Noname manuscript No.
(will be inserted by the editor)

CEVCLUS : Evidential clustering with instance-level constraints for relational data

V. Antoine · B. Quost · M.-H. Masson · T. Denoeux

Received: date / Accepted: date

Abstract Recent advances in clustering consider incorporating background knowledge in the partitioning algorithm, using, e.g., pairwise constraints between objects. As a matter of fact, prior information, when available, often makes it possible to better retrieve meaningful clusters in data. Here, this approach is investigated in the framework of belief functions, which allows us to handle the imprecision and the uncertainty of the clustering process. In this context, the EVCLUS algorithm was proposed for partitioning objects described by a dissimilarity matrix. It is extended here so as to take pairwise constraints into account, by adding a term to its objective function. This term corresponds to a penalty term that expresses pairwise constraints in the belief function framework. Various synthetic and real datasets are considered to demonstrate the interest of the proposed method, called CEVCLUS, and two applications are presented. The performances of CEVCLUS are also compared to those of other constrained clustering algorithms.

V. Antoine*
Université Blaise Pascal, PRES Clermont Université
LIMOS, UMR CNRS 6158
BP 10125, F-630000 Clermont-Ferrand, France
E-mail: violaine.antoine@univ-bpclermont.fr
Tel.: +33-473-40-52-13

B. Quost · T. Denoeux
Université de Technologie de Compiègne
Laboratoire Heudiasyc, UMR CNRS 7253
Compiègne, France

M.-H. Masson
Université de Picardie Jules Verne, IUT de l'Oise
Laboratoire Heudiasyc, UMR CNRS 7253
Compiègne, France

*This work has been mostly developed while the author was with Heudiasyc.

Keywords belief functions · evidence theory · Dempster-Shafer theory · relational data · pairwise constraints · constrained clustering

1 Introduction

Clustering is a well-known issue in pattern recognition and data mining. This problem consists in grouping objects with similar characteristics into clusters. Data are generally described either by numerical attributes (also called features) or directly by pairwise dissimilarities. In the latter case, they are referred to as relational data or more specifically dissimilarity data. Relational clustering methods are considered to be more general than clustering algorithms handling feature vectors, since the latter can always be transformed into dissimilarity data.

There exists a wide variety of clustering methods for attribute and relational data. These methods may be divided into two main families. The first one consists of hierarchical methods. Such methods provide an organization of the objects into a sequence of nested groups [9,20]. The second family encompasses clustering algorithms generating a partition. In this framework, fuzzy partitions differ from crisp partitions as they allow us to represent the uncertainty regarding the class membership of an object. A popular algorithm deriving a fuzzy partition from attribute data is the fuzzy c-means algorithm (FCM) [3]. Many variants of this algorithm have been developed [21,15]. In particular, a relational version, called Relational FCM, was introduced in [14].

Recently, a new concept of partition, the credal partition, has been proposed. Developed in the framework of belief function theory, it extends the concepts of crisp and fuzzy partition and makes it possible to represent both uncertainty and imprecision regarding the class membership of an object. Several evidential clustering methods generating credal partitions have been proposed [8,23–25].

Traditionally, clustering methods proceed by exploiting the input data only. However, in some situations, prior information may be available from domain knowledge. This background information can be translated into constraints at different levels such as the model [38,11], cluster [5] or instance level [27,28]. Incorporating such constraints into unsupervised clustering algorithms has recently become a topic of great interest as it helps extract correct groups [22, 36]. Here, we focus on two types of instance-level constraints proposed in [35]: Must-Link constraints, which specify that two objects should be in the same class and Cannot-Link constraints indicating that two objects should be in different clusters. This type of supervision constraints has been introduced in several attribute-based clustering algorithms producing hard [35], fuzzy [2,12,13] or credal partitions [1]. However, comparatively fewer methods have been developed for relational-based clustering [16,10] and none of them generates a credal partition.

In this paper, we propose to combine the concepts of relational and evidential clustering with that of constrained clustering in order to introduce a new method, called CEVCLUS, which takes advantage of background knowledge and generates a credal partition from dissimilarity data. The new algorithm expresses pairwise constraints in the framework of belief functions and integrates them in the evidential clustering algorithm EVCLUS [8].

The remainder of the paper is organized as follows. In Section 2, we outline the basics of belief functions theory and the notion of credal partition; we then present the EVCLUS algorithm. The CEVCLUS algorithm is introduced in Section 3 and applied to various synthetic and real datasets in Section 4. Finally, Section 5 presents some conclusions and directions for further research.

2 Background

2.1 Belief functions theory

The Dempster-Shafer theory of belief functions [7,31] is a mathematical framework for representing and reasoning with uncertain and imprecise knowledge.

Let $\Omega = \{\omega_1, \dots, \omega_c\}$ be a finite set of elements called the frame of discernment, or frame for short, and y a variable defined on Ω. A mass function m^Ω is a mapping from the power set of Ω to $[0,1]$ representing partial knowledge about the actual value taken by y and verifying the following constraint:

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

Each subset A of Ω such that $m(A) > 0$ is called focal set of m. The mass $m(A)$ represents the quantity of belief committed to A that cannot be assigned to any more specific subset due to lack of knowledge. When all the focal sets are singletons, m is said to be Bayesian: it corresponds to a probability distribution. The mass function m is said categorical if it has a single focal set. In particular, complete ignorance corresponds to the vacuous mass function such that $m(\Omega) = 1$, whereas full certainty about the value of y is represented by a certain mass function such that $m(\{\omega\}) = 1$ for some $\omega \in \Omega$.

A mass function m such that $m(\emptyset) = 0$ is said to be normalized. The normalization constraint, originally assumed by Shafer [31], can be relaxed in the so-called *open-world* assumption [32]. In this case, the mass function $m(\emptyset) > 0$ is interpreted as a quantity of belief given to the hypothesis that y might not belong to Ω.

The partial knowledge expressed by a mass function m can be equivalently represented by the corresponding belief function $bel : 2^\Omega \to [0,1]$ or the plausibility function $pl : 2^\Omega \to [0,1]$ defined, respectively, as

$$bel(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B) \quad \forall A \subseteq \Omega. \tag{2}$$

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B) \quad \forall A \subseteq \Omega, \tag{3}$$

These functions correspond to two facets of the same information and one can be retrieved from the other by:

$$pl(A) = 1 - m(\emptyset) - bel(\overline{A}), \tag{4}$$

where $\overline{A}$ denotes the complement of $A \subseteq \Omega$. The quantity $bel(A)$ measures the total support given to A, whereas $pl(A)$ is interpreted as the degree to which the evidence fails to support the complement of A.

The conjunctive rule makes it possible to aggregate two distinct mass functions m_1 and m_2 defined on the same frame, in order to get a new mass function containing all the information of m_1 and m_2:

$$(m_1 \cap\!\!\!\!\bigcirc\, m_2)(A) = \sum_{B \cap C = A} m_1(B) m_2(C) \quad \forall A \subseteq \Omega. \tag{5}$$

The quantity $K_{12} = (m_1 \cap\!\!\!\!\bigcirc\, m_2)(\emptyset)$ represents the degree of conflict between m_1 and m_2, i.e., the degree of disagreement between the information sources.

The concepts of marginalization and vacuous extension allow us to manipulate mass functions defined on different frames. Let $m^{\Omega \times \Theta}$ be a mass function defined on the Cartesian product $\Omega \times \Theta$. The corresponding marginal mass function on Ω is:

$$m^{(\Omega \times \Theta)\downarrow\Omega}(A) = \sum_{B \subseteq \Omega \times \Theta, B^{\downarrow\Omega} = A} m^{\Omega \times \Theta}(B) \qquad \forall A \subseteq \Omega, \tag{6}$$

where $B^{\downarrow\Omega}$ denotes the projection of B onto Ω, i.e., $B^{\downarrow\Omega} = \{\omega \in \Omega / \exists\theta \in \Theta,\ (\omega,\theta) \in B\}$.

Conversely, the vacuous extension [31] extends a mass function m^{Ω} to $m^{\Omega\times\Theta}$. It is defined as follows:

$$m^{\Omega\uparrow(\Omega\times\Theta)}(B) = \begin{cases} m^{\Omega}(A) & \text{if } B = A\times\Theta, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

For decision making, it is possible to transform the mass function into a pignistic probability distribution [33]:

$$BetP(\omega) = \sum_{A\subseteq\Omega|\omega\in A} \frac{m(A)}{|A|} \quad \forall\omega\in\Omega, \tag{8}$$

where $|A|$ denotes the cardinality of $A \subseteq \Omega$. This pignistic transformation evenly distributes the mass assigned to each subset A among the elements of A. For unnormalized mass functions, a preliminary normalization step has to be performed. This can be carried out by different methods. For example, Dempster's normalization consists in dividing all the masses by $1-m(\emptyset)$.

To quantify the degree of information given by a mass function, several measures have been proposed [26]. The most popular is the non-specificity measure, which evaluates the degree of imprecision of a mass function m [17]. It is defined as

$$N(m) = \sum_{A\subseteq\Omega\setminus\emptyset} m(A)\log_2|A| + m(\emptyset)\log_2|\Omega|. \tag{9}$$

2.2 Credal partitions

Let $O = \{o_1,\dots,o_n\}$ be a collection of n objects to be classified into c clusters and let $\Omega = \{\omega_1,\dots,\omega_c\}$ be the set of clusters. A mass function m_i on Ω can be used in order to represent partial knowledge regarding the class membership of object i. The n-tuple $M = (m_1,\dots,m_n)$ of mass functions related to all objects is called a credal partition. This concept models a wide variety of situations ranging from complete ignorance to full certainty.

