E2GK : Evidential evolving Gustafsson-Kessel algorithm for data streams partitioning using belief functions.

Lisa Serir, Emmanuel Ramasso, Noureddine Zerhouni

E2GK: Evidential Evolving Gustafsson-Kessel Algorithm For Data Streams Partitioning Using Belief Functions

Lisa Serir, Emmanuel Ramasso, and Noureddine Zerhouni

FEMTO-ST Institute, UMR CNRS 6174 - UFC / ENSMM / UTBM,
Automatic Control and Micro-Mechatronic Systems Dep., 25000, Besançon, France
(lisa.serir, emmanuel.ramasso, noureddine.zerhouni)@femto-st.fr

Abstract. A new online clustering method, called E2GK (Evidential Evolving Gustafson-Kessel) is introduced in the theoretical framework of belief functions. The algorithm enables an online partitioning of data streams based on two existing and efficient algorithms: Evidantial c-Means (ECM) and Evolving Gustafson-Kessel (EGK). E2GK uses the concept of credal partition of ECM and adapts EGK, offering a better interpretation of the data structure. Experiments with synthetic data sets show good performances of the proposed algorithm compared to the original online procedure.

1 Introduction

Given a set of N data points, clustering refers to a wide variety of algorithms that aim at discovering c groups (clusters) $\omega_1, ..., \omega_c$ whose members are similar in some way. The purpose is to summarize the data or to verify an existing structure of the data. In most cases, a cluster is defined as a subset of data for which the similarity between data within this subset is larger than the similarity with the data in other subsets. In many cases, the Euclidean distance between data is used as a dissimilarity measure.

A wide variety of clustering methods has been developed. The most commonly used methods are divided into two main categories: hierarchical and non-hierarchical methods. Among the latter, the K-means algorithm [4] is the most commonly used. The idea of K-means algorithm is to randomly create K clusters and to assign each data point to the closest one in an iterative way, reallocating points until a convergence criterion is satisfied.

Using hard partitioning methods, data are grouped in an exclusive way, i.e., data can't belong to two (or more) different clusters. In fuzzy partitioning, each data can belong to more than one cluster with different membership degrees. The most popular fuzzy partitioning method is Bezdek's Fuzzy C-means (FCM) algorithm [3]. One can also mention the Gustafson-Kessel fuzzy clustering algorithm [10] that is capable of detecting hyper-ellipsoidal clusters of different sizes and orientations by adjusting the covariance matrix of data.

Another concept of partition, introduced in [7], is the *credal* partition based on belief functions theory. A credal partition extends the existing concepts of hard, fuzzy (probabilistic) and possibilistic partition by allocating, for each data, a *mass of belief*, not only to single clusters, but also to any subset of $\Omega = \{\omega_1, ..., \omega_c\}$. This particular representation allows coding all the situations, from certainty to total ignorance of membership to clusters. In the Evidential c-Means (ECM) algorithm [13], the credal partition is in particular exploited for outliers detection.

Online clustering is an important problem that frequently arises in many fields, such as pattern recognition and machine learning [8]. Numerous techniques have been developed for clustering data in a static environment [4]. However, in many real-life applications, non-stationary data (i.e., with time-varying parameters) are commonly encountered. The task of online clustering is to group incoming data into clusters in a temporal sequence. Also called *incremental clustering* in machine learning [11], online clustering, is generally unsupervised and has to manage recursive training in order to incorporate new information gradually and to take into account model evolutions over time.

In this paper, we propose the Evidential Evolving Gustafson Kessel algorithm (E2GK) which permits to adapt a credal partition matrix as data gradually arrive. This clustering algorithm is introduced in the theoretical framework of belief functions, and more precisely of Smets' Transferable Belief Model (TBM, [14]). E2GK is composed of two main steps, both performed online:

1. Determination of clusters' prototypes (also called centers), either by moving existing prototypes or by creating new ones. To do so, we use some results from the Evolving Gustafson-Kessel algorithm (EGK) proposed in [9].
2. Allocation of the belief masses to the different subsets of classes. This step is based on some results of the Evidential c-means algorithm (ECM) [13].

