



On a New Evidential C-Means Algorithm with Instance-Level Constraints

Jiarui Xie^{1,2}(✉)  and Violaine Antoine¹ 

¹ Clermont Auvergne University, UMR 6158 CNRS, LIMOS,
63000 Clermont-Ferrand, France

jiarxie@foxmail.com, violaine.antoine@uca.fr

² School of Computer Science and Technology, Harbin Institute of Technology,
Harbin 150001, People's Republic of China

Abstract. Clustering is an unsupervised task whose performances can be highly improved with background knowledge. As a consequence, several semi-supervised clustering approaches have proposed to integrate prior information in the form of constraints, generally at the instance-level. Amongst them, evidential semi-supervised clustering algorithms, such as CECM or SECM algorithm, rely on the theoretical foundation of belief function which extends the probabilistic theory and allows us to express many types of uncertainty about the assignment of an object to a cluster. In this framework, no evidential clustering algorithm has ever mixed different types of instance-level constraints. We propose here to combine pairwise constraints and labeled data constraints in order to better retrieve information from the background knowledge. The new algorithm, called LPECM, shows good performances on synthetic and real data sets.

Keywords: Labeled data constraints · Pairwise constraints · Instance-level constraints · Belief function · Evidential clustering · Semi-supervised clustering

1 Introduction

Clustering is a classical data analysis method that aims at creating natural groups from a set of objects by assigning similar objects into the same cluster while separating dissimilar objects into different clusters. Clustering solutions can be expressed in the form of a partition. Amongst partitional clustering methods, some produce hard [6, 18], fuzzy [10, 19] and credal partitions [2–4, 14]. A hard partition assigns an object to a cluster with total certainty whereas a fuzzy partition allows us to represent the class membership of an object in the form of a probabilistic distribution. The credal partition, developed in the framework of belief function theory, extends the concepts of hard and fuzzy partition. It makes possible the representation of both uncertainty and imprecision regarding the class membership of an object.

Clustering is a challenging task since various clustering solutions can be valid although distinct. In order to lead clustering methods towards a specific

and desired solution, semi-supervised clustering algorithms integrate background knowledge, generally in the form of instance-level constraints. In [2, 3, 19], labeled data constraints are taken into account to improve the performances of the clustering. In [4, 6, 10, 18], two less informative constraints are introduced: the must-link constraint, which specifies that two objects have to be in the same cluster and the cannot-link constraint, which indicates that two objects should not be assigned in the same cluster.

The combination of the three types of instance-level constraints can help to retrieve as most information as possible and thus can achieve better performances. However, there exists currently very few methods able to deal with such constraints [17], more particularly, none generates a credal partition. In this paper, we propose to associate two evidential semi-supervised clustering algorithms, the first one handling pairwise constraints and the second one dealing with labeled data constraints. The goal is to create a more general algorithm that can obtain a large number of constraints from the background knowledge and that can generate a credal partition.

The rest of the paper is organized as follows. Section 2 recalls the necessary backgrounds about belief function, credal partition and evidential clustering algorithms. Section 3 introduces the new algorithm named LPECM and presents the objective function as well as the optimization steps. Several experiments are produced in Sect. 4. Finally, Sect. 5 makes a conclusion about the work.

2 Background

2.1 Belief Function and Credal Partition

Evidence theory [15] (or belief function theory) is a mathematical framework that enables to reflect the state of partial and uncertainty knowledge. Let \mathbf{X} be a data set composed of n objects such that $\mathbf{x}_i \in \mathbb{R}^p$ corresponds to the i^{th} object. Let $\Omega = \{\omega_1, \dots, \omega_c\}$ be the set of possible clusters. The mass function $m_{ik} : 2^\Omega \rightarrow [0, 1]$ applied on the instance \mathbf{x}_i measures the degree of belief that the real class of \mathbf{x}_i belongs to a subset $A_k \subseteq \Omega$. It satisfies:

$$\sum_{A_k \subseteq \Omega} m_{ik} = 1. \tag{1}$$

The collection $\mathbf{M} = [\mathbf{m}_1, \dots, \mathbf{m}_n]$ such that $\mathbf{m}_i = (m_{ik})$ forms a credal partition that is a generalization of a fuzzy partition. Indeed, any subset A_k such that $m_{ik} > 0$ is named a focal set of \mathbf{m}_i . When all focal elements are singletons, the mass function is equivalent to a probability distribution. If such situation occurs for all objects, the credal partition \mathbf{M} can be seen as a fuzzy partition.

