mass function m_{ij}. By convention, let us use the same symbol to denote an equivalence relation R and its adjacency matrix, i.e., $R_{ij} = 1$ if $(o_i, o_j) \in R$ and $R_{ij} = 0$ otherwise. With these notations, we have

$$pl_{ij}(R) = \begin{cases} m_{ij}(\mathcal{R}_{ij}) + m_{ij}(\mathcal{R}) & \text{if } R \in \mathcal{R}_{ij}, \\ m_{ij}(\mathcal{R}) & \text{otherwise,} \end{cases} \tag{22a}$$

$$= \begin{cases} 1 & \text{if } R_{ij} = 1, \\ 1 - \alpha_{ij} & \text{otherwise,} \end{cases} \tag{22b}$$

for all $R \in \mathcal{R}$, which can be expressed more concisely as follows,

$$pl_{ij}(R) = (1 - \alpha_{ij})^{1 - R_{ij}}. \tag{23}$$

From (4), the plausibility of R induced by the combined mass function m (obtained by Dempster's rule) is proportional to the product of the N mass functions pl_{ij}:

$$pl(R) \propto \prod_{i<j} (1 - \alpha_{ij})^{1 - R_{ij}}. \tag{24}$$

Assuming that $\alpha_{ij} < 1$ for all i and j, the logarithm of $pl(R)$ is given by

$$\ln pl(R) = -\sum_{i<j} R_{ij} \ln(1 - \alpha_{ij}) + C, \tag{25}$$

$$= -E(R) + C, \tag{26}$$

where C is a constant and

$$E(R) = -\sum_{i<j} v_{ij} R_{ij} \tag{27}$$

is identical to energy function (19), with $v_{ij} = -\ln(1-\alpha_{ij})$ and $R_{ij} = \sum_{k=1}^{c} s_{ik}s_{jk}$.

The EK-NNclus algorithm described in Section 3 thus searches for a partition with maximum plausibility, based on the evidence of pairwise distances between objects. However, we can remark that the global maximum of $pl(R)$ corresponds to the trivial partition for which all objects belong to the same class, in which case we have $R_{ij} = 1$ for all i and j. We are obviously not interested in this solution. It is thus essential that the algorithm converges to a local maximum. Starting from a the finest partition with as many clusters as objects can be seen as a way of incorporating prior knowledge to drive the algorithm away from the trivial global maximum. In the next section, the effectiveness of this procedure will be demonstrated by numerical experiments.

4 Experiments

In this section, we present some experimental results with real and simulated data, showing the effectiveness of the EK-NNclus algorithm. A sensitivity analysis will first be presented in Section 4.1 using the Wine dataset. Comparative experiments will then be reported in Section 4.2.

4.1 Sensitivity analysis

To study the behavior of the EK-NNclus, we first considered the Wine dataset from the UCI Machine Learning Repository[1]. The data results form chemical analysis of 178 Italian wines. Each wine is described by 13 attributes which are the compositions of chemical constituents. The ground-truth partition consists in three groups.

We first centered and scaled the variables, and then ran EK-NNclus with K ranging between 15 and 50 (with increments of 5). As in Example 1, we set $\varphi(d_{ij}) = \exp(-\gamma d_{ij}^2)$, where d_{ij} is the Euclidean distance between objects i and j. Parameter γ was fixed to the inverse of the q-quantile of the set

$$\Delta = \{d_{ij}^2, i \in \{1, \ldots, n\}, j \in N_K(i)\}.$$

[1]This dataset is available at `https://archive.ics.uci.edu/ml/datasets.html`.

Parameter q was varied in the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. For each pair (K, q), the algorithm was run 100 times, starting from the finest partition (with exactly one cluster for each object). The results were evaluated using the Adjusted Rand Index (ARI) [17]. We recall that this index equals 1 for two equal partitions, and equals 0 on average when a partition is compared to a randomly-generated one. We note that the ARI only compares hard partitions, i.e., the additional information contained in the credal partition is not evaluated. It is not clear, however, how the relevance of this information could be measured, all the more so when comparing clustering algorithm generating different representations (such as hard and fuzzy clustering algorithms).

Boxplots of the ARI criterion for the different values of K and q are shown in Figures 4 and 5. We can see that the algorithm performs well for $K \geq 30$ and $q \geq 0.3$. For this dataset, the performance of the algorithm is thus not too sensitive to these parameters. We can also see that a too small value of K (such as $K = 20$ or $K = 25$) can be compensated by choosing a large value of q, i.e., a small value of γ.

The algorithm found, in most cases, a partition in three classes. For instance, for $K = 40$ and $q = 0.5$, a there-class partition was found in 99 runs out of 100, and a two-class partition was found only once. Figure 6 shows a typical partition, corresponding to an ARI of 0.897. The data are displayed in the space of the first two principal components. Only 6 objects out of 178 are misclassified.