As an example, let us consider a collection of five objects that need to be classified into two classes. A credal partition is presented in Table 1. The class of the first object is known with certainty, whereas the class of the fourth object is totally unknown. The mass function m_2 is Bayesian. The fifth object, with the whole unit mass allocated to the empty set, can be considered as an outlier.

We may remark that, when each m_i is certain, the credal partition boils down to a hard partition. When each m_i is Bayesian, M corresponds to a fuzzy partition of Ω.

As underlined in [24,25], a credal partition conveys a large amount of information but it may be useful to summarize it to help the user interpreting the results. The most common operation consists in converting the credal partition into a fuzzy one using the pignistic transformation (8). Various classical tools can then be applied. Another strategy is to assign each object to the subset of classes with the highest mass. The resulting clustering structure, called a hard credal partition, contains at most 2^c groups and allow us to detect ambiguous objects.

Table 1 Example of credal partition

A	$m_1(A)$	$m_2(A)$	$m_3(A)$	$m_4(A)$	$m_5(A)$
$\emptyset$	0	0	0.1	0	1
$\{\omega_1\}$	1	0.9	0	0	0
$\{\omega_2\}$	0	0.1	0.8	0	0
Ω	0	0	0.1	1	0

2.3 The EVCLUS algorithm

The EVCLUS evidential clustering algorithm was the first method proposed to derive a credal partition from dissimilarity data [8]. Let us assume that the available data consist of a $n\times n$ dissimilarity matrix $\Delta = (\delta_{ij})$ such that the diagonal elements are zero and $\delta_{ij} = \delta_{ji}$ represents the dissimilarity between two objects o_i and o_j. As explained in [8], an evidential partition can be derived from the input dissimilarities by reasoning as follows.

Let us consider two objects o_i and o_j for which mass functions m_i and m_j are already known. In the Cartesian product $\Omega^2 = \Omega\times\Omega$, the belief regarding the joint class membership of both objects may be expressed by a mass function denoted $m_{i\times j}$. This mass function is obtained by combining the vacuous extensions of m_i and m_j [33]; it is defined as follows:

$$m_{i\times j}(A\times B) = m_i(A)\, m_j(B) \quad A,B\subseteq\Omega, A\neq\emptyset, B\neq\emptyset, \tag{10a}$$

$$m_{i\times j}(\emptyset) = m_i(\emptyset) + m_j(\emptyset) - m_i(\emptyset)\, m_j(\emptyset). \tag{10b}$$

In the Cartesian product Ω^2, let us denote by

$$\theta_{ij} = \{(\omega_1,\omega_1),(\omega_2,\omega_2),\dots,(\omega_c,\omega_c)\}$$

the event "Objects o_i and o_j belong to the same class". The plausibility $pl_{i\times j}(\theta_{ij})$ can be determined from $m_{i\times j}$:

$$pl_{i\times j}(\theta_{ij}) = \sum_{A\cap B\neq\emptyset} m_i(A)\, m_j(B), \tag{11}$$

Let K_{ij} be the degree of conflict between the mass functions m_i and m_j corresponding to objects o_i and o_j. The following equality holds:

$$pl_{i\times j}(\theta_{ij}) = 1 - K_{ij}. \tag{12}$$

Thus, it seems reasonable to require that the more dissimilar the objects, the less plausible it is that they belong to the same class and the higher the degree of conflict between their mass functions.

To obtain a credal partition from Δ, EVCLUS minimizes an error function inspired from multidimensional scaling (MDS) methods. In particular, the following error function is close to Sammon's stress function [30]:

$$J_{EVCLUS}(M,a,b) = \frac{1}{C}\sum_{i<j}\frac{(aK_{ij}+b-\delta_{ij})^2}{\delta_{ij}}, \tag{13}$$

where a and b are two coefficients, and C is a normalizing constant defined as

$$C = \sum_{i<j}\delta_{ij}. \tag{14}$$

In addition, all the mass functions in M must be positive and sum to unity. In order to avoid using a constrained optimization algorithm, we may use the following parameterization:

$$m_i(A_k) = \frac{\exp(\alpha_{ik})}{\sum_{l=1}^{2^c}\exp(\alpha_{il})}, \tag{15}$$

where $A_k, k\in\{1,\dots 2^c\}$ are the focal sets and the α_{ik} are the $(n\times 2^c)$ real parameters representing the credal partition. In this case, the positivity constraints are implicitly satisfied. Criterion (13) can be minimized iteratively with respect to α_{ik}, a and b, using a gradient-based procedure. The partial derivatives of J_{EVCLUS} with the respect to all parameters have the following expressions:

$$\frac{\partial J_{EVCLUS}}{\partial a} = \frac{2}{\sum_{i<j}\delta_{ij}}\sum_{i=1}^{n}\sum_{j=i+1}^{n}\frac{K_{ij}(aK_{ij}+b-\delta_{ij})}{\delta_{ij}}, \tag{16a}$$

$$\frac{\partial J_{EVCLUS}}{\partial b} = \frac{2}{\sum_{i<j}\delta_{ij}}\sum_{i=1}^{n}\sum_{j=i+1}^{n}\frac{(aK_{ij}+b-\delta_{ij})}{\delta_{ij}}, \tag{16b}$$

$$\frac{\partial J_{EVCLUS}}{\partial \alpha_{il}} = \frac{2a}{\sum_{i<j}\delta_{ij}}\sum_{j=i+1}^{n}\frac{(aK_{ij}+b-\delta_{ij})}{\delta_{ij}}\frac{\partial K_{ij}}{\partial \alpha_{il}}, \tag{16c}$$

$$\frac{\partial K_{ij}}{\partial \alpha_{il}} = \sum_{k,k'}\frac{\partial m_{ik}}{\partial \alpha_{il}} m_{jk'}\xi_{kk'}, \tag{16d}$$

with $\xi_{kk'}=1$ if $A_k\cap A'_k=\emptyset$ and $\xi_{kk'}=0$ otherwise; $m_{ik}=m_i(A_k)$ and

$$\frac{\partial m_{ik}}{\partial \alpha_{il}} = \begin{cases} m_{ik}(1-m_{ik}) & \text{if } l=k,\\ -m_{ik}m_{il} & \text{otherwise.}\end{cases} \tag{16e}$$

3 A constrained EVCLUS algorithm

3.1 Expression of the constraints

As explained in Section 2.3, the mass function $m_{i\times j}$ (10a) expresses the belief regarding the joint class membership of two objects o_i and o_j. Thus, it can be used to translate Must-Link and Cannot-Link constraints in the framework of belief functions. More specifically, we can employ the plausibility $pl_{i\times j}(\theta_{ij})$ (11) and the plausibility $pl_{i\times j}(\overline{\theta_{ij}})$ defined in Equation (17) below as θ_{ij} represents the event "Objects o_i and o_j belong to the same class" and $\overline{\theta_{ij}}$ the complementary event "Objects o_i and o_j do not belong to the same class":

Table 2 Plausibilities for the events θ_{ij} and $\overline{\theta_{ij}}$

F	$pl_{1\times 2}(F)$	$pl_{1\times 3}(F)$	$pl_{1\times 4}(F)$	$pl_{1\times 5}(F)$
θ_{ij}	0.9	0.1	1	0
$\overline{\theta_{ij}}$	0.1	0.9	1	0

$$pl_{i\times j}(\overline{\theta_{ij}}) = 1 - m_{i\times j}(\emptyset) - bel_{i\times j}(\theta_{ij}), \tag{17a}$$

$$= 1 - m_{i\times j}(\emptyset) - \sum_{k=1}^{c} m_i(\{\omega_k\})\, m_j(\{\omega_k\}). \tag{17b}$$

To illustrate this point, let us consider an example. From the credal partition shown in Table 1, it is possible to compute the joint class plausibilities expressed in Table 2. In particular, we observe a low plausibility $pl_{1\times 2}(\theta_{12})$ and a high plausibility $pl_{1\times 2}(\overline{\theta_{12}})$, and the converse for $pl_{1\times 3}$. This reflects the knowledge derived from Table 1 regarding the joint class membership for o_1, o_2 and o_3. The credal partition contains no information concerning the class of the fourth object: its relationship with the object o_1 is then unknown. In that case, the values of the joint class plausibilities on both events θ_{14} and $\overline{\theta_{14}}$ are high. Conversely, an outlier like object o_5 gives low values for $pl_{1\times 5}(\theta_{15})$ and $pl_{1\times 5}(\overline{\theta_{15}})$.

To sum up, the relationship between two objects can be deduced from the plausibilities $pl_{i\times j}(\theta_{ij})$ and $pl_{i\times j}(\overline{\theta_{ij}})$. In particular, two objects o_i and o_j are surely in the same class if $pl_{i\times j}(\overline{\theta_{ij}})=0$ and $pl_{i\times j}(\theta_{ij})=1$. Conversely, two objects o_i and o_j are surely in different classes if $pl_{i\times j}(\theta_{ij})=0$ and $pl_{i\times j}(\overline{\theta_{ij}})=1$.