Several transformations of a mass function \mathbf{m}_i are possible in order to extract particular information. The plausibility function $pl(A) : 2^\Omega \rightarrow [0, 1]$ defined in Eq. (2) corresponds to the maximal degree of belief that could be given to subset A :

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

To make a decision, a mass function can also be transformed into a pignistic probability distribution [16]. Finally, a hard credal partition can be obtained by assigning each object to the subset of cluster with the highest mass. This allows us to easily detect objects located in an ambiguous region.

2.2 Evidential C-Means Algorithm

Evidential C-Means (ECM) [14] is the credibilistic version of Fuzzy C-Means algorithm (FCM) [5]. In the FCM algorithm, each cluster is represented by a point called centroid or prototype. The ECM algorithm, which generates a credal partition, generalizes the cluster representation by considering a centroid \mathbf{v}_k in \mathbb{R}^p for each subset $A_k \subseteq \Omega$. The objective function is:

$$J_{\text{ECM}}(\mathbf{M}, \mathbf{V}) = \sum_{i=1}^n \sum_{A_k \neq \emptyset} |A_k|^\alpha m_{ik}^\beta d_{ik}^2 + \sum_{i=1}^n \rho^2 m_{i\emptyset}^\beta, \tag{3}$$

subject to

$$\sum_{A_k \subseteq \Omega, A_k \neq \emptyset} m_{ik} + m_{i\emptyset} = 1 \quad \text{and} \quad m_{ik} \geq 0 \quad \forall i \in \{1, \dots, n\}. \tag{4}$$

where $|A_k|$ corresponds to the cardinality of the subset A_k , \mathbf{V} is the set of prototypes and d_{ik}^2 represents the squared Euclidean distance between \mathbf{x}_i and the centroid \mathbf{v}_k . Outliers are handled with masses $m_{i\emptyset}, \forall i \in 1, \dots, n$, allocated to the empty set and with the $\rho^2 > 0$ parameter. The two parameters $\alpha \geq 0$ and $\beta > 1$ are introduced to penalize the degree of belief assigned to subsets with high cardinality and to control the fuzziness of the partition.

An extension of the ECM algorithm has been proposed in order to deal with a Mahalanobis distance [4]. Such metric is adaptive and handles various ellipsoidal shapes of clusters, giving more flexibility for the algorithm to better find the inherent structure of the data. Mahalanobis distance d_{ik}^2 between a point \mathbf{x}_i and a subset A_k is defined as follows:

$$d_{ik}^2 = \|\mathbf{x}_i - \mathbf{v}_k\|_{\mathbf{S}_k}^2 = (\mathbf{x}_i - \mathbf{v}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{v}_k), \tag{5}$$

where \mathbf{S}_k represent the evidential covariance matrix associated to subset A_k and is calculated as the average of the covariance matrices of the singletons included in subset A_k . Finally, objective function (3) has to be minimized with the respect to the credal partition matrix \mathbf{M} , the centroids matrix \mathbf{V} and the covariance matrix $\mathbf{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_c\}$ the set composed of covariance matrices dedicated to clusters.

2.3 Evidential Constrained C-Means Algorithm

Several evidential C-Means based algorithms have already been proposed [1–4, 8, 13] to deal with background knowledge. For each of them, constraints are

expressed in the framework of a belief function and a term penalizing the constraints violation is incorporated in the objective function of the ECM algorithm.

In [2,3], labeled data constraints are introduced in the algorithms, i.e. the expert can express the uncertainty about the label of an object by assigning it to a subset. Objective functions of the algorithms are written in such a way that any mass function which partially or fully respects a constraint on a specific subset has a high weighted plausibility given to a singleton included in the subset.

$$T_{ij} = T_i(A_j) = \sum_{A_j \cap A_i \neq \emptyset} \frac{|A_j \cap A_i|^{\frac{r}{2}}}{|A_i|^r} m_{i|}, \quad \forall i \in \{1 \dots n\}, A_i \subseteq \Omega, \quad (6)$$

where $r \geq 0$ is a fixed parameter. Notice that if $r = 0$, then $\frac{|A_j \cap A_i|^{\frac{r}{2}}}{|A_i|^r} = 1$, which implies that T_{ij} is identical to the plausibility pl_{ij} .