The results obtained with EK-NNclus were also compared with those of the following clustering algorithms:

- HCM: Hard c-means (function `kmeans` in the R package `stats`);
- FCM: Fuzzy c-means (function `FKM` in the R package `fclust`);
- A density-based nonparametric clustering method [1,29] (function `pdfCluster` in the R package `pdfCluster`);
- Model-based clustering with mixtures of Gaussian and the EM algorithm (function `Mclust` in the R package `mclust`) [11, 12].

The HCM and FCM algorithms were run 100 times with $c = 3$ clusters. In each case, the result with the best value of the objective function was kept. The ARI values were 0.754 and 0.897 for HCM and FCM, respectively. The FCM algorithm thus performs as well as EK-NNclus for this dataset, with the important provision that it needs to be supplied with the correct number of clusters.

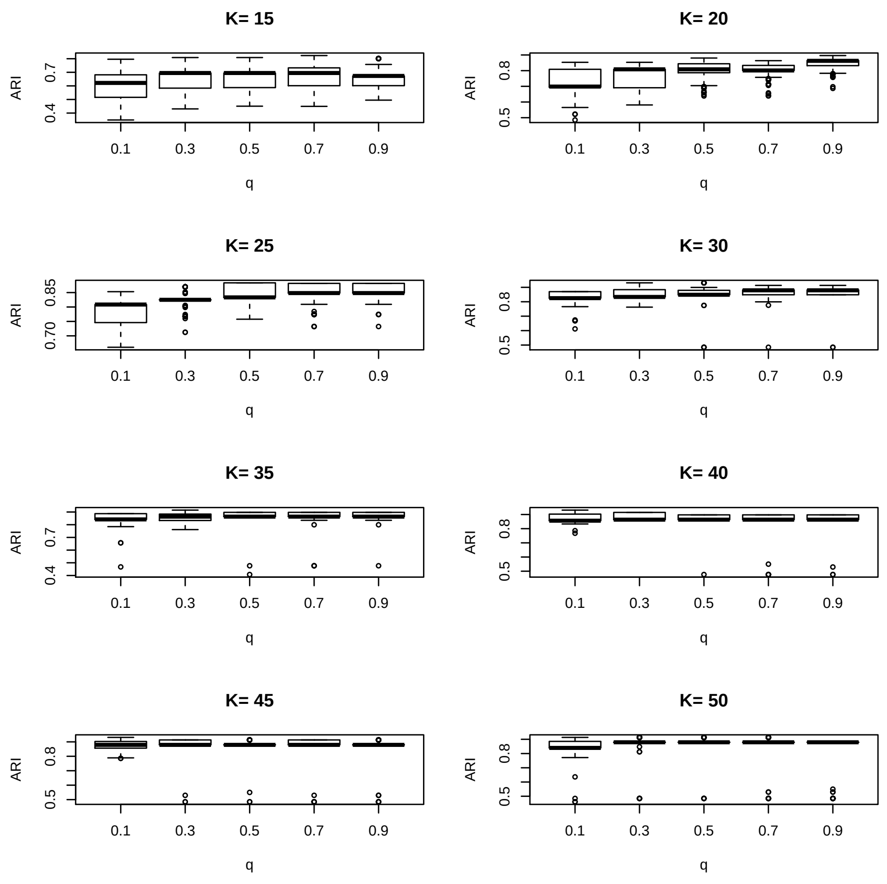


Figure 4: Adjusted Rand Index as a function of parameter q, for different values of the number K of neighbors.