3.2 Objective function of CEVCLUS

In an evidential clustering algorithm, the credal partition is unknown and needs to be learnt from data. However, some background knowledge may be available in the form of Must-Link and Cannot-Link constraints. Both types of constraints can be simply formulated with joint class plausibilities. As explained before, a Must-Link constraint implies a low value for $pl_{i\times j}(\overline{\theta_{ij}})$ and a high value for $pl_{i\times j}(\theta_{ij})$. Conversely, a Cannot-Link constraint requires $pl_{i\times j}(\theta_{ij})$ to be low and $pl_{i\times j}(\overline{\theta_{ij}})$ to be high.

A penalty term representing the cost of violating pairwise constraints can then be formulated as follows:

$$J_{CONST} = \frac{1}{2(|\mathscr{M}|+|\mathscr{C}|)}\left(J_{\mathscr{M}}+J_{\mathscr{C}}\right), \tag{18a}$$

$$J_{\mathscr{M}} = \sum_{(o_i,o_j)\in\mathscr{M}} pl_{i\times j}(\overline{\theta_{ij}}) + 1 - pl_{i\times j}(\theta_{ij}), \tag{18b}$$

$$J_{\mathscr{C}} = \sum_{(o_i,o_j)\in\mathscr{C}} pl_{i\times j}(\theta_{ij}) + 1 - pl_{i\times j}(\overline{\theta_{ij}}), \tag{18c}$$

where $\mathscr{M}$ and $\mathscr{C}$ correspond, respectively, to the sets of Must-Link and Cannot-Link constraints and $|\mathscr{M}|$ and $|\mathscr{C}|$ denote the cardinalities of these sets.

We propose to add the penalty term J_{CONST} to the objective function J_{EVCLUS} (13). The minimization of the new criterion $J_{CEVCLUS}$ will allow us to obtain a credal partition that, on the one hand, is compatible with the input dissimilarity matrix and, on the other hand, respects the instance-level constraints. This criterion will be defined as follows:

$$J_{CEVCLUS} = J_{EVCLUS} + \xi J_{CONST}, \tag{19}$$

where $\xi \geq 0$ is a hyperparameter that controls the trade-off between the fit to the distance data and the constraints.

For a credal partition $M = (m_1, \ldots, m_n)$, the computation of the objective function $J_{CEVCLUS}$ can be summarized by the following algorithm:

Algorithm 1 Computation of $J_{CEVCLUS}$.

Input: $M, a, b, \xi, \Delta, \mathscr{M}, \mathscr{C}$
Output: $J_{CEVCLUS}$
  for $i = 1$ to n **do**
    for $j = i+1$ to n **do**
      Compute the degree of conflict K_{ij} between m_i and m_j
      Compute $pl_{i\times j}(\theta_{ij})$ using (12)
      Compute $pl_{i\times j}(\overline{\theta_{ij}})$ using (17)
    end for
  end for
  Compute J_{EVCLUS} using (13)
  Compute J_{CONST} using (18)
  $J_{CEVCLUS} = J_{EVCLUS} + \xi J_{CONST}$.

3.3 Optimization

As in EVCLUS, we use the parametrization (15) for the mass functions. Any classical gradient-based procedure may then be employed to minimize iteratively the objective function (19) with respect to the learning parameters α_{il}, a and b. The particular optimization algorithm[1] used in the experiments reported in the next section is described in Appendix A.

We can remark that the penalty term J_{CONST} does not depend on the coefficients a and b. Consequently, the partial derivatives of $J_{CEVCLUS}$ with respect to a and b are similar to the ones computed for J_{EVCLUS} and are given by Equations (16a) and (16b), respectively.

In contrast, the mass functions expressed using the α_{il} (where i is the index of an object and A_l a subset of Ω) appear in J_{CONST}. We must then compute the partial derivatives of the objective function with respect to α_{il}:

$$\frac{\partial J_{CEVCLUS}}{\partial \alpha_{il}} = \frac{\partial J_{EVCLUS}}{\partial \alpha_{il}} + \xi \frac{1}{2(|\mathscr{M}|+|\mathscr{C}|)} \frac{\partial J_{CONST}}{\partial \alpha_{il}}. \tag{20}$$

The first term of this derivative, concerning J_{EVCLUS}, is given in Equation (16c). The second term is computed as follows:

$$\frac{\partial J_{CONST}}{\partial \alpha_{il}} = \sum_{(o_i,o_j)\in\mathscr{M}} \left(\frac{\partial pl(\overline{\theta_{ij}})}{\alpha_{il}} - \frac{\partial pl(\theta_{ij})}{\alpha_{il}} \right) + \sum_{(o_i,o_j)\in\mathscr{C}} \left(\frac{\partial pl(\theta_{ij})}{\alpha_{il}} - \frac{\partial pl(\overline{\theta_{ij}})}{\alpha_{il}} \right), \tag{21a}$$

$$\frac{\partial pl(\overline{\theta_{ij}})}{\alpha_{il}} = -\frac{\partial m_i(\emptyset)}{\partial \alpha_{il}} + \frac{\partial m_i(\emptyset)}{\partial \alpha_{il}} m_j(\emptyset) - \sum_{A_k,|A_k|=1} \frac{\partial m_{ik}}{\partial \alpha_{il}} m_{jk}, \tag{21b}$$

$$\frac{\partial pl(\theta_{ij})}{\alpha_{il}} = \sum_{A_k\cap A_{k'}\neq\emptyset} m_{jk'} \frac{\partial m_{ik}}{\partial \alpha_{il}}. \tag{21c}$$

The computation of the gradient of $J_{CEVCLUS}$ can be summarized as follows:

Algorithm 2 Computation of the gradient of $J_{CEVCLUS}$.

Input: $\{\alpha_{il} | 1 \leq i \leq n, 1 \leq j \leq 2^c\}$, a,b, ξ, Δ, $\mathscr{M}$, $\mathscr{C}$
Output: $\frac{\partial J_{CEVCLUS}}{\partial a}$, $\frac{\partial J_{CEVCLUS}}{\partial b}$, $\{\frac{\partial J_{CEVCLUS}}{\partial \alpha_{il}} | 1 \leq i \leq n, 1 \leq l \leq 2^c\}$
  for $i = 1$ to n **do**
    for $l = 1$ to 2^c **do**
      Compute m_{il} using (15)
    end for
  end for
  for $i = 1$ to n **do**
    for $j = i+1$ to n **do**
      Compute the degree of conflict K_{ij} between m_i and m_j
    end for
  end for
  Compute $\frac{\partial J_{CEVCLUS}}{\partial a}$ using (16a)
  Compute $\frac{\partial J_{CEVCLUS}}{\partial b}$ using (16b)
  for $i = 1$ to n **do**
    for $l = 1$ to 2^c **do**
      Compute $\frac{\partial J_{EVCLUS}}{\partial \alpha_{il}}$ using (16c)
      Compute $\frac{\partial J_{CONST}}{\partial \alpha_{il}}$ using (21)
      $\frac{\partial J_{CEVCLUS}}{\partial \alpha_{il}} = \frac{\partial J_{EVCLUS}}{\partial \alpha_{il}} + \xi \frac{1}{2(|\mathscr{M}|+|\mathscr{C}|)} \frac{\partial J_{CONST}}{\partial \alpha_{il}}$
    end for
  end for

As in EVCLUS, the CEVCLUS algorithm may converge to a local minimum. Thus, to obtain an optimal final solution, a suitable strategy is to perform several experiments using different starting points and keep the solution with the smallest criterion value.

[1] A Matlab implementation of the CEVCLUS algorithm is available at https://www.hds.utc.fr/˜tdenoeux.

We can remark that the computational complexity of CEVCLUS (as that of EVCLUS) depends on n, the number of objects and c, the number of clusters. As a matter of fact, as most relational clustering, the complexity of CEVCLUS scales quadratic with the number of object and increases exponentially with the number of clusters, since it searches the set of all the subsets of Ω, i.e., 2^c subsets. Thus, the algorithm has a complexity of $\mathscr{O}(n^2 \times 2^c)$. This issue can be solved by reducing the number of subsets, for example by constraining the focal sets to be either $\emptyset$, Ω or to be composed of at most two classes. In this way, the number of subsets decreases to $c^2 n$. However, the unconstrained version (with 2^c subsets) was used in the experiments described below.

3.3.1 Active learning

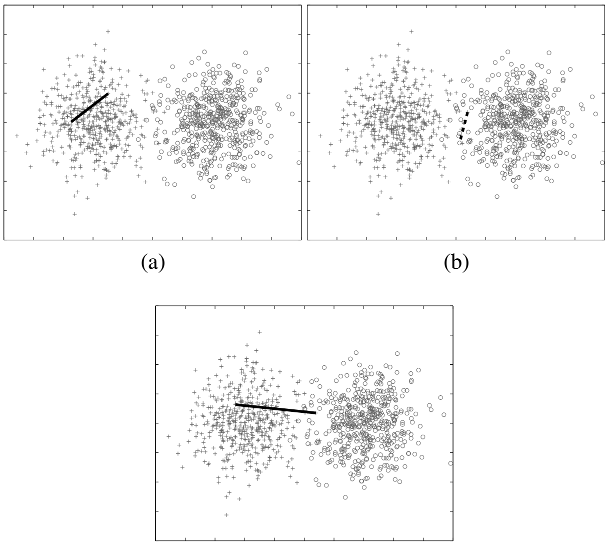
In some cases, there is no obvious way to get or to automatically create pairwise constraints with the background knowledge, but an expert may be able to provide relevant information. It is then possible to set up a scheme that actively generates pairwise constraints. This approach, referred to as active learning, selects the most informative and the less redundant pairs of objects [12, 1]. These objects are then presented to an expert, in order to identify the nature of the corresponding constraints. The goal of this method is to improve clustering performance using as few queries as possible.