In [4], authors assumed that pairwise constraints (i.e. must-link and cannot-link constraints) are available. A plausibility to belong or not to the same class is then defined. This plausibility allows us to add a penalty term having high values when there exists a high plausibility that two objects are (respectively are not) in the same cluster although they have a must-link constraint (respectively a cannot-link constraint).

$$\begin{aligned} pl_{l \times j}(\theta) &= \sum_{\{A_l \times A_j \subseteq \Omega^2 \mid (A_l \times A_j) \cap \theta \neq \emptyset\}} m_{l \times j}(A_l \times A_j) \\ &= \sum_{A_l \cap A_j \neq \emptyset} m_l(A_l) m_j(A_j), \end{aligned} \quad (7)$$

$$\begin{aligned} pl_{l \times j}(\bar{\theta}) &= 1 - m_{l \times j}(\emptyset) - bel_{l \times j}(\theta) \\ &= 1 - m_{l \times j}(\emptyset) - \sum_{k=1}^c m_l(A_k) m_j(A_k), \end{aligned} \quad (8)$$

where, θ denotes the event that objects x_i and x_j belong to the same class corresponds to the subset $\{(\omega_1, \omega_1), (\omega_2, \omega_2), \dots, (\omega_k, \omega_k)\}$ within Ω^2 , whereas $\bar{\theta}$ denotes the event that objects x_i and x_j do not belong to the same class corresponds to its complement.

3 The LPECM Algorithm with Instance-Level Constraints

3.1 Objective Function

We propose a new algorithm called Labeled and Pairwise constraints Evidential C-Means (LPECM), which is based on the ECM algorithm [14], handles Mahalanobis distance and combines the advantages of pairwise constraints and labeled data constraints by adding three penalty terms:

$$J_{LPECM}(\mathbf{M}, \mathbf{V}, \mathbf{S}) = \xi J_{ECM}(\mathbf{M}, \mathbf{V}, \mathbf{S}) + \gamma J_{\mathcal{M}}(\mathbf{M}) + \eta J_{\mathcal{C}}(\mathbf{M}) + \delta J_{\mathcal{L}}(\mathbf{M}), \quad (9)$$

with respect to constraints (4). Formulation of J_{ECM} corresponds to equation (3) and (5), $J_{\mathcal{M}}$ is a penalty term used for must-link constraints, $J_{\mathcal{C}}$ is dedicated to cannot-link constraints and $J_{\mathcal{L}}$ handles labeled data constraints. Coefficients ξ , γ , η and δ allow us to give more importance to the structure of the data, the pairwise constraints or the labeled data constraints, respectively.

Penalty terms for pairwise constraints and labeled data constraints are defined similarly to [2, 4]:

$$J_{\mathcal{M}}(\mathbf{M}) = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} \left(1 - (m_{i\emptyset} + m_{j\emptyset} - m_{i\emptyset}m_{j\emptyset}) - \sum_{A_k \subseteq \Omega, |A_k|=1} m_{ik}m_{jk} \right), \quad (10)$$

$$J_{\mathcal{C}}(\mathbf{M}) = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}} \sum_{A_k \cap A_l \neq \emptyset} m_{ik}m_{jl}, \quad (11)$$

$$J_{\mathcal{L}}(\mathbf{M}) = \sum_{i=1}^n \sum_{A_k \subseteq \Omega, A_k \neq \emptyset} b_{ik} \left(1 - \left(\sum_{A_k \cap A_l \neq \emptyset} \frac{|A_k \cap A_l|^{\frac{r}{2}}}{|A_l|^r} m_{il} \right) \right), \quad (12)$$

where b_{ik} denotes whether the i^{th} instance belongs to the subset A_k or not:

$$b_{ik} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is constrained to subset } A_k, \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

It should be emphasized that in this study, unlike [2], each labeled object is constrained to only one subset. Indeed, it makes more coherent the set of constraints retrieved from the background knowledge. Constraints are gathered in three different sets such that \mathcal{M} corresponds to the set of must-link constraints, \mathcal{C} to the set of cannot-link constraints and \mathcal{L} denotes the labeled data constraints set. The $J_{\mathcal{M}}$ function returns the sum of the plausibilities that must-link constrained objects to belong to the same class. Similarly, $J_{\mathcal{C}}$ returns the sum of the plausibilities that cannot-link constrained objects are not in the same class. The $J_{\mathcal{L}}$ term calculates for each labeled object a weighted plausibility to belong to the label.