E2GK benefits from two efficient algorithms: EGK and ECM, by dealing with - in an online manner - doubt between clusters and outliers. Doubt is generally encountered in data transition and can be useful to limit the number of clusters in the final partition. Moreover, outliers are well managed using the conflict degree explicitly emphasized in the TBM framework.

In Section 2, we present GK and ECM algorithms as well as some tools of the theory of belief functions giving the necessary background for Section 3 in which we introduce E2GK. Some results are finally presented in Section 4.

2 Background

Let the data be in the form of a collection $\{x_1, \ldots, x_k, \ldots, x_N\}$ of feature vectors $x_k \in \Re^q$, and c the number of clusters, each of them characterized by a prototype (or a center) $v_i \in \Re^q$.

2.1 Gustafson-Kessel Algorithm

Clustering algorithms based on an optimization process aim at minimizing a suitable fuction J that represents the fitting error of the clusters regarding the data:

$$J(V,U) = \sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^{\beta}d_{ik}^{2} \ , \tag{1}$$

where

- u_{ik} is the membership degree of point k to the i-th prototype (cluster center),
- $U = [u_{ij}]$ is the resulting partition matrix with dimension $c \times N$,
- $V = [v_i]$ is the $c \times q$ matrix of prototypes,
- d_{ik} is the distance between the k-th data point x_k and the i-th prototype,
- Paramater $\beta > 1$ is a weighting exponent that controls the fuzziness of the partition (it determines how much clusters may overlap).

The distance d_{ik} used in the GK algorithm is a squared inner-product distance norm (Mahalanobis) that depends on a positive definite symmetric matrix A_i defined by:

$$d_{ik}^{2} = \|x_k - v_i\|_{A_i}^{2} = (x_k - v_i)A_i(x_k - v_i)^{T} \ . \tag{2}$$

This adaptive distance norm is unique for each cluster as the norm inducing matrix A_i, $i = 1...c$, is calculated by estimates of the data covariance

$$A_i = [\rho_i det(F_i)]^{1/q} F_i^{-1} \ , \tag{3}$$

where ρ_i is the cluster volume of the i-th cluster and F_i is the fuzzy covariance matrix calculated as follows:

$$F_i = \frac{\sum_{k=1}^{N}(u_{ik})^{\beta}(x_k - v_i)^{T}(x_k - v_i)}{\sum_{k=1}^{N}(u_{ik})^{\beta}} \ . \tag{4}$$

The objective function is minimized using an iterative algorithm, which alternatively optimizes the cluster centers and the membership degrees:

$$v_i = \frac{\sum_{k=1}^{N}(u_{ik})^{\beta}x_k}{\sum_{k=1}^{N}(u_{ik})^{\beta}}, \ i = 1 \cdots c, \ k = 1 \cdots N \ , \tag{5}$$

and

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}(d_{ik}/d_{jk})^{2/\beta - 1}}, \ i = 1 \cdots c, \ k = 1 \cdots N \ . \tag{6}$$

The GK algorithm has the great advantage to adapt the clusters according to their real shape.

2.2 Belief Functions and Credal partition

Dempster-Shafer theory of evidence, also called belief functions theory, is a theoretical framework for reasoning with partial and unreliable information. It was first introduced by A. P. Dempster (1968), then developed by G. Shafer (1976). Later, Ph. Smets proposed a general framework, the *Transferable Belief Model* (TBM) [14], for uncertainty representation and combination of various pieces of information without additional priors.