As remarked in [6, 34], constraints must be selected carefully: some constraints may be non-informative (Figure 1(a)) or even deteriorate the clustering performances (Figure 1(b)). Thus, to select a pair of objects, we implemented the approach proposed in [1] and represented in Figure 1(c):

- The first object is selected as an object classified with a high degree of uncertainty, in order to build an informative constraint.
- The second object corresponds to an object classified with a low degree of uncertainty. If not, the constraint may result in the misclassification of the two objects.

Here, we propose to use the non-specificity measure defined by (9) to quantify the degree of certainty regarding the class membership of an object. The points whose class membership is uncertain are characterized by higher values of non-specificity.

In most papers [1, 12], in order to conduct experiments, the true relationship between pairs of objects selected by an active learning scheme is identified using the true partition of the data. However, in the context of an application, an expert may find it difficult to determine the nature of a constraint. If he/she has some doubts, he/she will not provide the type of link between the two objects. However, it can be very important to obtain some information about the selected point that is classified with uncertainty. Therefore, we propose to ask the expert to provide its relationships with several other objects classified with certainty in different clusters. Thus, we increase the chance to retrieve at least one informative constraint. Algorithm 3 outlines the steps achieved during the active learning phase.



(a) (b)

(c)

Fig. 1 Pairwise constraints patterns for a toy dataset where symbols represent the real class. Some constraints are useless (a), some may cause misclassification if both constrained objects are classified with a low degree of uncertainty (b) and some are informative (c), leading the algorithm towards better performances.

Algorithm 3 Active learning scheme.

Input: credal partition M, number of cluster c
Output: a set $\mathscr{P}$ of pairwise links
$\mathscr{P} = \{\}$
Compute crisp partition $\hat{P}$ (maximal pignistic probability rule on M)
Compute for each object the non-specificity measure N (9)
Select the first object $o_{i^\star}$ such that $i^\star = \arg\max_{i=1...n} N(m_i)$
for $k = 1$ to c **do**
  Select object $o_{j^\star}$ such that $j^\star = \arg\min_{o_j \in \omega_k} N(m_j)$
  Add $(o_{i^\star}, o_{j^\star})$ in $\mathscr{P}$
end for

Note that if the number of links required is different from the number of clusters, then the algorithm can simply be performed as many time as needed and finally the last pairwise links obtained can be truncated in order to exactly fit with the desired number of links.

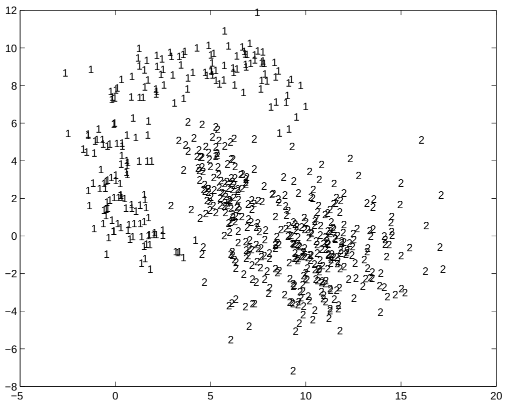


Fig. 2 Toys2c dataset.

Table 3 Datasets used in the experiments

Datasets	Number of objects n	Number of classes c	Number of attributes
Wine	178	3	13
Letters	227	3	16
Ecoli	272	3	7
20Newsgroups	1136	4	100
ChickenPieces	446	5	/
Toys2c	800	2	2

4 Experiments

4.1 Methodology

4.1.1 Datasets

In order to illustrate the interest of our approach, experiments were first conducted using four well-known datasets from the UCI repository[2]: Wine, Letters, Ecoli, 20Newsgroups and a synthetic dataset: Toys2c. Toys2c consists of two classes in a two-dimensional space. The first class was generated according to a Gaussian distribution transformed with the Cartesian equation of an ellipse and the second one corresponds to the concatenation of two Gaussians (cf. Figure 2). Thus the separation between the two clusters is nonlinear.

In addition, we carried out experiments on the ChickenPieces dataset, which is composed of 446 binary images. Each image represents the silhouette of a specific part of the chicken. Table 3 synthesizes the main characteristics of the six datasets.

In our experiments, we chose to normalize the Wine dataset. Furthermore, the Letters dataset was transformed as proposed in [2]: we kept only three classes, corresponding to the three letters $\{I, J, L\}$, and we randomly selected 10% of the data in each class.

2 Available at http://archive.ics.uci.edu/ml.

Table 4 The characteristics of the 20Newsgroups dataset.

class	subject	number of objects
1	Computer science	322
2	Entertainment	247
3	Natural sciences	186
4	Controversial subjects	382
	total	1136

Similarly, the 20Newsgroups database was reduced. This dataset is originally composed of messages collected from 20 different newsgroups[3]. Each message is described by a 100-dimensional feature vector corresponding to encoding the presence or the absence of 100 given words in the message. For our experiments, we created a sample by keeping only four topics (corresponding to the classes) and by randomly selecting 7% of patterns from each class, as explained in [29]. Table 4 presents the characteristics of this dataset.

We can remark that most of the datasets contain attribute data. Since CEVCLUS is a relational clustering algorithm, data should be transformed in order to provide dissimilarity matrices. For the Wine, Letters, Ecoli and Toys2c datasets, we suppose we do not have background knowledge about the best distance to use, so we chose by default to compute the Euclidean distance between the available objects. Since the 20Newsgroups database corresponds to a binary dataset, we used a distance based on the correlation between the objects [29]:

$$D_{corr}(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{2}\left(1 - \frac{\mathbf{x}_1{}^\top \mathbf{x}_2}{\|\mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2 - 2\mathbf{x}_1{}^\top \mathbf{x}_2}\right). \tag{22}$$

Finally, for the ChickenPieces dataset, dissimilarity matrices were obtained using pairwise comparisons on the contours of the images [4]: first, the contour lines of the silhouette are detected using means of an edge detector. The contour lines are then transformed into vectors with a constant length in order to construct a string consisting of the angles between the consecutive vectors. The result leads to a rotation-invariant cyclic string. Distances beetween the different images are finally computed using a cyclic string edit distance algorithm on the strings. We chose for our experiments the dissimilarity matrix $S = (s_{ij})$ called chickenpieces-20-90[4]. Since the data are slightly asymmetric, we computed a new matrix $D = (\delta_{ij})$ as follows: $\delta_{ij} = \frac{1}{2}(s_{ij} + s_{ji})$.

4.1.2 Performance evaluation

As explained in Section 2.2, it is possible to obtain a fuzzy partition by applying the pignistic transformation (8) to the credal partition computed by CEVCLUS. This fuzzy partition can then be transformed into a crisp partition $\hat{P}$ using the

3 Available at http://people.csail.mit.edu/jrennie/20Newsgroups.
4 Available on http://algoval.essex.ac.uk/data/sequence/chicken.

maximal pignistic probability rule. Since the actual partition P is known for all the datasets used, we can evaluate the accuracy of our clustering algorithm by comparing P with the crisp partition $\hat{P}$. A popular measure of agreement between two partitions P and $\hat{P}$ is the Rand Index (RI) defined as:

$$RI(P,\hat{P}) = \frac{2(f+g)}{n(n-1)}, \tag{23}$$

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

4.1.3 Choice of the constraints

In this work, three methods for choosing pairwise constraints have been used. First, we used random selection, which consists in randomly picking pairs of objects in a dataset. The true relationship between the two objects is then determined using their real labels. This technique allows us to study the behavior of the algorithm in various situations. It is used in the experiments reported in Sections 4.2.1 and 4.2.2 to demonstrate the interest of adding constraints and to compare the performances of our algorithm with those of other relational constrained clustering methods.

In real applications, instance-level constraints may sometimes be obtained from prior domain knowledge. Section 4.3.1 presents an example of such a situation.

Finally, the active learning strategy proposed in Section 3.3.1 is tested in Section 4.3.2.

4.1.4 Guidelines for tuning ξ

Parameter ξ controls the tradeoff between the pairwise constraints and the fit with the distance data. To find a suitable value for this parameter, several experiments were carried out on some of the UCI datasets. Various values of ξ were tried for a fixed number of constraints. Tables 5 and 6 present the evolution of the average Rand Index (computed over 100 trials with a random selection of the constraints for each trial) for different values of ξ. The confidence intervals were computed as $\overline{R} \pm t_{0.975}\sigma/10$, where $\overline{R}$ is the average Rand index over the 100 trials, σ is the standard deviation, and $t_{0.975}$ is the 0.975 quantile of the Student t distribution with 99 degrees of freedom. In order to avoid local minima, each trial consists of five runs of CEVCLUS with different initializations, including one initialization coming from the credal partition resulting of EVCLUS.