3.2 Optimization

The objective function is minimized as the ECM algorithm, i.e. by carrying out an iterative scheme where first \mathbf{V} and \mathbf{S} are fixed to optimize \mathbf{M} , second \mathbf{M} and \mathbf{S} are fixed to optimize \mathbf{V} and finally \mathbf{M} and \mathbf{V} are fixed to optimize \mathbf{S} .

Centroids Optimization. It can be observed from (9) that the three penalty terms included in the objective function of the LPECM algorithm do not depend on the cluster centroids. Hence, the update scheme of \mathbf{V} is identical to the ECM algorithm [14].

Masses Optimization. In order to obtain a quadratic objective function with linear constraints, we set parameter $\beta = 2$. A classical optimal approach can then be used to solve the problem [7]. The following equations present how to transform the objective function (9) in order to obtain a format accepted by most usual quadratic optimization function.

Let us define $\mathbf{m}_i^T = (m_{i\emptyset}, m_{i\omega_1}, \dots, m_{i\Omega})$ the vector of masses for object \mathbf{x}_i . The first term of J_{LPECM} is then:

$$J_{ECM}(\mathbf{M}) = \sum_{i=1}^n \mathbf{m}_i^T \Phi^i \mathbf{m}_i, \quad (14)$$

where $\Phi^i = [\phi_{kl}^i]$ is a diagonal matrix of size $(2^c \times 2^c)$ associated to object \mathbf{x}_i and defined such as:

$$\phi_{kl}^i = \begin{cases} \rho^2 & \text{if } A_k = A_l \text{ and } A_k = \emptyset, \\ d_{ik}^2 |A_k|^\alpha & \text{if } A_k = A_l \text{ and } A_k \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

Penalty term used for must-link constraints can be rewritten as follows:

$$J_{\mathcal{M}}(\mathbf{M}) = n_{\mathcal{M}} + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} (\mathbf{F}_{\mathcal{M}}^T \mathbf{m}_i + \mathbf{F}_{\mathcal{M}}^T \mathbf{m}_j) + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} \mathbf{m}_i^T \Delta^{\mathcal{M}} \mathbf{m}_j, \quad (16)$$

where $n_{\mathcal{M}}$ denotes the number of must-link constraints, $\mathbf{F}_{\mathcal{M}}$ is a vector of size 2^c and $\Delta^{\mathcal{M}} = [\delta_{kl}^{\mathcal{M}}]$ corresponds to a matrix $(2^c \times 2^c)$ such that:

$$\mathbf{F}_{\mathcal{M}}^T = \underbrace{[-1, 0, \dots, 0]}_{2^c} \quad \text{and} \quad \delta_{kl}^{\mathcal{M}} = \begin{cases} 1 & \text{if } A_k = \emptyset \text{ or } A_l = \emptyset, \\ -1 & \text{if } A_k = A_l \text{ and } |A_k| = |A_l| = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

The penalty term associated to cannot-link constraints is:

$$J_{\mathcal{C}}(\mathbf{M}) = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}} \mathbf{m}_i^T \Delta^{\mathcal{C}} \mathbf{m}_j, \quad (18)$$

where $\Delta^{\mathcal{C}} = [\delta_{kl}^{\mathcal{C}}]$ is a matrix $(2^c \times 2^c)$ such that:

$$\delta_{kl}^{\mathcal{C}} = \begin{cases} 1 & \text{if } A_k \cap A_l \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