Considering a variable ω taking values in a finite set called the *frame of discernment* Ω, the *belief* of an agent in subsets of Ω can be represented by a *basic belief assignment* (BBA), also called *belief mass assignment*:

$$\begin{aligned} m : 2^{\Omega} \rightarrow & \quad [0,1] \\ A & \mapsto m(A) \ , \end{aligned} \tag{7}$$

with $\sum_{A \subseteq \Omega} m(A) = 1$. A belief mass can not only be assigned to a singleton ($|A| = 1$), but also to a *subset* ($|A| > 1$) of variables *without any assumption concerning additivity*. This property permits the explicit modeling of doubt and conflict, and constitutes a fundamental difference with probability theory. The subsets A of Ω such that $m(A) > 0$, are called the *focal elements* of m. Each focal element A is a set of possible values of ω. The quantity $m(A)$ represents a fraction of a unit mass of belief allocated to A. Complete ignorance corresponds to $m(\Omega) = 1$, whereas perfect knowledge of the value of ω is represented by the allocation of the whole mass of belief to a unique singleton of Ω, and m is then said to be *certain*. In the case of all focal elements being singletons, m boils down to a probability function and is said to be *bayesian*.

A positive value of $m(\emptyset)$ is considered if one accepts the *open-world assumption* stating that the set Ω might not be complete, and thus ω might take its values outside Ω. This value represents the *degree of conflict* and is then interpreted as a mass of belief given to the hypothesis that ω might not lie in Ω. This interpretation is useful in clustering for outliers detection [13].

Belief functions theory is largely used in clustering and classification problems [6, 12]. Recently (2003) was proposed the use of belief functions for cluster analysis. Similar to the concept of fuzzy partition but more general, the concept of *Credal Partition* was introduced. It particularly permits a better interpretation of the data structure. A credal partition is constructed by assigning a BBA to each possible subset of clusters. Partial knowledge regarding the membership of a datum i to a class j is represented by a BBA m_{ij} on the set $\Omega = \{\omega_1, \ldots, \omega_c\}$. This particular representation makes it possible to code all situations, from certainty to total ignorance.

Example 1. Considering $N = 4$ data and $c = 3$ classes, Tab. 1 gives an example of a credal partition. BBAs for each datum in Tab. 1 illustrate various situations: datum 1 certainly belongs to class 1, whereas the class of datum 2 is completely unknown. Partial knowledge is represented for datum 3. As $m_4(\emptyset) = 1$, datum 4 is considered as an outlier, i.e., its class does not lie in Ω.

Table 1. Example of a credal partition.

A	$\emptyset$	ω_1	ω_2	$\{\omega_1,\omega_2\}$	ω_3	$\{\omega_1,\omega_3\}$	$\{\omega_2,\omega_3\}$	$\{\omega_1,\omega_2,\omega_3\}$
$m_1(A)$	0	1	0	0	0	0	0	0
$m_2(A)$	0	0	0	0	0	0	0	1
$m_3(A)$	0	0	0	0	0.2	0.5	0	0.3
$m_4(A)$	1	0	0	0	0	0	0	0

2.3 ECM: Evidential C-Means algorithm

Our approach for developing E2GK (Evidential Evolving GK algorithm) is based on the concept of credal partition as described in ECM [13] where the objective function was defined as:

$$J_{ECM}(M,V) = \sum_{k=1}^{N} \sum_{\{i/A_i \neq \emptyset, A_i \subseteq \Omega\}} |A_i|^{\alpha} \, m_{ki}^{\beta} d_{ki}^2 + \sum_{k=1}^{N} \delta^2 m_k(\emptyset)^{\beta} \ , \tag{8}$$

subject to

$$\sum_{\{i/A_i \neq \emptyset, A_i \subseteq \Omega\}} m_{ki} + m_k(\emptyset) = 1 \quad \forall k = 1, \ldots, N \ , \tag{9}$$

where:

- α is used to penalize the subsets of Ω with high cardinality,
- $\beta > 1$ is a weighting exponent that controls the fuzziness of the partition,
- d_{ki} denotes the Euclidean distance between datum k and prototype v_i,
- δ controls the amount of data considered as outliers.