These results show that the performances of our algorithm generally do not depend much on the precise value of ξ, particularly when the number of constraints is small. For the three datasets and the four values of C, the RI remains constant for a wide range of ξ. Whereas the optimal value depends on the dataset and the number of constraints, the

Table 5 Average Rand index and 95% confidence interval over 100 trials as a function of ξ for $C = 20$ and $C = 50$ randomly chosen constraints. The best results are displayed in bold.

C	ξ	Wine	Letters	Ecoli
20	0	0.87 ± 0.00	0.61 ± 0.00	0.88 ± 0.01
	0.05	**0.95** ± 0.00	0.61 ± 0.00	0.90 ± 0.00
	0.1	**0.95** ± 0.00	0.62 ± 0.00	**0.91** ± 0.00
	0.2	0.93 ± 0.00	**0.63** ± 0.00	**0.91** ± 0.00
	0.5	0.93 ± 0.00	**0.63** ± 0.00	**0.91** ± 0.00
	0.8	0.93 ± 0.00	**0.63** ± 0.00	**0.91** ± 0.00
	1	0.93 ± 0.00	**0.63** ± 0.00	**0.91** ± 0.00
	1.5	0.93 ± 0.00	0.62 ± 0.00	**0.91** ± 0.00
	2	0.93 ± 0.00	**0.63** ± 0.00	**0.91** ± 0.00
	2.5	0.93 ± 0.00	0.62 ± 0.00	**0.91** ± 0.00
	3	0.93 ± 0.00	**0.63** ± 0.00	**0.91** ± 0.00
	5	0.93 ± 0.00	**0.63** ± 0.00	**0.91** ± 0.00
50	0	0.87 ± 0.00	0.61 ± 0.00	0.88 ± 0.01
	0.05	0.95 ± 0.00	0.61 ± 0.00	0.91 ± 0.00
	0.1	**0.96** ± 0.00	0.62 ± 0.00	0.91 ± 0.00
	0.2	0.95 ± 0.00	0.63 ± 0.00	0.92 ± 0.00
	0.5	0.95 ± 0.00	**0.65** ± 0.01	0.92 ± 0.00
	0.8	0.94 ± 0.00	0.64 ± 0.00	**0.93** ± 0.00
	1	0.94 ± 0.00	0.64 ± 0.00	0.92 ± 0.00
	1.5	0.94 ± 0.00	0.64 ± 0.00	**0.93** ± 0.00
	2	0.93 ± 0.00	0.63 ± 0.00	**0.93** ± 0.00
	2.5	0.93 ± 0.01	0.64 ± 0.00	0.92 ± 0.00
	3	0.93 ± 0.00	0.63 ± 0.00	0.92 ± 0.00
	5	0.93 ± 0.00	0.64 ± 0.00	0.92 ± 0.00

Table 6 Average Rand index and 95% confidence interval over 100 trials as a function of ξ for $C = 100$ and $C = 200$ randomly chosen constraints. The best results are displayed in bold.

C	ξ	Wine	Letters	Ecoli
100	0	0.87 ± 0.00	0.61 ± 0.00	0.88 ± 0.01
	0.05	0.95 ± 0.00	0.61 ± 0.00	0.91 ± 0.00
	0.1	**0.96** ± 0.00	0.61 ± 0.00	0.91 ± 0.00
	0.2	**0.96** ± 0.00	0.63 ± 0.00	0.93 ± 0.00
	0.5	**0.96** ± 0.00	**0.71** ± 0.01	**0.94** ± 0.00
	0.8	**0.96** ± 0.00	**0.71** ± 0.01	**0.94** ± 0.00
	1	**0.96** ± 0.00	0.68 ± 0.01	**0.94** ± 0.00
	1.5	0.95 ± 0.00	0.65 ± 0.01	**0.94** ± 0.00
	2	0.94 ± 0.01	0.65 ± 0.00	**0.94** ± 0.00
	2.5	0.93 ± 0.01	0.65 ± 0.00	**0.94** ± 0.00
	3	0.92 ± 0.01	0.64 ± 0.00	**0.94** ± 0.00
	5	0.92 ± 0.01	0.64 ± 0.00	**0.94** ± 0.00
200	0	0.87 ± 0.00	0.61 ± 0.00	0.88 ± 0.01
	0.05	0.96 ± 0.00	0.61 ± 0.00	0.91 ± 0.00
	0.1	0.97 ± 0.00	0.61 ± 0.00	0.91 ± 0.00
	0.2	0.97 ± 0.00	0.63 ± 0.00	0.94 ± 0.00
	0.5	**0.98** ± 0.00	0.79 ± 0.01	**0.96** ± 0.00
	0.8	**0.98** ± 0.00	**0.84** ± 0.01	**0.96** ± 0.00
	1	**0.98** ± 0.00	0.83 ± 0.01	**0.96** ± 0.00
	1.5	**0.98** ± 0.00	0.77 ± 0.01	**0.96** ± 0.00
	2	0.97 ± 0.00	0.71 ± 0.01	**0.96** ± 0.00
	2.5	0.96 ± 0.01	0.68 ± 0.01	**0.96** ± 0.00
	3	0.94 ± 0.01	0.67 ± 0.01	**0.96** ± 0.00
	5	0.95 ± 0.01	0.62 ± 0.01	**0.96** ± 0.00

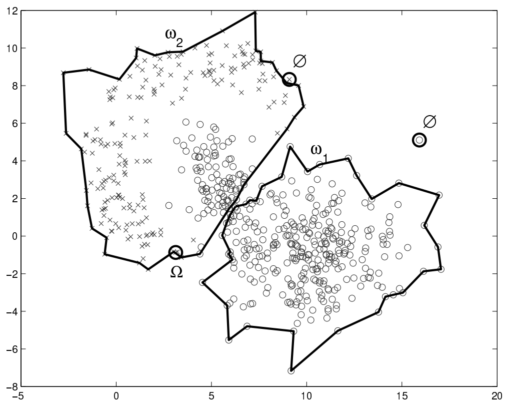


Fig. 3 Hard credal partition obtained using EVCLUS on the Toys2c dataset. Each point is represented by a symbol corresponding to its real class.

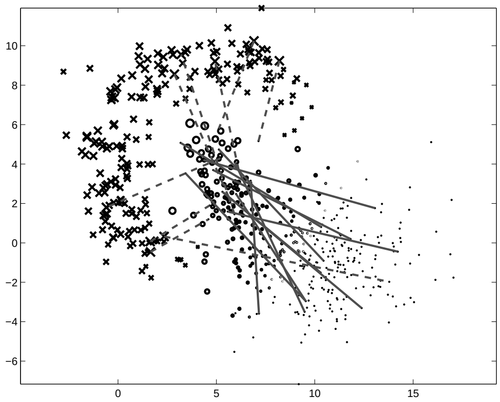


Fig. 4 Belief masses assigned to ω_1 using CEVCLUS on the Toys2c dataset. Each point is represented by a symbol corresponding to its real class. The size of each symbol is proportional to the mass function assigned to the point. Solid and dashed lines represent Must-Link and Cannot-Link constraints, respectively.

value $\xi = 1$ generally yields close to optimal results. This value has been adopted in the rest of our experiments.

4.1.5 Comparison with reference methods

The performances of CEVCLUS have been compared to those of two constrained clustering methods dedicated to relational data. The first one is referred to as CCL [16] and represents the first relational clustering algorithm proposed to integrate background knowledge. It is a modification of the complete-link approach in hierarchical clustering [9]. In this algorithm, the input dissimilarity matrix is altered using pairwise constraints. In order to compare the results with ours, we cut the final dendrogram to obtain a crisp partition with the appropriate number of classes. The second method, called SSCARD [10], is based on the relational FCM algorithm. As in CEVCLUS, a penalty term is added to the objective function in order to integrate pairwise constraints. This algorithm generates a fuzzy partition. It can be then considered as the closest algorithm to CEVCLUS.

We can remark that other constrained clustering methods exist, such as algorithms dedicated to attribute data [1,2,12, 35] or graph data [18,37]. However, we have restricted the comparison to relational clustering algorithms.

4.2 Results

4.2.1 Benefits of adding constraints

We first illustrate the benefits of adding constraints using the Toys2c dataset. The hard credal partition given by EVCLUS is presented in Figure 3. We can see that the algorithm was unable to detect to shape of the classes. Indeed, without any prior information, the algorithm finds a linear boundary between the two groups of objects. The Rand Index obtained is 0.73.

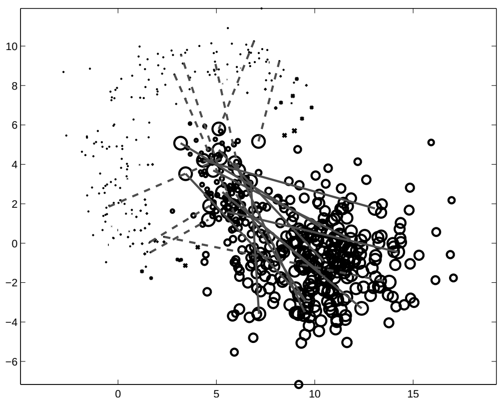


Fig. 5 Belief masses assigned to ω_2 using CEVCLUS on the Toys2c dataset. Each point is represented by a symbol corresponding to its real class. The size of each symbol is proportional to the mass function assigned to the point. Solid and dashed lines represent Must-Link and Cannot-Link constraints, respectively.