Finally, the penalty term for the labeled data constraints is denoted as follows:

$$J_{\mathcal{L}}(\mathbf{M}) = n_{\mathcal{L}} - \sum_{i=1}^n \mathbf{F}_{\mathcal{L}}^T \mathbf{m}_i, \quad (20)$$

where $n_{\mathcal{L}}$ denotes the number of labeled data constraints and $\mathbf{F}_{\mathcal{L}}$ is a vector of size 2^c such that:

$$\mathbf{F}_{\mathcal{L}}^T = v_{ikl}c_{lk}, \quad \forall A_l \in \Omega, \quad (21)$$

$$c_{lk} = \frac{|A_k \cap A_l|^{\frac{r}{2}}}{|A_l|^r}, \quad (22)$$

$$v_{ikl} = \begin{cases} 1 & \text{if } (\mathbf{x}_i, A_k) \in \mathcal{L} \text{ and } A_k \cap A_l \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \quad (23)$$

where expression $(\mathbf{x}_i, A_k) \in \mathcal{L}$ means that the labeled data constraint on object i is the subset A_k . Function $v_{ikl} = \{0, 1\}$ equals to 1 for subsets A_l that has an intersection with A_k knowing the constraint $\mathbf{x}_i \in A_k$.

Now, let us define $\mathbf{m}^T = (\mathbf{m}_1^T, \dots, \mathbf{m}_n^T)$ the vector of size $n2^c$ containing the masses for each object and each subset, \mathbf{H} a matrix of size $(n2^c \times n2^c)$ and \mathbf{F} a vector of size $n2^c$ such that:

$$\mathbf{H} = \begin{pmatrix} \Phi^1 & \Delta_{12} & \cdots & \Delta_{1n} \\ \Delta_{21} & \Phi^2 & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_{n1} & \cdots & & \Phi^n \end{pmatrix}, \quad \text{where } \Delta_{ij} = \begin{cases} \Delta^{\mathcal{M}}, & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}, \\ \Delta^{\mathcal{C}}, & \text{else if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}, \\ 0, & \text{otherwise.} \end{cases} \quad (24)$$

$$\mathbf{F}^T = (\mathbf{F}_1 \cdots \mathbf{F}_i \cdots \mathbf{F}_n), \quad \text{where } \mathbf{F}_i = t_i \mathbf{F}_{\mathcal{M}} - b_i \mathbf{F}_{\mathcal{L}}, \quad (25)$$

$$t_i = \begin{cases} 1, & \text{if } \mathbf{x}_i \in \mathcal{M}, \\ 0, & \text{otherwise.} \end{cases}, \quad \text{and } b_i = \begin{cases} 1, & \text{if } \mathbf{x}_i \in \mathcal{L}, \\ 0, & \text{otherwise.} \end{cases}. \quad (26)$$

Finally, the objective function (9) can be rewritten as follows:

$$J_{LPECM}(\mathbf{M}) = \mathbf{m}^T \mathbf{H} \mathbf{m} + \mathbf{F}^T \mathbf{m}. \quad (27)$$

3.3 Metric Optimization

It can be observed from (9), the three penalty terms of the LPECM algorithm objective function do not depend on the Mahalanobis distance. Since the set of metric \mathbf{S} only appears in J_{ECM} , the update method is identical to the ECM algorithm [4]. The overall procedure of the LPECM algorithm is summarized in Algorithm 1.

4 Experiments

4.1 Experimental Protocols

Performances and time consumption of the LPECM algorithm have been tested on a toy data set and several classical data sets from UCI Machine Learning Repository [9]. For the Letters data set, we kept only the three letters {I,J,L} as done in [6]. As in [14], fixed parameters associated to the ECM algorithm

Algorithm 1. The LPECM algorithm with an adaptive metric

Require: c : Number of desired clusters; $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ the data set; \mathcal{C} : Set of cannot-link constraints ; \mathcal{M} : Set of must-link constraints ; \mathcal{L} : Set of labeled data constraints ;

Ensure: credal partition matrix \mathbf{M} , centroids matrix \mathbf{V} , distance metric matrix \mathbf{S}

1: Initialization of \mathbf{V} ;

2: **repeat**

3: Calculate the new credal partition matrix \mathbf{M} by solving the quadratic programming problem defined by (27) subject to (4);

4: Calculate the new centroids matrix \mathbf{V} by solving the linear system defined as in the ECM algorithm [14];

5: Calculate the new metric matrix \mathbf{S} and new associated distances using [4];

6: **until** No significant change in \mathbf{V} between two successive iterations;

were set such as $\alpha = 1$, $\beta = 2$ and $\rho^2 = 100$. In order to balance the importance of the data structure, must-link constraints, cannot-link constraints and labeled data constraints respectively, we respectively set $\xi = \frac{1}{n2^c}$, $\gamma = \frac{1}{|\mathcal{M}|}$, $\eta = \frac{1}{|\mathcal{C}|}$ and $\delta = \frac{1}{|\mathcal{L}|}$ as coefficients.