The $N \times 2^c$ partition matrix M is derived by determining, for each datum k, the BBAs $m_{ki} = m_k(A_i)$, $A_i \subseteq \Omega$ such that m_{ki} is low (resp. high) when the distance d_{ki} between datum k and focal element A_i is high (resp. low). The matrix M is computed by the minimization of criterion (8) and was shown to be [13], $\forall k = 1 \ldots N$, $\forall i / A_i \subseteq \Omega$, $A_i \neq \emptyset$:

$$m_{ki} = \frac{|A_i|^{-\alpha/(\beta-1)} \, d_{ki}^{-2/(\beta-1)}}{\sum\limits_{A_l \neq \emptyset} |A_l|^{-\alpha/(\beta-1)} \, d_{kl}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}} \ , \tag{10}$$

and $m_k(\emptyset) = 1 - \sum_{A_i \neq \emptyset} m_{ki}$. The distance between a datum and any non empty subset $A_i \subseteq \Omega$ is then defined by computing the center of each subset A_i. The latter is the barycenter $\overline{v_i}$ of the clusters' centers (obtained by minimizing criterion (8)) composing A_i.

3 Deriving E2GK

GK algorithm [10] has the great advantage to adapt the clusters according to their real shape. The resulting clusters are hyper-ellipsoids with arbitrary orientation and are well suited for a variety of practical problems. However, GK is not

able to deal with streams of data (relies on an iterative optimization scheme). Moreover, it assumes that the number of clusters is known in advance.

In [9], an online version of GK clustering algorithm (EGK) was developed to enable online partitioning of data streams based on a similar principle to the one used in the initial GK algorithm [10]. In particular, online updating of the fuzzy partition matrix relies on the same formula (6). Rules were then proposed to decide whether a new cluster has to be created or existing prototypes should evolve.

3.1 E2GK: Evidential Evolving Gustafsson-Kessel algorithm

The adaptation of the EGK algorithm to belief functions is introduced in this section. The E2GK algorithm is presented in Tab. 2. It relies on some parts developed in [9] and the proposed adaptations are emphasized in bold characters.

Step 1 – Initialization: At least one cluster's center should be provided. Otherwise, the first point is chosen as the first prototype. If more than one prototype is assumed in the initial data, GK or ECM algorithm can be applied to identify an initial partition matrix. The result of the initialization phase is a set of c prototypes v_i and a covariance matrix[1] F_i.

Step 2 – Decision making: The boundary of each cluster is defined by the cluster radius r_i, defined as the *medium* distance between the cluster center v_i and the points belonging to this cluster with membership degrees larger or equal to a given threshold u_h:

$$r_i = \underset{\forall x_j \in \, i\text{-}th \text{ } cluster \text{ } and \text{ } P_{ji} > u_h}{\text{median}} \|v_i - x_j\|_{A_i} \quad . \tag{11}$$

where P_{ij} is the confidence degree that point j belongs to $\omega_i \in \Omega$ and can be obtained by three main processes: either by using the belief mass $m_j(\omega_i)$, or the pignistic transformation [14] that converts a BBA into a probability distribution, or by using the plausibility transform [5]. We propose here to choose the pignistic transformation. The *median* value is used (instead of the *maximum* rule in EGK) to reduce the sensitivity to extreme values. Moreover, the minimum membership degree u_h - initially introduced in [9] and requiring to decide whether a data point belongs or not to a cluster - can be difficult to assess. It may depend on the density of the data as well as on the level of cluster overlapping. We rather set u_h automatically to $1/c$ in order to reduce the number of parameters while ensuring a natural choice for its value.

Step 3 – Computing the partition matrix: Starting from the resulting set of clusters at a given iteration, we build the partition matrix M (10) using the Mahalanobis distance (2)(3). We assumed that each cluster volume $\rho_i = 1$ as in standard GK algorithm.