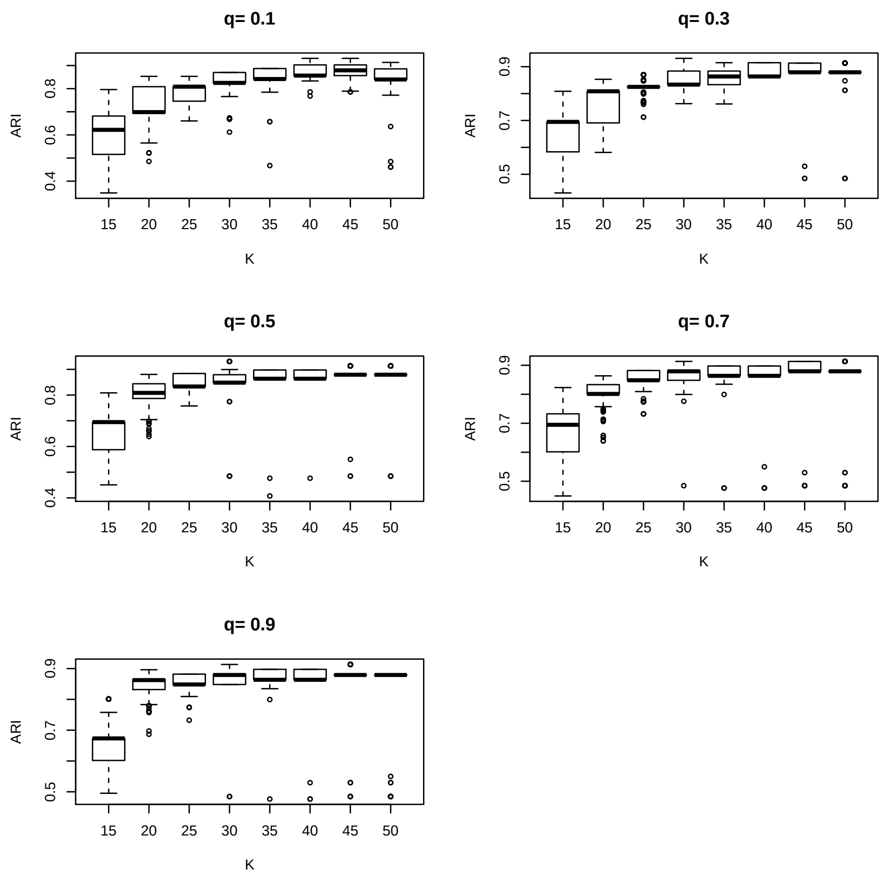


Figure 5: Adjusted Rand Index as a function of the number K of neighbors, for different values of parameter q.

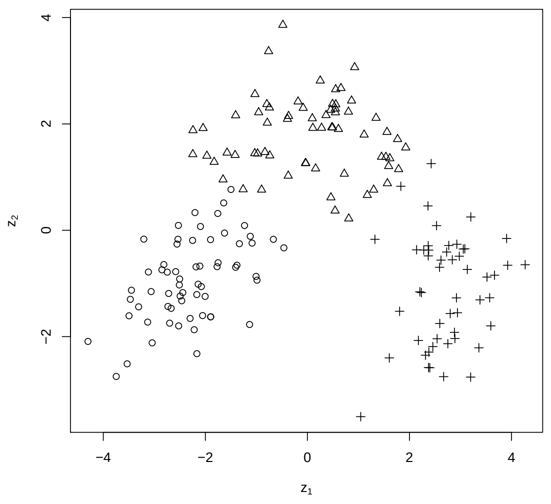


Figure 6: Wine data in the space of the first two principal components, and partition generated by the EK-NNclus algorithm.

The other two alternative methods have mechanisms for determining the number of clusters automatically. The `pdfCluster` procedure computes a kernel density estimate of the data and searches for modes of the estimated density. The `Mclust` procedures searches through different number of components of the Gaussian mixture, and different assumptions (such as spherical, diagonal or ellipsoidal shapes, and equal volume of shape of clusters) by maximizing the Bayesian Information Criterion (BIC). Here, we specified a number of clusters between 1 and 9.

With the default settings, the `pdfCluster` procedure found 10 clusters with an ARI of 0.471. This poor result may be due to the fact that non-parametric density estimation works well only in low dimensional spaces. We thus performed Principal Component Analysis (PCA) of the data to extract informative features (components). When applied to the space of the first 2, 3 and 4 components, we obtained, respectively values of the ARI equal to 0.803, 0.832 and 0.689. The correct number of clusters was obtained for 3 and 4 components. The `pdfCluster` procedure thus works better after performing some dimensionality reduction, but it requires to perform a search for the best reduced space.

The model-based approach also did not perform too well on this dataset.

Without any restriction on the form of the clusters, we obtained 8 components and $ARI = 0.481$. By restricting the search to the simplest model (spherical cluster shape and equal volume), we get better results, with 4 components and $ARI = 0.78$. Fixing the number of components to 3, we get $ARI = 0.897$. It thus appears that the method works well when supplied with the correct number of clusters, but it fails to identify that number automatically.

From these experiments, we can conclude that, for this dataset, the performances of the EK-NNclus algorithm are not too sensitive to the choice of the two parameters K and q, and that this method performs better than the alternative clustering algorithms considered in this study. In particular, the method seems able to correctly identify the true number of clusters. In the following section, we will report results from experiments with a larger number of datasets.