As shown in Figures 4 and 5, adding 20 constraints leads the algorithm towards a different, more suitable solution. The hard credal partition displays a non linear boundary which is closer to the desired boundary. The new Rand index has increased to 0.90.

The evolution of the average Rand Index (computed over 100 trials) as a function of the number of pairwise constraints for the Toys2c, Wine, Letters and Ecoli datasets is represented in Figures 6, 7, 8 and 9 respectively. For each trial, CEVCLUS was run 10 times with different initializations, in order to avoid local minima.

We first remark that the total Rand Index increases with the number of constraints. Thus, introducing constraints improves the classification accuracy on constrained objects. Most of the time, we also observe that the Rand Index computed over unconstrained objects increases with the number of constraints too. Therefore, constraining some pairs of

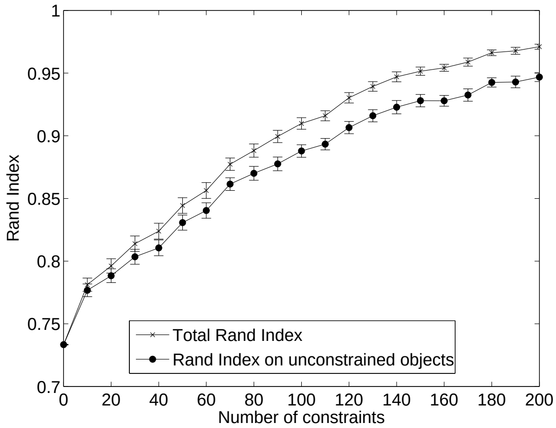


Fig. 6 Toys2c dataset : average Rand Index and 95% confidence interval as a function of the number of randomly selected constraints.

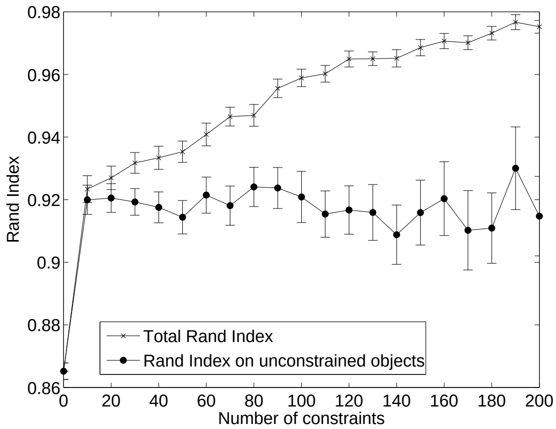


Fig. 7 Wine dataset : average Rand Index and 95% confidence interval as a function of the number of randomly selected constraints.

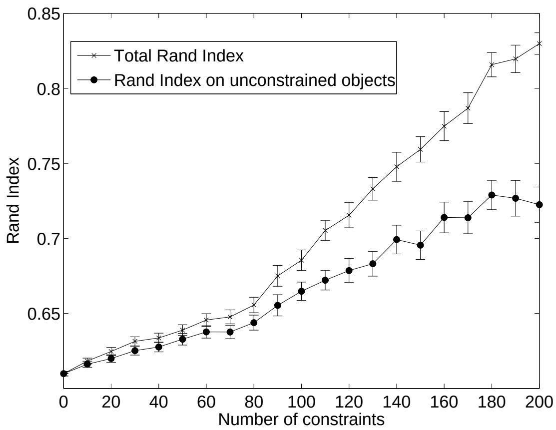


Fig. 8 Letters dataset : average Rand Index and 95% confidence interval as a function of the number of randomly selected constraints.

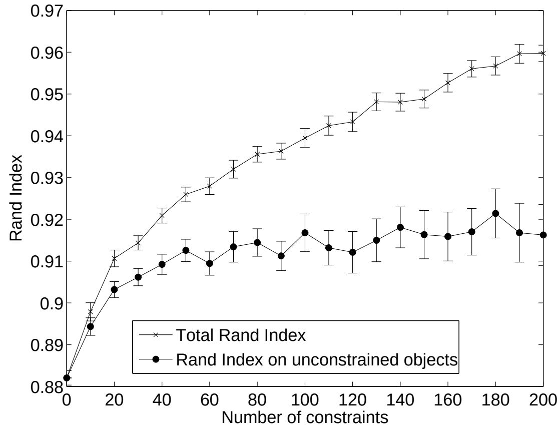


Fig. 9 Ecoli dataset : average Rand Index and 95% confidence interval as a function of the number of randomly selected constraints.

objects also improves the clustering of other, unconstrained objects.

However, this behavior is not observed with the Wine dataset (Figure 7). Indeed, the EVCLUS algorithm already yields good results on this dataset. Boundaries between classes do not need to be strongly modified, but a local refinement slightly improves the quality of the clustering. As a matter of fact, regions where objects are classified with uncertainty correspond to areas where real classes are mixed. In those regions, an unconstrained object can be close to constrained objects assigned to different classes. It leads the unconstrained object towards an uncertain class membership. Conversely, constrained objects located in those regions are well-classified thanks to their constraints. Hence, such objects contribute to obtaining better results, as shown by Figure 7.

4.2.2 Performance comparison

The results with CEVCLUS were compared to those obtained with CCL [16] and SSCARD [10] on the six datasets. For the three algorithms, we used the same input dissimilarities and the same number of clusters. Credal and fuzzy partitions were transformed into crisp ones in order to make decisions. Figures 10 to 15 present the evolution of the mean Rand Index obtained (computed over 100 trials) according to the number of constraints.

These experimental results clearly show the superiority of our approach. Indeed, most of the time CEVCLUS is equivalent to or outperforms SSCARD and CCL, and it generally achieves better results when there are a few constraints. These results can be simply explained: instance-level constraints contain useful information allowing us to improve the clustering solution and the credal partition provides the flexibility to easily accomplish this task.

The behavior of the algorithms may sometimes seem surprising at first glance. For example, adding constraints in

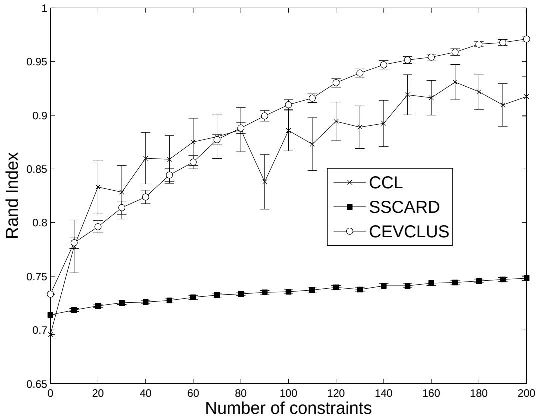


Fig. 10 Average Rand Index and its 95% confidence interval obtained by different algorithms for the Toys2c dataset.

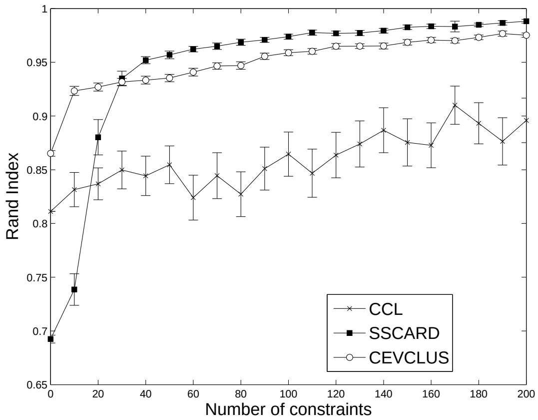


Fig. 11 Average Rand Index and its 95% confidence interval obtained by different algorithms for the Wine dataset.

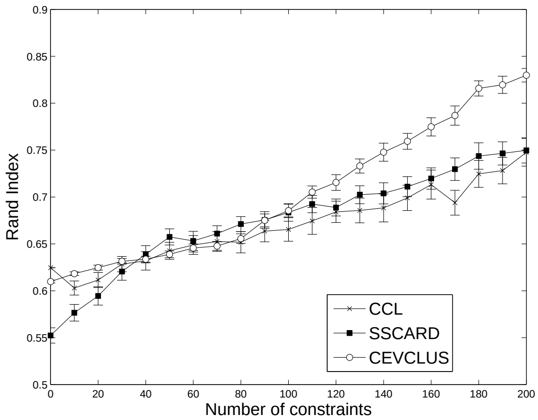


Fig. 12 Average Rand Index and its 95% confidence interval obtained by different algorithms for the Letters dataset.

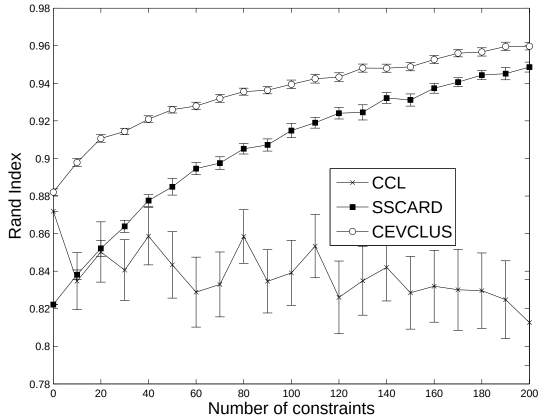


Fig. 13 Average Rand Index and its 95% confidence interval obtained by different algorithms for the Ecoli dataset.