[1] To obtain a covariance matrix from ECM, one can also use the Mahalanobis distance as proposed in [1].

Step 4 – Adapting the structure: Given a new data point x_k, two cases are considered:

Case 1: x_k belongs to an existing cluster, thus a clusters' update has to be performed. Data point x_k is assigned to the closest cluster p if $d_{pk} \leq r_p$. Then, the p-th cluster is updated:

$$v_{p,new} = v_{p,old} + \theta \cdot (x_k - v_{p,old}) \quad , \tag{12}$$

and

$$F_{p,new} = F_{p,old} + \theta \cdot \left((x_k - v_{p,old})^T (x_k - v_{p,old}) - F_{p,old} \right) \quad , \tag{13}$$

where θ is a learning rate, $v_{p,new}$ and $v_{p,old}$ denote respectively the new and old values of the center, and $F_{p,new}$ and $F_{p,old}$ denote respectively the new and old values of the covariance matrix.

Case 2: x_k is not within the boundary of any existing cluster (i.e. $d_{pk} > r_p$), thus a new cluster may be defined and a clusters' update has to be performed. The number of clusters is thus incremented: $c = c + 1$. Then, the incoming data x_k is accepted as the center v_{new} of the new cluster and its covariance matrix F_{new} is initialized with the covariance matrix of the closest cluster $F_{p,old}$.

In the initial EGK algorithm [9], a parameter P_i was introduced to assess the number of points belonging to the i-th cluster. The authors suggested a threshold parameter P_{tol} to guarantee the validity of the covariance matrices and to improve the robustness. This (context-determined) parameter corresponds to the desired minimal amount of points falling within the boundary of each cluster. The new created cluster is then rejected if it contains less than P_{tol} data points.

After creating a new cluster, the data structure evolves. However, the new cluster may contain data points previously assigned to another cluster. Thus, the number of data points in previous clusters could change. We propose an additional step to verify, after the creation of a new cluster, that all clusters have at least the required minimum amount of data points (P_{tol} or more). If not, the cluster with the lowest number of points is deleted. Therefore, compared to the initial EGK algorithm, in which the number of clusters only increases, E2GK is more flexible because the structure can change either by increasing or decreasing the number of clusters.

The overall algorithm is presented in Tab. 2 where the proposed adaptation appears in bold.

4 Application of E2GK

To illustrate the ability of the proposed algorithm, let consider the following synthetic data randomly generated from five different bivariate gaussian distributions with parameters as given in Tab. 3.

Initial clusters (Fig. 1) of $N = 15$ data points each, of type G_1 and G_2, were identified by batch GK procedure with $u_h = 0.5$, $P_{tol} = 20$ and $\theta = 0.1$. To test the updating procedure, we gradually (one point at a time) added the following data points (in this given order): 1) 15 data points of type G_1, 2) 15 data points

Table 2. E2GK algorithm

Initialization	1. Take the first point as a center or apply the off-line GK or ECM algorithm to get the initial number of clusters c and the corresponding centers V and covariances F_i, $i = 1 \cdots c$ 2. **Calculate $\overline{v_j}$, the barycenter of the clusters' centers** composing $A_j \subseteq \Omega$ 3. **Calculate the credal partition M, using (10)**
Updating	*Repeat* for each new data point x_k 4. Find the closest cluster p 5. Decision-making: Calculate the radius r_p of the closest cluster using (11) with the **median value** *If* $d_{pk} \leq r_p$ 6. Update the center v_p (12) 7. Update the covariance matrix F_p (13) *else* 8. Create a new cluster: $v_{c+1} := x_k$; $F_{c+1} := F_p$ *end* 9. **Recalculate the credal partition M using (10)** 10. **Check the new structure:** remove the cluster with the minimum number of data points if less than P_{tol}