4.2 Comparative experiments

In this section, we present the results of comparative experiments with different datasets available from "Clustering datasets"[2] web site:

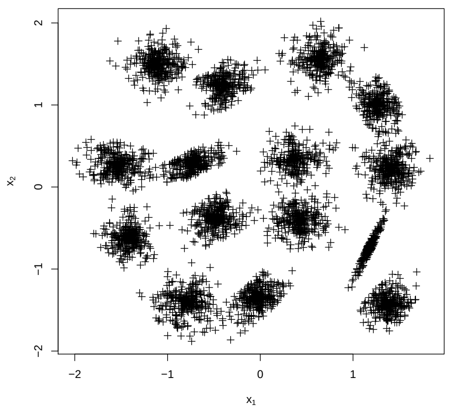
1. The "S-sets" are two-dimensional datasets with 15 clusters and different degrees of overlapping [14];
2. The "A-sets" are two-dimensional datasets with numbers of clusters ranging from 20 to 50 [18];
3. The "DIM-sets" are datasets with 16 clusters, with dimensionality ranging from 32 to 1064 [13].

We also report results with five datasets from the UCI Machine Learning Repository[3].

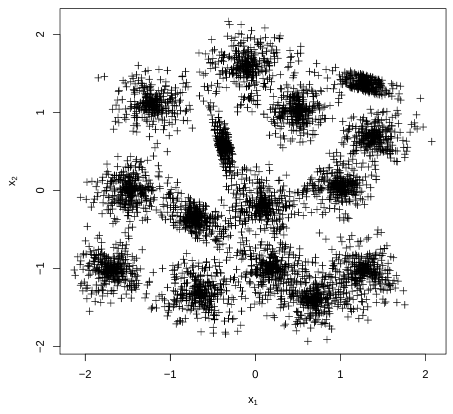
Experiments with the S-sets

We first compared our method to the c-means algorithm, the `pdfCluster` procedure and the EM algorithm with Gaussian mixtures and the BIC criterion, using the four S-sets shown in Figure 7. These are synthetic two-dimensional data with $n = 5000$ vectors and 15 Gaussian clusters with different degree of cluster overlapping.

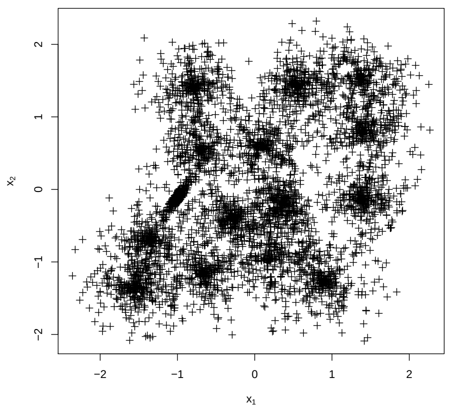
[2]`http://cs.joensuu.fi/sipu/datasets`
[3]`https://archive.ics.uci.edu/ml`



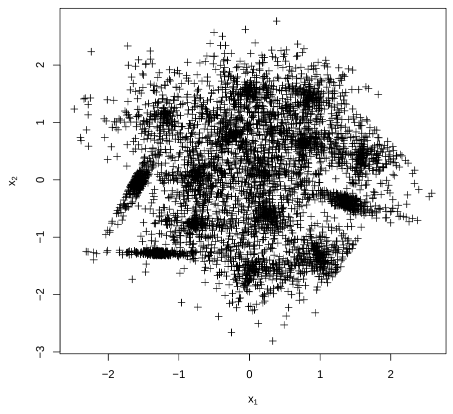
(a)



(b)



(c)



(d)

Figure 7: The S-sets: S1 (a), S2 (b), S3 (c) and S4 (d).

Our algorithm was initialized with a random partition in 500 clusters. The previous study (Section 4.1) suggested that a value of K of the order of two or there times $\sqrt{n}$ and q greater than 0.5 are suitable. We thus set $K = 200$ and $q = 0.9$. The K nearest neighbors were computed using function `get.knn` in the R package `FNN`. The algorithm was run ten times for each dataset. As before, the implementations of the c-means, `pdfCluster` and model-based clustering are from the R packages `stats`, `pdfCluster` and `mclust`, respectively. For the `Mclust` procedure, the number of clusters was specified to be in the range 10–20. The procedure was run without any restriction on the form of the clusters, and with the most constrained model (spherical cluster shape and equal volume).

The results are shown in Table 1. In this table, we report the number of clusters, the Adjusted Rand Index (ARI) and the computing time (in seconds) for the five methods. For E-K-NNclus, we give the median and interquartile range (difference between the third and first quartiles) over the 10 trials, for each of the three indices. The c-means algorithm performed well and was much faster than other methods, but it requires to fix the number of clusters in advance. The model-based algorithm failed to identify the correct number of clusters, except for the constrained model and the most separated cluster case (dataset S1). In contrast, The E-K-NNclus and `pdfCluster` methods found the correct number of clusters and yielded comparable values of the ARI, the former method being five to six times faster.