Experiment on a data set consists of 20 simulations with a random selection of the constraints. For each simulation, five runs of the LPECM algorithm with random initialization of the centroids are performed. Then, in order to avoid local optimum, the clustering solution with the minimum value of the objective function is selected.

The accuracy of the obtained credal partition is measured with the Adjusted Rand Index (ARI) [12], which is the corrected-for-chance version of the Rand Index that compares a hard partition with the true partition of a data set. As a consequence, the credal partition generated by the LPECM algorithm is first transformed into a fuzzy partition using the pignistic transformation and then the maximum of probability on each object is retrieved to obtain a hard partition.

4.2 Toy Data Set

In order to show the interest of the LPECM algorithm, we started our experiments with a tiny synthetic data set composed of 15 objects and three classes. Figure 1 presents the hard credal partition obtained using the ECM algorithm. Big cross marks denote the centroid of each cluster. Centroids for subsets with higher cardinalities are not represented to ease the reading. As it can be observed, objects located between two clusters are assigned in subsets with cardinality equal to two. Notice also that, due to the stochastic initialization of the centroids, there may exist a small difference between the results obtained from every execution of the ECM algorithm. After the addition of background knowledge in the form of must-link constraints, cannot-link constraints and labeled data constraints and the execution of the LPECM algorithm with a Euclidean distance, it is interesting to observe that previous uncertainties have vanished.

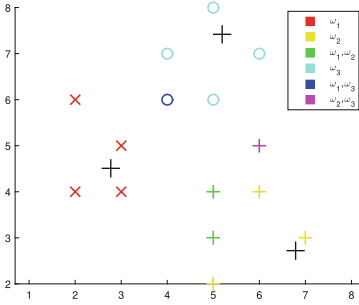


Fig. 1. Hard credal partition obtained on Toy data set with the ECM algorithm

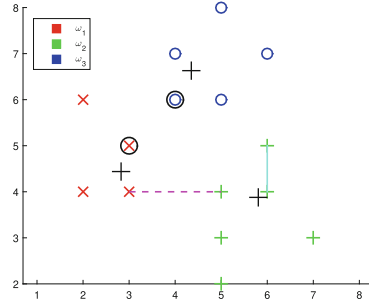


Fig. 2. Hard credal partition obtained on Toy data set with the LPECM algorithm

Figure 2 presents the hard credal partition obtained. The magenta dashed line describes cannot-link constraints, the light green solid line represents must-link constraints and the circled point corresponds to the labeled data constraints .

Figure 3 illustrates, for the execution of the LPECM algorithm, the mass distribution for singletons with respect to the point numbers, allowing us a more distinct sight of the masses allocations. Table 1 displays the accuracy as well as time consumption for the ECM algorithm and the LPECM algorithm when first only the cannot-link constraint is incorporated, second when the cannot-link and the must-link constraint are introduced (Cannot-Must-Link line in Table 1), finally when all constraints are added (Cannot-Must-Labeled line in Table 1). Our results demonstrate that the combination of pairwise constraints and labeled data constraints improved the performance of the semi-clustering algorithm with tolerable time consumption. As expected, the more constraints are added, the better are the performance.

4.3 Real Data Sets

The LPECM algorithm has been tested on three known data sets from the UCI Machine Learning Repository namely Iris, Glass, and Wdbc and a derived Letters data set from UCI. Table 2 indicates for each data set its number of objects, its number of attributes and its number of classes.

For each data set, we randomly created 5%, 8%, and 10% of each type of constraints out of the whole objects, leading to a total of 15%, 24%, and 30% of constraints. As an example, Fig. 4 shows the hard credal partition obtained with the Iris data set after executing the LPECM algorithm with a Mahalanobis distance and 24% of constraints in total. As can be observed, all the constrained objects are clustered with certainty in a singleton. Ellipses represent the covariance matrices obtained for each cluster.