Table 3. Parameters of the synthetic data

type	μ	σ
G_1	[0 5]	0.3
G_2	[0 0]	0.3
G_3	[6 6]	0.6
G_4	[6 0]	0.6
noise	[2.5 2.5]	2

of type G_2, 3) 15 data points of type G_3, 4) 30 data points of type G_4, 5) 15 data points of type G_3, 6) 90 data points of type "noise", 7) 6 data points at the following positions: [10.1 3.2], [10.1 −3.2], [−4.1 −3.1], [−2.3 8.3], [8.6 −3.1] and [6.2 9.2]. E2GK parameters were set to: $P_{tol} = 20$, $\theta = 0.1$, $\delta = 10$, $\alpha = 1$ and $\beta = 2$.

Each new incoming data point leads to a new credal partition. Figure 2 shows the final resulting partition. The center of gravity of each cluster is marked by a big star (the notation ω_{ij} stands for $\{\omega_i, \omega_j\}$). A data point falling in a subset ω_{ij} means that this point could either belong to ω_1 or ω_2. The points represented in circles are those with the highest mass given to the empty set and considered as outliers. It can be seen that a meaningful partition is recovered and that outliers are correctly detected.

The online adaptation of the clusters is illustrated in Figure 3. One can see how E2GK assigns each new data point to the desired cluster or subset. The figure depicts the evolution of the partition regarding the order of arrival of the data (like mentionned before). The first 30 points are used to initialize clusters ω_1 and ω_2. Then, from $t = 31$ to 45 points are assigned by E2GK to cluster ω_2.

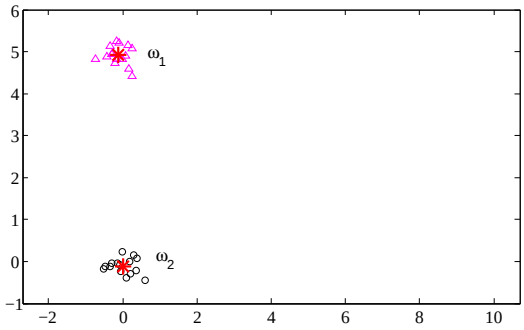

Fig. 1. Initialization of E2GK algorithm using some data from two clusters. Centers are represented by stars.

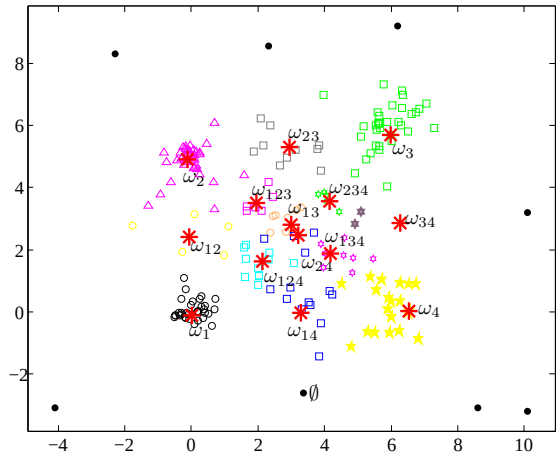

Fig. 2. Credal partition with $\delta = 10$, $\alpha = 1$, $\beta = 2$, $\theta = 0.1$, $P_{tol} = 20$. Big stars represent centers. We also displayed the centers corresponding to subsets, e.g. ω_{123}, and atypical data (dots) are well detected.

The next 15 points are assigned to ω_1 then to ω_4, ω_3 (30 points) and to ω_4. The next points correspond to noise and are mainly assigned to subsets, for example point 160 to ω_{134}.