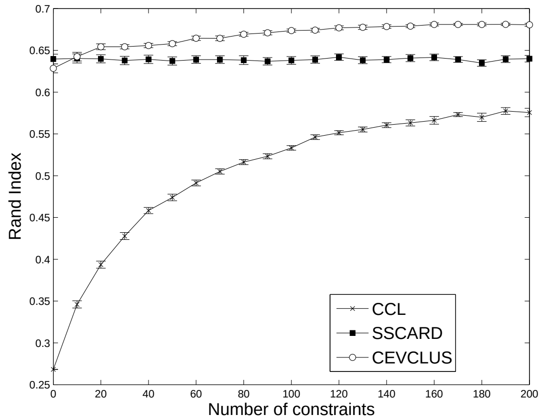


Fig. 14 Average Rand Index and its 95% confidence interval obtained by different algorithms for the 20Newsgroups dataset.

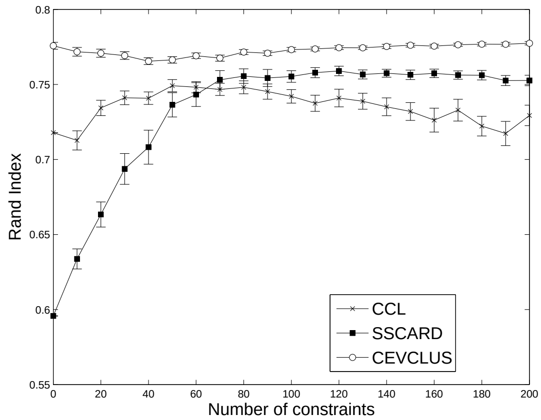


Fig. 15 Average Rand Index and its 95% confidence interval obtained by different algorithms for the ChickenPieces dataset.

the CCL algorithm sometimes decreases the quality of the solution. Indeed, for CCL, a set of constraints modifies distances and influences the aggregation of clusters. Depending on the dataset, this can result in better (Figure 10) or worse performances (Figure 13). We can also remark that CCL is more sensitive to the choice of constraints than CEVCLUS or SSCARD, as shown by the larger confidence intervals.

For the Wine dataset, the SSCARD algorithm slightly outperforms CEVCLUS when more than 30 constraints are used. As a matter of fact, with CEVCLUS, the unconstrained objects have a high degree of uncertainty regarding their class membership. This indecision leads to a slower improvement of the partition as compared to SSCARD.

Finally, we may remark that, for the three algorithms, adding constraints sometimes have no impact on the results, or a very small one (see Figure 14). This phenomenon concerns datasets that need either a higher number of constraints or a particular selection of the constraints. The next paragraphs investigate both cases in greater detail.

4.3 Applications

4.3.1 Generation of constraints by simple rules

In all the above experiments, pairwise constraints were artificially generated using the random selection method. In this following section, we show how Must-Link and Cannot-Link constraints may be retrieved using real domain knowledge. For that purpose, we used the 20Newsgroups database, which is composed of messages. In order to constitute two sets of Must-Link and Cannot-Link constraints, we built some rules from the characteristics of the messages.

The basic idea is to define a Must-Link constraint between two messages when they contain at least two preselected words, and a Cannot-Link constraint when each message includes a pair of words related to different topics. For example, all the documents including the words “Bible” and “religion” should be assigned to the same class, whereas a document containing these two words and a document containing the words “computer” and “disk” should not belong to the same class. It should be emphasized that we decided to select pairs of words and not single ones for the creation of constraints because a word that is supposed to belong to a group may appear in an other group. For instance, the word “mac” is a computer science word, but is also the prefix of family names. Consequently, it can be found in messages from every group. Figure 16 presents the sets of words associated with the different topics.

In that way, 3947 constraints were created. We can remark that, among these constraints, some of them may have been incorrectly defined. Indeed, based on the real label of the constrained objects, we observed that 4.1% of constraints correspond a wrong type of link, i.e., a Cannot-Link (a Must-Link, respectively) for two objects in the same class (in a different class, respectively).

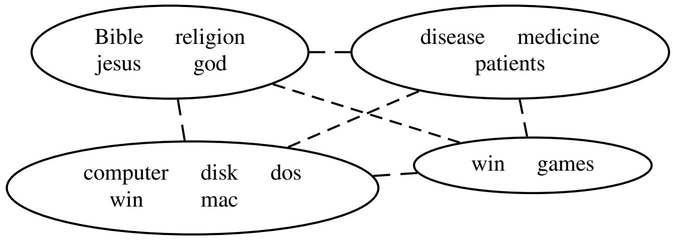


Fig. 16 Scheme of the rules used to build Must-Link and Cannot-Link constraints for the 20Newsgroups dataset. The circles represent groups of identical words and the dashed lines linking groups correspond to the notion of dissimilarity between them.

Table 7 Rand index for the 20newsgroups dataset using CEVCLUS, SSCARD and CCL.

nb constraints	CEVCLUS	SSCARD	CCL
0	0.63	0.64	0.27
3947	0.68	0.65	0.54

Table 8 The ChickenPieces characteristics.

class	piece	number of objects
1	breast	96
2	back	76
3	thigh and back	61
4	wings	117
5	drumstick	96
	total	446

First, the EVCLUS algorithm was run, yielding a Rand index of 0.63. Then, ten trials of CEVCLUS were carried out with $\xi = 1$, and we selected the solution with the lowest value of the objective function. The Rand Index obtained was increased to 0.68. Hence, incorporating constraints allowed us to improve the solution. As a comparison, Table 7 presents the results obtained using the same set of constraints with SSCARD and CCL. The best performances are obtained using the CEVCLUS algorithm.

We can observe that, as expected, the results reported in Table 7 are similar to those obtained using random constraints (see Figure 14). As a matter of fact, this section only stresses out a concrete case of constraint generation using the background knowledge.

4.3.2 Active learning

The active learning strategy was tested with the ChickenPieces dataset. We recall that the dataset is composed of 446 binary images, each one of which represents the silhouette of a specific part of the chicken. Table 8 shows the distribution of the classes.

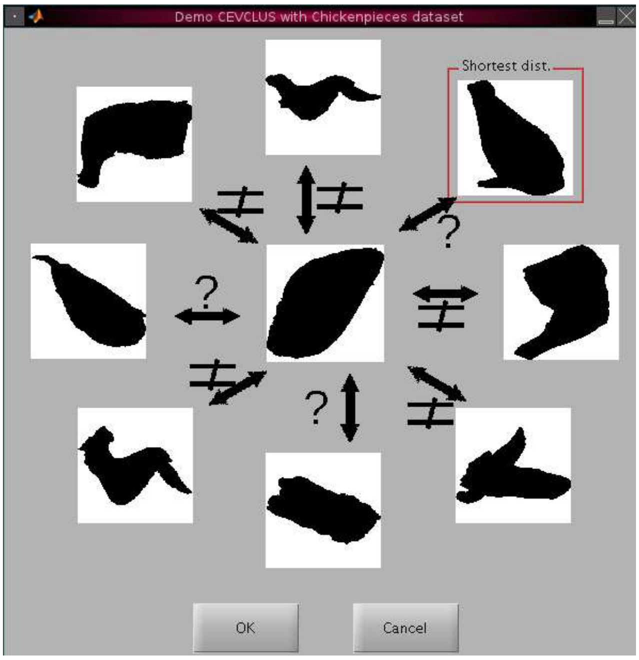


Fig. 17 ChickenPiece dataset: graphical user interface for active learning. Symbols "?" (respectively, "$\neq$") correspond to a unknown relationship (respectively, Cannot-link constraint) between two objects. Although not represented in the figure, a Must-Link constraint can also be set with the symbol "$=$".

First, we performed five times the EVCLUS algorithm and select the output having the lowest objective function in order to get an initial credal partition. Objects were then selected using the non-specificity measure and were presented to an expert via a graphical user interface (Figure 17). The image in the center represents the object of interest (that is classified with uncertainty) while the others are references (i.e., objects classified with certainty). Remark that the closest reference to the object of interest is indicated to the expert by a frame. It corresponds to the constraint supposing to be one of the most informative. The expert determines the nature of the constraints he/she knows. Then, CEVCLUS can be applied. The new credal partition makes it possible to incorporate additional constraints, and CEVCLUS can be run again. This process is iterated until the stabilization of the credal partition.

Table 9 shows the evolution of the Rand index as a function of the number of questions. We can observe that using even a few constraints significantly improves the final solution. As no random process is involved in the active learning scheme, it always yields the same results if the expert does not modify his/her answers.

The results can be compared to those reported in Figure 15. We observe that the proposed active learning strategy yields better partitions than those obtained using random constraint selection, even with a a very small number of questions. This confirms the interest of using this strategy when expert knowledge is available.

Table 9 ChickenPiece dataset: evolution of the Rand Index during the active learning scheme

nb of questions	nb constraints	RI
1	0	0.76
2	5	0.80
3	12	0.82
4	19	0.83
5	20	0.83

5 Conclusion

A new constrained clustering method has been introduced. The new algorithm, called CEVCLUS, is based on the EVCLUS [8] algorithm, proposed in the theoretical framework of belief functions. It is designed for dissimilarity data and makes it possible to integrate background knowledge in the form of pairwise constraints.