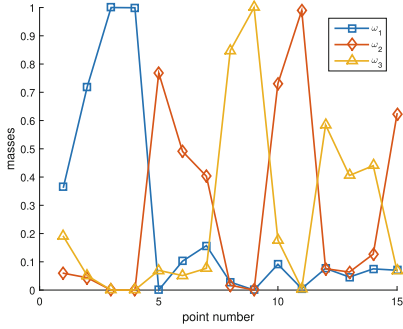


Fig. 3. Mass curve obtained on Toy data set with the LPECM algorithm

Table 1. Performance obtained on toy data set with the LPECM algorithm

	ARI	Time(s)
ECM	0.60	0.07
LPECM-Cannot	0.68	0.41
LPECM-C-Must	0.85	0.29
LPECM-C-M-labeled	1.00	0.22

Tables 3 and 4 illustrate for all data sets the accuracy results with a Euclidean and a Mahalanobis distance respectively when the different percentage of constraints are employed. Mean and standard deviation are calculated over 20 simulations. As it can be observed, incorporating constraints lead most of the time to significant improvement of the clustering solution. Using a Mahalanobis distance particularly help to achieve better accuracy than using a Euclidean distance. Indeed, the Mahalanobis distance corresponds to an adaptive metric giving more freedom than a Euclidean distance to respect the constraints while finding a coherent data structure.

Table 2. Description of the data sets from UCIMLR

Name	Objects	Attributes	Clusters
Iris	150	4	3
Letters	227	16	3
Wdbc	569	31	2
Glass	214	10	3

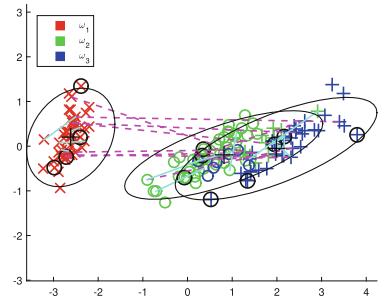


Fig. 4. Hard credal partition obtained on Iris data set with the LPECM algorithm

For the time consumption, as it can be observed from Fig. 5, (1) Adding constraints gives higher computation time than no constraints. (2) most of the time, the more constraints are added, the less time is needed to finish the computation.

Table 3. LPECM’s performance (ARI) with Euclidean distance

	ECM	LPECM		
		5.00%	8.00%	10.00%
Iris	0.59 ± 0.00	0.70 ± 0.01	0.71 ± 0.00	0.70 ± 0.01
Letters	0.04 ± 0.01	0.09 ± 0.03	0.09 ± 0.04	0.10 ± 0.02
Wdbc	0.67 ± 0.00	0.71 ± 0.00	0.71 ± 0.01	0.71 ± 0.00
Glass	0.59 ± 0.07	0.60 ± 0.07	0.62 ± 0.06	0.65 ± 0.08

Table 4. LPECM’s performance (ARI) with Mahalanobis distance

	ECM	LPECM		
		5.00%	8.00%	10.00%
Iris	0.67 ± 0.01	0.71 ± 0.05	0.82 ± 0.01	0.83 ± 0.04
Letters	0.08 ± 0.01	0.45 ± 0.03	0.47 ± 0.02	0.60 ± 0.05
Wdbc	0.73 ± 0.02	0.74 ± 0.03	0.75 ± 0.02	0.77 ± 0.05
Glass	0.56 ± 0.03	0.60 ± 0.03	0.65 ± 0.02	0.65 ± 0.03

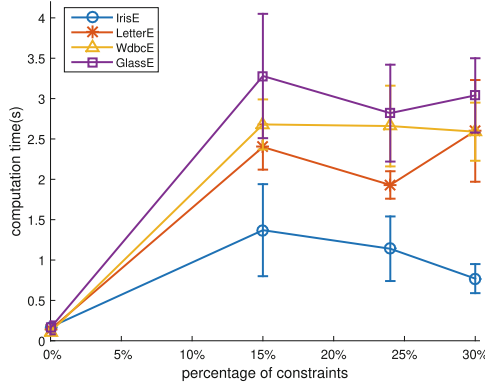


Fig. 5. Time consumption (CPU) of the LPECM algorithm with Euclidean distance

5 Conclusion

In this paper, we introduced a new algorithm named Labeled and Pairwise constraints Evidential C-Means (LPECM). It generates a credal partition and mixes three main types of instance-level constraints together, allowing us to retrieve more constraints from the background knowledge than other semi-supervised clustering algorithms. In addition, the framework of belief function employed in our algorithm allows us (1) to represent doubts for the labeled data constraints (2) to clearly express, with the credal partition as a result,