Figure 4 also depicts the structure evolution, that is the number of clusters at each instant. The scenario given at the begining of this section is recovered: at $t = 76$ data from group G_3 arrive but still, not enough data are available to create clustrs while a cluster is created at $t = 93$ and $t = 110$ for group G_4 and G_3 respectively. "Noise" and atypical points arriving from $t = 181$ to $t = 211$

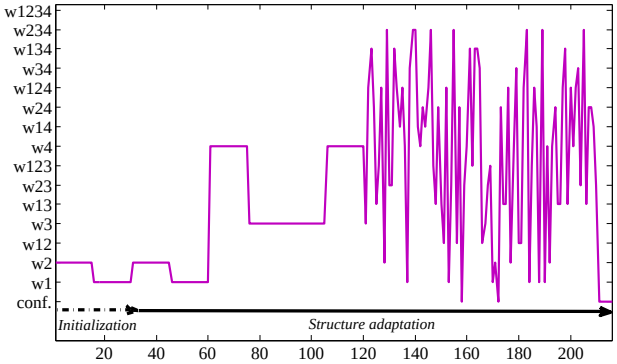

Fig. 3. Structure adaptation: a datum arrives at each instant (x-axis) and is assigned to one of all possible subsets (y-axis). The set of possible subsets also evolves with the number of clusters.

do not affect the structure. This figure does not illustrate clusters' removing because this operation is made within the algorithm.

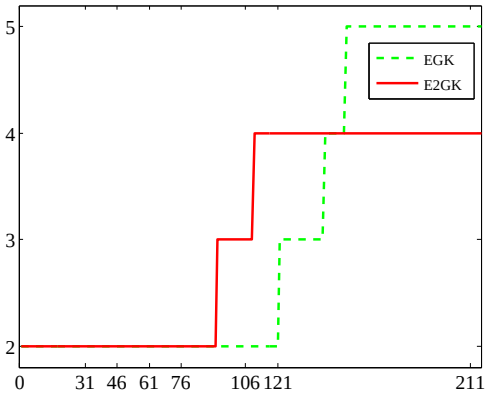

Fig. 4. Structure evolution: the number of clusters at each instant varies as data arrive.

Figure 5 describes the dataset partitioning after decision making by applying the pignistic transformation [14] on the final credal partition matrix. Datatips provide the center coordinates, which are close to the real parameters (Tab 3). In comparison, we also provide in Figure 6 the centers obtained by EGK algorithm with parameters $P_{tol} = 20$, u_h=1/c and $\theta = 0.1$ (the same as in E2GK).

5 Conclusion

To our knowledge, only one *incremental* approach to clustering using belief functions has been proposed [2]. However, in this approach the number of clusters

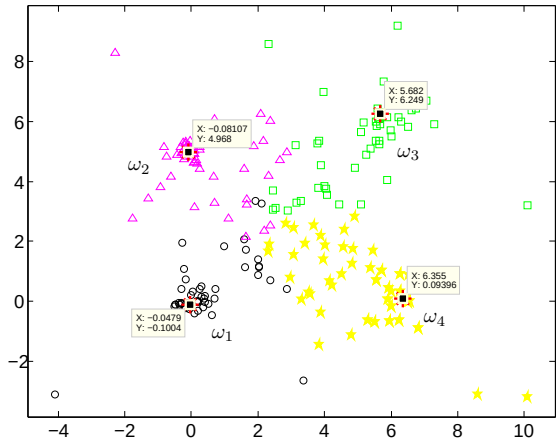

Fig. 5. Decision on clusters for each point based on the pignistic probabilities obtained from the credal partition (Fig. 2) using E2GK algorithm. Also are displayed the coordinates of the centers found by E2GK.