Experiments conducted on a synthetic database have shown that introducing constraints allows us to guide the algorithm towards a desired solution. We have then illustrated the performances of the algorithm on various datasets, and compared our approach with two other constrained clustering methods. Our method outperforms the reference methods in most experiments, demonstrating the interest of exploiting the credal partition in order to elicit constraints.

Finally, two applications have been proposed in order to present different ways of retrieving pairwise constraints. The first one consists in generating rules to construct constraints from domain knowledge. Such a strategy may provide an large amount of constraints, which are not necessarily informative but are fast to create. The second strategy corresponds to an active learning scheme and involves the use of a graphical user interface. In order to select automatically informative pairs of patterns, we took advantage of the richness of the evidential framework. As an active learning scheme corresponds to a human-based approach, this method provides fewer constraints than the former approach, but they proved to be very informative.

In both cases, we assumed that the constraints are assigned with total certainty. In the future, we intend to consider soft constraints, following [19]. We may address this problem in the evidential framework by associating a degree of plausibility to each constraint. Taking into account such constraints while keeping the resulting optimization problem tractable is a difficult problem, which we are currently investigating.

Acknowledgment

This work was carried out in the framework of the Labex MS2T, which was funded by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

References

[1] Antoine V., Quost B., Masson M.-H., Denœux T. (2012) CECM: Constrained Evidential C-Means algorithm. Computational Statistics and Data Analysis 56:894–914
[2] Basu S., Bilenko M., Banerjee A., Mooney R. (2006) Semi-Supervised Learning, Chapelle, O. and Schölkopf, B. and Zien, A., MIT Press Cambridge, MA, USA, chap Probabilistic semi-supervised clustering with constraints, pp 71–98
[3] Bezdek J.C. (1981) Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell, MA, USA
[4] Bunke H., Bühler U. (1993) Applications of approximate string matching to 2d shape recognition. Pattern recognition 26(12):1797–1812
[5] Davidson I., Ravi S. (2005) Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In: Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), Springer, Porto, Portugal, vol 3721, pp 59–70
[6] Davidson I., Wagstaff K., Basu S. (2006) Measuring constraint-set utility for partitional clustering algorithms. In: Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), Berlin, Germany, vol 4213, pp 115–126
[7] Dempster A. (1967) Upper and lower probabilities induced by multivalued mapping. Annals of Mathematical Statistics AMS-38:325–339
[8] Denœux T., Masson M.-H. (2004) EVCLUS: evidential clustering of proximity data. IEEE Transactions on Systems, Man and Cybernetics: B 34(1):95–109
[9] Everitt BS., Landau S., Leese M. (2009) Cluster Analysis, 4th edn, Wiley, chap Hierarchical clustering, pp 55–89
[10] Frigui H., Hwang C. (2007) Adaptive concept learning through clustering and aggregation of relational data. In: Proceedings of the 7th SIAM International Conference on Data Mining, Minneaplis, USA, pp 90–101
[11] Gondek D., Hofmann T. (2007) Non-redundant data clustering. Knowledge and Information Systems 12(1):1–24
[12] Grira N., Crucianu M., Boujemaa N. (2008) Active semi-supervised fuzzy clustering. Pattern Recognition 41(5):1834–1844
[13] Hamasuna Y., Endo Y. (2012) On semi-supervised fuzzy c-means clustering for data with clusterwise tolerance by opposite criteria. Soft Computing - A Fusion of Foundations, Methodologies and Applications pp 1–11
[14] Hathaway R., Davenport J., Bezdek J. (1989) Relational duals of the c-means clustering algorithms. Pattern Recognition 22(2):205–212
[15] Kannan S., Sathya A., Ramathilagam S. (2011) Effective fuzzy clustering techniques for segmentation of breast mri. Soft Computing - A Fusion of Foundations, Methodologies and Applications 15:483–491
[16] Klein D., Kamvar S., Manning C. (2002) From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, pp 307–314
[17] Klir G., Wierman M. (1999) Uncertainty-based information: Elements of generalized information theory. Springer Verlag, New York
[18] Kulis B., Basu S., Dhillon I., Mooney R. (2005) Semi-supervised graph clustering: A kernel approach. In: 22nd International Conference on Machine Learning (ICML), Bonn, Germany, pp 457–464
[19] Law M., Topchy A., Jain A. (2004) Clustering with soft and group constraints. In: Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Springer, Lisbon, Portugal, vol 3138, pp 662–670
[20] Lazzerini B., Marcelloni F. (2007) A hierarchical fuzzy clustering-based system to create user profiles. Soft Computing - A Fusion of Foundations, Methodologies and Applications 11:157–168
[21] Li Y.L., Shen Y. (2010) An automatic fuzzy c-means algorithm for image segmentation. Soft Computing - A Fusion of Foundations, Methodologies and Applications 14:123–128
[22] Liu Y., Jin R., Jain A. (2007) Boostcluster: boosting clustering by pairwise constraints. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, San Jose, CA, USA, pp 450–459
[23] Masson M.-H., Denœux T. (2004) Clustering interval-valued data using belief functions. Pattern Recognition Letters 25(2):163–171
[24] Masson M.-H., Denœux T. (2008) ECM: An evidential version of the fuzzy c-means algorithm. Pattern Recognition 41(4):1384–1397
[25] Masson M.-H., Denœux T. (2009) RECM: Relational evidential c-means algorithm. Pattern Recognition Letters 30(11):1015–1026
[26] Pal N., Bezdek J., Hemasinha R. (1992) Uncertainty measures for evidential reasoning i: A review. International Journal of Approximate Reasoning 7(3-4):165–183
[27] Pedrycz, W, Loia V., Senatore S. (2004) P-FCM: a proximity-based fuzzy clustering. Fuzzy Sets and Systems 148(1):21–41
[28] Pedrycz, W. (2007) Collaborative and knowledge-based fuzzy clustering. International Journal of Innovative, Computing, Information and Control 1(3):1–12
[29] Pekalska E., Duin R. (2005) The Dissimilarity Representation for Pattern Recognition, vol 64, foundations and applications, world scientific edn. Singapore
[30] Sammon J.W. (1969) A nonlinear mapping for data structure analysis. IEEE Transactions on Computers 18:401–409
[31] Shafer G. (1976) A mathematical theory of evidence. Princeton University Press, Princeton, NJ
[32] Smets P. (1990) The combination of evidence in the transferable belief model. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(5):447–458
[33] Smets P., Kennes R. (1994) The transferable belief model. Artificial Intelligence 66:191–234
[34] Wagstaff K. (2007) Value, cost, and sharing: Open issues in constrained clustering. In: Knowledge Discovery in Inductive Databases (KDID), Springer, Berlin, Germany, vol 4747, pp 1–10
[35] Wagstaff K., Cardie C., Rogers S., Schroedl S. (2001) Constrained k-means clustering with background knowledge. In: Proceedings of the 18th International Conference on Machine Learning, Williamstown, MA, USA, pp 577–584
[36] Xing E., Ng A., Jordan M., Russell S. (2002) Distance metric learning with application to clustering with side-information. In: Proceedings of the 15th Annual Conference on Advances in Neural Information Processing Systems (NIPS), Vancouver, British Columbia, Canada, pp 521–528
[37] Zhenguo L., Jianzhuang L., Xiaoou T. (2009) Constrained Clustering via Spectral Regularization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, pp 421–428
[38] Zhong S., Ghosh J. (2003) A unified framework for model-based clustering. The Journal of Machine Learning Research 4:1001–1037

A Optimization algorithm

The minimization of $J_{CEVCLUS}$ can be performed using any unconstrained nonlinear programming algorithm. In the experiments reported in Section 4, we used the same gradient-based optimization as in [8]. This method is briefly sketched below.

Let $\mathbf{w}$ be the vector of parameters and $J(\mathbf{w})$ the objective function to be minimized. The algorithm is a variant of gradient descent in which each parameter w_i has its own step size η_j, and the step sizes are adapted during the optimization process, depending on the evolution of the objective function and on the sign of the derivatives at successive iterations. Let t be the iteration counter. Let us first assume that the objective function has decreased between iterations $t-1$ and t. Then the following rule is applied to update each step size η_j:

$$\eta_j(t) = \begin{cases} \beta\ \eta_j(t-1) \text{ if } \dfrac{\partial J}{\partial w_j}(t-1) \cdot \dfrac{\partial J}{\partial w_j}(t) > 0 \\ \gamma\, \eta_j(t-1) \ \text{ otherwise,} \end{cases} \tag{24}$$

where $\beta > 1$ and $\gamma < 1$ are two coefficients. Hence, the step size is increased if the derivatives have kept the same sign during two iterations, and it is increased if the sign of the derivative has changed, which indicates that we have "jumped over" a minimum. The parameters are then updated by:

$$w_j(t+1) = w_j(t) - \eta_j(t) \frac{\partial J}{\partial w_j}(t). \tag{25}$$

If now the objective function has increased between iterations $t-1$ and t, all step sizes are decreased simultaneously:

$$\eta_j(t) = \delta\ \eta_j(t-1) \quad \forall j \tag{26}$$

with $\delta < 1$, and the parameters are updated starting from where they were at the previous iteration:

$$w_j(t+1) = w_j(t-1) - \eta_j(t) \frac{\partial J}{\partial w_j}(t-1). \tag{27}$$

As in [8], we set the parameters β, γ and δ to 1.2, 0.8 and 0.5 in our experiments.