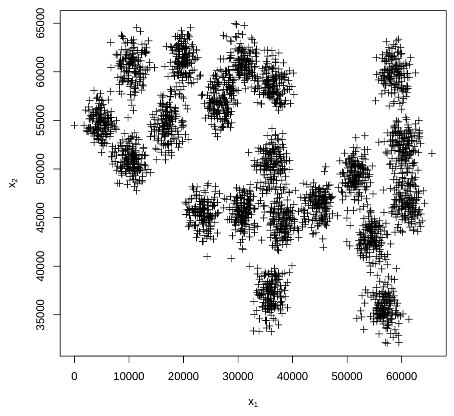
Experiments with the A-sets

The A-sets are three two-dimensional synthetic datasets with varying numbers of clusters. There are 150 vectors per cluster. Datasets A1, A2 and A3 have, respectively, 20, 35 and 50 clusters. The datasets are displayed in Figure 8.

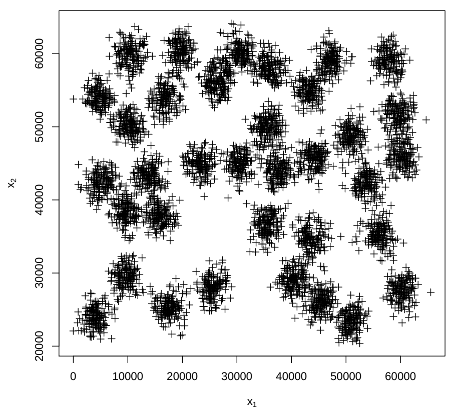
As, in this experiment, the emphasis is on the determination of clusters, we compared EK-NNclus with two competing methods to determine the number of clusters: `pdfCluster` and the model-based approach with the BIC criterion. Parameter q of the EK-NNclus algorithm was fixed to $q = 0.9$. The number of neighbors was fixed to $K = 150$ for dataset A1, and $K = 200$ for datasets A2 and A3 (i.e., consistently with the rule of thumb that K should be of the order of two to three times $\sqrt{n}$). Two initialization methods were used: $c_0 = n$ initial clusters, and $c_0 = 1000$ random initial clusters. As before, the EK-NNclus algorithm was run 10 times. For the `pdfCluster` procedure, the default settings were not suitable (the function

Table 1: Results on the S-sets.

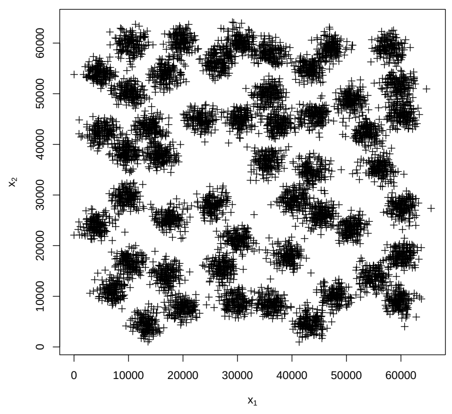
Dataset	Result	EK-NNclus	c-means	pdfCluster	model-based (constrained)	model-based
S1	c	15(0)	15 (fixed)	15	20	15
	ARI	0.99 (4.4e-4)	0.99	0.99	0.93	0.99
	time	7.70 (0.30)	0.60	52.16	52.84	25.54
S2	c	15 (0)	15	15	20	20
	ARI	0.94 (4.2e-4)	0.94	0.94	0.76	0.90
	time	8.26 (0.29)	0.72	51.36	63.46	31.2
S3	c	15 (0)	15	15	20	20.0
	ARI	0.73 (9.6e-4)	0.73	0.73	0.55	0.7
	time	10.77 (0.50)	1.06	57.52	65.51	31.5
S4	c	15 (0)	15	15	19	19
	ARI	0.64 (2.9e-3)	0.63	0.65	0.52	0.63
	time	10.95 (1.38)	1.59	63.36	79.35	31.55



(a)



(b)



(c)

Figure 8: The A-sets: A1 (a), A2 (b), and A3 (c).

issues a warning message). Following the advice of the authors [29], parameter `n.grid` was increased to 1500. For the model-based method `Mclust`, the number of clusters was varied between 3 and $c^* + 5$, where c^* is the true number of clusters.