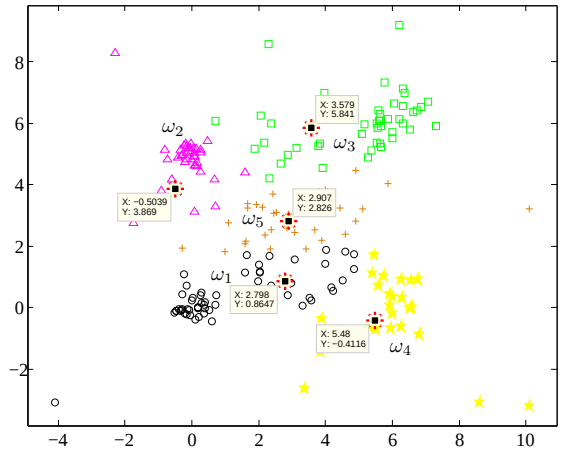

Fig. 6. Decision on clusters for each point based on the maximum degree of membership from the fuzzy partition using GK algorithm. Also are displayed the coordinates of the centers found by EGK. The parameter u_h was set to $1/c$ and the other parameters are the same as in E2GK ($\theta = 0.1$ and $P_{tol} = 20$).

is known in advance so this is not adapted for online applications. Moreover, data are described by a given number of attributes, each labeled by a mass of belief provided by an expert. This prior information is generally not available in pattern recognition problems.

E2GK algorithm, described in this paper, is an evolving clustering algorithm using belief functions theory, which relies on the credal partition concept. This type of partition allows a finer representation of datasets by emphasizing doubt

between clusters as well as outliers. Doubt is important for data streams analysis from real systems because it offers a suitable representation of gradual changes in the stream. E2GK relies on some parts of EGK algorithm [9], initially based on a fuzzy partition, to which we bring some modifications:

- using the median operator to calculate cluster radius (vs. max. for EGK),
- using the credal partitioning (vs. fuzzy for EGK),
- changing the partitionning structure by adding or removing clusters (vs. adding only in EGK).

Simulation results show that E2GK discovers relatively well the changes in the data structure. A thorough analysis of parameters' sensitivity (P_{tol} and θ) is now required to properly and automatically set them.

References

1. Antoine, V., Quost, B., Masson, M.H., Denoeux, T.: Cecm - adding pairwise constraints to evidential clustering. In: IEEE World Congress on Computational Intelligence (July 2010)
2. Ben-Hariz, S., Elouedi, Z.: IK-BKM: An incremental clustering approach based on intra-cluster distance. In: Eighth ACS/IEEE International Conference on Computer Systems and Applications (2010)
3. Bezdek, J.C.: Pattern Recognition with fuzzy objective function algorithms. Plenum Press, New York (1981)
4. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press (1995)
5. Cobb, B., Shenoy, P.: On the plausibility transformation method for translating belief function models to probability models. International journal of approximate reasoning 41(3), 314–330 (2006)
6. Denoeux, T.: A k-nearest neighbor classification rule based on dempster-shafer theory. IEEE Trans. on Systems, Man and Cybernetics 25(5), 804–813 (1995)
7. Denoeux, T., Masson, M.H.: Evclus: Evidential clustering of proximity data. IEEE Transactions on Systems, Man and Cybernetics Part B 34(1), 95 – 109 (2004)
8. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley, New york (2001)
9. Georgieva, O., Filev, D.: Gustafson-kessel algorithm for evolving data stream clustering. In: International Conference on Computer Systems and Technologies (2009)
10. Gustafson, E., Kessel, W.: Fuzzy clustering with a fuzzy covariance matrix. In: IEEE Conference on Decision and Control (1978)
11. Janichen, S., Perner, P.: Acquisition of concept description by conceptual clustering. Machine Learning and Data Mining in Pattern Recognition 3587, 153 – 163 (2005)
12. Kim, H., Swain., P.H.: Evidential reasoning approach to multisource-data classification in remote sensing. IEEE Transactions on Systems, Man and Cybernetics 25(8), 1257 – 1265 (1995)
13. Masson, M.H., Denoeux, T.: ECM: An evidential version of the fuzzy c-means algorithm. Pattern Recognition 41(4), 1384 – 1397 (2008)
14. Smets, P., Kennes, R.: The transferable belief model. Artificial Intelligence 66, 191–234 (1994)