the uncertainties about the class memberships of the objects. Experiments show that the LPECM algorithm does obtain better accuracy with the introduction of constraints, particularly with a Mahalanobis distance. Further investigations have to be performed to fine-tune parameters and to study the influence of the constraints on the clustering solution. The LPECM algorithm can also be applied for a real application to show the interest in gathering various types of constraints. In this framework, active learning schemes, which automatically retrieve few informative constraints with the help of an expert, are interesting to study. Finally, in order to scale and fast the LPECM algorithm, a new minimization process can be developed by relaxing some optimization constraints.

References

1. Antoine, V., Quost, B., Masson, M.H., Denœux, T.: CEVCLUS: evidential clustering with instance-level constraints for relational data. *Soft Comput. - Fusion Found. Methodol. Appl.* **18**(7), 1321–1335 (2014)
2. Antoine, V., Gravouil, K., Labroche, N.: On evidential clustering with partial supervision. In: Destercke, S., Denœux, T., Cuzzolin, F., Martin, A. (eds.) *BELIEF 2018*. LNCS (LNAI), vol. 11069, pp. 14–21. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99383-6_3
3. Antoine, V., Labroche, N., Vu, V.V.: Evidential seed-based semi-supervised clustering. In: *International Symposium on Soft Computing & Intelligent Systems*, Kitakyushu, Japan (2014)
4. Antoine, V., Quost, B., Masson, M.H., Denœux, T.: CECM: constrained evidential C-means algorithm. *Comput. Stat. Data Anal.* **56**(4), 894–914 (2012)
5. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the fuzzy C-means clustering algorithm. *Comput. Geosci.* **10**(2–3), 191–203 (1984)
6. Bilenko, M., Basu, S., Mooney, R.: Integrating constraints and metric learning in semi-supervised clustering. In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ACM New York, NY, USA (2004)
7. Coleman, T.F., Li, Y.: A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM J. Optim.* **6**(4), 1040–1058 (1996)
8. Denœux, T.: Evidential clustering of large dissimilarity data. *Knowl.-Based Syst.* **106**(C), 179–195 (2016)
9. Dua, D., Graff, C.: UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
10. Grira, N., Crucianu, M., Boujemaa, N.: Active semi-supervised fuzzy clustering. *Pattern Recogn.* **41**(5), 1834–1844 (2008)
11. Gustafson, D.E., Kessel, W.C.: Fuzzy clustering with a fuzzy covariance matrix. In: *IEEE Conference on Decision & Control Including the Symposium on Adaptive Processes*, New Orleans, LA (2007)
12. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**(1), 193–218 (1985)
13. Li, F., Li, S., Denœux, T.: k-CEVCLUS: constrained evidential clustering of large dissimilarity data. *Knowl.-Based Syst.* **142**, 29–44 (2018)
14. Masson, M.H., Denœux, T.: ECM: an evidential version of the fuzzy C-means algorithm. *Pattern Recogn.* **41**(4), 1384–1397 (2008)
15. Shafer, G.: *A Mathematical Theory of Evidence*, vol. 42. Princeton University Press, Princeton (1976)

16. Smets, P., Kennes, R.: The transferable belief model. *Artif. Intell.* **66**, 191–234 (1994)
17. Vu, V.V., Do, H.Q., Dang, V.T., Do, N.T.: An efficient density-based clustering with side information and active learning: a case study for facial expression recognition task. *Intell. Data Anal.* **23**(1), 227–240 (2019)
18. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, Williamstown, MA, USA, vol. 1, pp. 577–584 (2001)
19. Zhang, H., Lu, J.: Semi-supervised fuzzy clustering: a kernel-based approach. *Knowl.-Based Syst.* **22**(6), 477–481 (2009)