The results are shown in Table 2. In this table, we report the number of clusters and the computing time (in seconds) for the five methods. As before, for E-K-NNclus, we give the median and interquartile range (difference between the third and first quartiles) over the 10 trials, for both the number of clusters and the computing times. As we can see, the model-based method (with or without restrictions on cluster shape) overestimates the number of clusters, except for the A3 dataset. The `pdfCluster` method is the most time-consuming method, and it severely underestimates the number of clusters. In contrast, the EK-NNclus algorithm estimates the correct number of clusters quite accurately for all three datasets. Initializing the algorithm with as many clusters as data points yields slightly better results in terms of final number of clusters (with less variability), but it is time-consuming for large datasets. Randomly initializing the algorithm with a smaller number of clusters considerably shortens the computing time, at the cost of a slight underestimation of the number of clusters. This is due to the fact that two or more real clusters can be grouped as one cluster, as shown in Figure 9. This issue could be easily detected and fixed by inspecting within-cluster distances.

Experiments with the DIM-sets

As a third experiment, we compared our method to the c-means, pdfCluster and model-based methods on three of the DIM datasets. These are high-dimensional data sets $n = 1024$ and 16 Gaussian clusters. We used the files dim256, dim512 and dim1024 with, respectively, 256, 512 and 1024 dimensions.

Parameters q and K of the EK-NNclus algorithm were fixed to $q = 0.9$ and $K = 50$. The algorithm was initialized with $c_0 = n$ clusters and was run 10 times. The c-means algorithm was run 100 times with $c = 16$ clusters and the result with the best value of the objective function was kept. As the `pdfCluster` procedure cannot be used in high dimensions, we performed a PCA of the data and used the first two principal components, with parameter `n.grid` set to 1000. For the model-based method `Mclust`, the constrained model (spherical cluster shape and equal volume) was assumed and the number of clusters was varied from 3 to 20.

The results are shown in Table 3. We can see that the EK-NNclus

Table 2: Results on the A-sets. For the EK-NNclus algorithm, the results are given with two different initialization methods: $c_0 = n$ initial clusters, and $c_0 = 1000$ random initial clusters.

Dataset	Result	EK-NNclus	EK-NNclus	`pdfCluster`	model-based	model-based
		($c_0 = n$)	($c_0 = 1000$)			(constrained)
A1	c	20 (0)	20 (0)	17	24	24
$n = 3000$	time	32.9 (3.14)	9.8 (0.2)	84.5	31.8	7.88
A2	c	35 (0)	34 (1)	26	39	39
$n = 5250$	time	193 (9.81)	23.8 (0.6)	298	138	36.2
A3	c	49 (1)	49 (2.5)	34	50	51
$n = 7500$	time	358 (8.23)	35.1 (1.09)	629	412	99.4

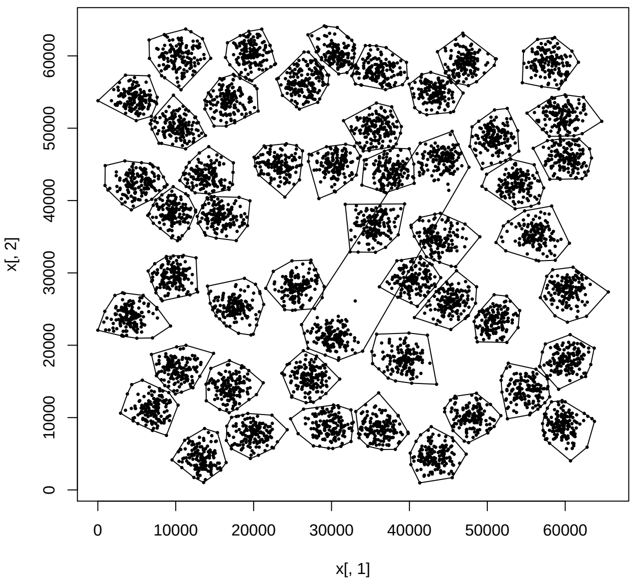


Figure 9: Example of underestimation of the number of cluster using the EK-NNclus algorithm with $c_0 < n$ initial clusters: two real clusters are merged into one cluster.

Table 3: Results on the DIM-sets.

Dataset	Result	EK-NNclus	c-means	`pdfCluster`	model-based (constrained)
dim256	c	16 (0)	16 (fixed)	5	16
	ARI	1.0 (0)	0.94	0.23	1
	time	1.4 (0.058)	2.76	11.30	116
dim512	c	16 (0)	16(fixed)	9	16
	ARI	1 (0)	0.94	0.5	1
	time	1.4 (0.11)	13.27	10.9	467
dim1024	c	16 (0)	16 (fixed)	8	18
	ARI	1 (0)	0.94	0.28	0.998
	time	1.4 (0.14)	36.38	11.13	23

algorithm always recovers the true 16-cluster partition perfectly and is the fastest of all methods. In contrast, the c-means algorithm provided with the correct number of clusters does not recover the true partition exactly. The model-based method also recovers the true partition, except for the dim1024 dataset, but it takes more computing time. As with other datasets, the pdfCluster severely underestimates the number of clusters. Overall, the EK-NNclus algorithm outperforms the other methods on these datasets.

Experiments with real datasets

As a final comparative experiment in this section, we considered five classical benchmark datasets from the UCI repository of databases, whose characteristics are summarized in Table 4. As in [26], we only considered in the dataset Ecoli the classes 'im', 'pp' and 'imU', which are close and hard to separate. As before, the EK-NNclus algorithm was compared to the c-means algorithm (with the correct number of clusters), the `pdfCluster` algorithm, and the model-based method with and without contraints on the covariance matrices. For the model-based methods, the number of clusters was searched between 2 and 10. The `pdfCluster` algorithm was not applied to the Heart dataset, because it contains discrete attributes that cannot be handled by this procedure. Our algorithm was run 100 times with the following values of the coefficients: $K = 40$, $q = 0.9$, except for the Iris dataset, which has fewer instances, and for which we set $K = 30$.

Table 4: Characteristics of the real datasets.

Name	Clusters	Attributes	Instances
Wine	3	13	178
Seeds	3	7	201
Ecoli	3	7	164
Iris	3	4	150
Statlog (Heart)	2	13	270

The results are shown in Table 5. We can see that the EK-NNclus algorithm correctly identified the number of clusters in all five datasets, while the other methods for determining the number of clusters generally performed poorly. The quality of the obtained partition, as measured by the ARI criterion, is usually close to that of the c-means partition. However, the c-means algorithm was provided with the extra knowledge of the correct number of partitions. We can also note that EK-NNclus was significantly faster than `pdfCluster` and the model-based approach without constraints. Overall, EK-NNclus thus seems to provide a valuable alternative to alternative techniques in terms of both accuracy and computing time.

5 Conclusions

A new credal clustering method based on the EK-NN rule has been proposed. Starting from an initial partition with as many clusters as objects in the dataset, the method, called EK-NNclus, iteratively applies the EK-NN rule to change the class labels, until a stable partition has been obtained.

The algorithm can be implemented in a competitive Hopfield neural network model with as many neurons as objects in the database, neuron states corresponding to clusters. The energy function of this network, which is minimized by the algorithm, is related to the plausibility of the partition: the algorithm can thus be seen as searching from the most plausible partition of the dataset. The end result produced by the algorithm is a credal partition, in which the cluster membership of each object is described by a mass function assigning a mass to each cluster, as well as to the set of all clusters. The mass assigned to the set of clusters can be used to identify outliers in the dataset.

The method has been compared to standard clustering algorithms on several datasets. The experiments have shown that the method generally

Table 5: Results on the real datasets.

Dataset	Result	EK-NNclus	c-means	pdfCluster	model-based	model-based (constrained)
Wine	c	3 (0)	3 (fixed)	10	3	4
	ARI	0.86 (0.034)	0.90	0.47	0.93	0.78
	time	0.016 (4.0e-3)	0.018	0.95	6.97	0.040
Seeds	c	3 (0)	3 (fixed)	2	4	10
	ARI	0.74 (0.048)	0.77	0.46	0.57	0.32
	time	0.026 (6e-3)	0.019	1.068	5.88	0.046
Ecoli	c	3 (0)	3 (fixed)	2	6	7
	ARI	0.75 (0.11)	0.72	0.49	0.65	0.62
	time	0.050 (0.011)	0.022	0.78	2.24	0.11
Iris	c	3 (1)	3 (fixed)	2	2	10
	ARI	0.51 (0.13)	0.62	0.57	0.57	0.32
	time	0.015 (5e-3)	0.016	0.16	1.093	0.030
Heart	c	2 (0.25)	2 (fixed)	NA	5	5
	ARI	0.41 (0.12)	0.45	NA	0.11	0.21
	time	0.043 (7e-3)	0.023	NA	0.75	0.089

performs better than density-based and model-based clustering procedures, especially when it comes to determining the number of clusters. It is also faster than the non-parameteric density-based approach, and it performs much better with high-dimensional data.

As compared to other credal clustering algorithms such as EVCLUS or CEM, the method generates simpler credal partitions (with only $c+1$ masses for each object, c being the number of clusters). It is thus better suited for clustering large datasets. As the EK-NNclus is based on distances, it can be applied to any proximity data, and it can be kernelized to handle data with complex cluster shapes. These research directions are currently being investigated.

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] A. Azzalini and N. Torelli. Clustering via nonparametric density estimation. *Statistics and Computing*, 17:71–80, 2007.

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

[3] A.-S. Capelle, O. Colot, and C. Fernandez-Maloigne. Evidential segmentation scheme of multi-echo MR images for the detection of brain tumors using neighborhood information. *Information Fusion*, 5(3):203–216, 2004.

[4] G. Celeux and G. Govaert. A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics & Data Analysis*, 14(3):315–332, 1992.

[5] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.

[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 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. Denœux and P. Smets. Classification using belief functions: the relationship between the case-based and model-based approaches. *IEEE Transactions on Systems, Man and Cybernetics B*, 36(6):1395–1406, 2006.

[9] D. Dubois and H. Prade. Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4:244–264, 1988.

[10] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley and Sons, New-York, 2001.

[11] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.

[12] C. Fraley, A. E. Raftery, T. B. Murphy, and L. Scrucca. mclust version 4 for R: Normal mixture modeling for model-based clustering, classification, and density estimation. Technical Report 597, Department of Statistics, University of Washington, 2012.

[13] P. Fränti, O. Virmajoki, , and V. Hautamäki. Fast agglomerative clustering using a k-nearest neighbor graph. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(11):1875–1881, 2006.

[14] P. Fränti and O. Virmajoki. Iterative shrinking method for clustering problems. *Pattern Recognition*, 39(5):761–775, 2006.

[15] G. Galán-Marín and J. Muñoz Pérez. Design and analysis of maximum Hopfield networks. *IEEE Transactions on Neural Networks*, 12(2):329–339, Mar 2001.

[16] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79:2554–2558, 1982.

[17] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–ñ218, 1985.

[18] I. Kärkkäinen and P. Fränti. Dynamic local search algorithm for the clustering problem. Technical Report A-2002-6, Department of Computer Science, University of Joensuu, 2002.

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

[20] 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.

[21] 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.

[22] 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.

[23] Z.-G. Liu, Q. Pan, and J. Dezert. Evidential classifier for imprecise data based on belief functions. *Knowledge-Based Systems*, 52(0):246 –257, 2013.

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

[25] Z.-G. Liu, Q. Pan, and J. Dezert. Classification of uncertain and imprecise data based on evidence theory. *Neurocomputing*, 133:459 – 470, 2014.

[26] 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.

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

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

[29] G. Menardi and A. Azzalini. Clustering via nonparametric density estimation: The R package pdfcluster. *Journal of Statistical Software*, 57(11), 2014.

[30] N. K. Pal and S. Gosh. Some classification algorithms integrating Dempster-Shafer theory of evidence with the rank nearest neighbor rule. *IEEE Trans. on Systems, Man and Cybernetics – Part A*, 31(1):59–66, 2001.

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

[32] H. Shen and K.-C. Chou. Predicting protein subnuclear location with optimized evidence-theoretic K-nearest classifier and pseudo amino acid composition. *Biochemical and Biophysical Research Communications*, 337(3):752–756, 2005.

[33] H. Shen and K.-C. Chou. Using optimized evidence-theoretic K-nearest neighbor classifier and pseudo-amino acid composition to predict membrane protein types. *Biochemical and Biophysical Research Communications*, 334(1):288–292, 2005.

[34] Z.-G. Su and P.-H. Wang. Improved adaptive evidential k-NN rule and its application for monitoring level of coal powder filling in ball mill. *Journal of Process Control*, 19(10):1751–1762, 2009.

[35] Y. Takefuji, K.-C. Lee, and H. Also. An artificial maximum neural network: a winner-take-all neuron model forcing the state of the system in a solution domain. *Biological Cybernetics*, 67(3):243–251, 1992.

[36] Z.-B. Xu, G.-Q. Hu, and C.-P. Kwong. Asymmetric Hopfield-type networks: Theory and applications. *Neural Networks*, 9(3):483–501, 1996.

[37] B.-S. Yang and K. J. Kim. Application of Dempster-Shafer theory in fault diagnosis of induction motors using vibration and current signals. *Mechanical Systems and Signal Processing*, 20:403–420, 2006.

[38] N. M. Zaki, S. Deris, S. N. V. Arjunan, and R. M. Illias. Assignment of protein sequence to functional family using neural network and Dempster-Shafer theory. *Journal of Theoretics*, 5(1), 2003.

[39] W. Zhong, G. Altun, X. Tian, R. Harrison, P. Tai, and Y. Pan. Parallel protein secondary structure prediction based on neural networks.

In *Proceedings of the 26th Annual International Conference of the Engineering in Medicine and Biology Society, 2004 (EMBC 2004)*, pages 2968 – 2971. IEEE, 2004.

[40] 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.

[41] H. Zhu and O. Basir. An adaptive fuzzy evidential nearest neighbor formulation for classifying remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 43(8):1874–1888, 2005.

[42] L. M. Zouhal and T. Denœux. An evidence-theoretic k-NN rule with parameter optimization. *IEEE Trans. on Systems, Man and Cybernetics C*, 28(2):263–271, 1998.