HAL open science

Evidential Evolving Gustafson-Kessel Algorithm For Online Data Streams Partitioning Using Belief Function Theory.

Lisa Serir, Emmanuel Ramasso, Noureddine Zerhouni

To cite this version:

Lisa Serir, Emmanuel Ramasso, Noureddine Zerhouni. Evidential Evolving Gustafson-Kessel Algorithm For Online Data Streams Partitioning Using Belief Function Theory.. International Journal of Approximate Reasoning, 2012, 53 (5), pp.747-768. 10.1016/j.ijar.2012.01.009 . hal-00719620

HAL Id: hal-00719620

https://hal.science/hal-00719620v1

Submitted on 20 Jul 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Evidential Evolving Gustafson-Kessel Algorithm For Online Data Streams Partitioning Using Belief Function Theory

Lisa Serir[a,*], Emmanuel Ramasso[a,**], Noureddine Zerhouni[a,]

[a] *FEMTO-ST Institute, UMR CNRS 6174 - UFC / ENSMM / UTBM, Automatic Control and Micro-Mechatronic Systems Dep., 24 rue Alain Savary, 25000, Besançon, France*

Abstract

A new online clustering method called E2GK (Evidential Evolving Gustafson-Kessel) is introduced. This partitional clustering algorithm is based on the concept of credal partition defined in the theoretical framework of belief functions. A credal partition is derived online by applying an algorithm resulting from the adaptation of the Evolving Gustafson-Kessel (EGK) algorithm. Online partitioning of data streams is then possible with a meaningful interpretation of the data structure. A comparative study with the original online procedure shows that E2GK outperforms EGK on different entry data sets. To show the performance of E2GK, several experiments have been conducted on synthetic data sets as well as on data collected from a real application problem. A study of parameters' sensitivity is also carried out and solutions are proposed to limit complexity issues.

Keywords: Belief functions, Clustering, Evolving Systems

1. Introduction

Clustering is an unsupervised learning technique that aims at discovering meaningful groups called clusters within a set of patterns (observations, data items, or feature vectors), based on similarity [1]. Intuitively, patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster.

Pattern proximity, or similarity, is usually measured by a distance function defined on pairs of patterns. A variety of distance measures is used in the various communities, reflecting implicit assumptions about cluster shape [2, 3, 4]. For instance, the Euclidean distance is often used to reflect dissimilarity between two patterns and is known to work well when all clusters are spheroids or when all clusters are well separated. The variety

*Corresponding author
**Principal corresponding author
Email addresses: lisa.serir@femto-st.fr (Lisa Serir), emmanuel.ramasso@femto-st.fr (Emmanuel Ramasso), noureddine.zerhouni@ens2m.fr (Noureddine Zerhouni)

of techniques for representing data, measuring proximity between data elements, and grouping data elements has produced a large number of clustering methods [5, 6].

Several research communities are interested in clustering techniques and use different terminologies and assumptions for the clustering process. Clustering is appropriate in situations where little prior information is available about data, as it explores relationships between data points and makes assessment of their general structure.

Among the range of existing approaches, there is a distinction between hierarchical and partitional clustering depending on the properties of the generated clusters. Hierarchical methods produce a nested series of partitions based on a criterion for merging or splitting clusters based on similarity. Partitional clustering algorithms identify the partition that optimizes a clustering criterion. Additional techniques for the grouping operation include probabilistic [7] and graph-theoretic [8] clustering methods. The clustering algorithm presented in this paper is in line with partitional clustering methods.

1.1. Partitional Clustering

A partitional clustering algorithm obtains a single partition of the data instead of a clustering structure, like in a hierarchical technique. Partitional methods have advantages in applications involving large data sets. A typical problem encountered in partitional algorithms is the choice of the number of desired output clusters. The partitional techniques usually produce clusters by optimizing a criterion function. The most intuitive and frequently used criterion function is the squared error, which tends to work well with isolated and compact clusters. The k-means is the simplest and most commonly used algorithm employing this criterion [9].

The output clustering can be either hard (a partition of the data into groups) or fuzzy, where each pattern has a variable degree of membership in each of the output clusters. Traditional clustering approaches generate partitions, in which each pattern exclusively belongs to one cluster. Hence, the clusters in a hard clustering are disjoint. Fuzzy clustering extends this notion to associate each pattern with every cluster using a membership function [10]. A fuzzy clustering method assigns degrees of membership in several clusters to each input pattern. A fuzzy clustering can be converted to a hard clustering by assigning each pattern to the cluster with the largest measure of membership. The most popular fuzzy partitioning method is Bezdek's Fuzzy c-means (FCM) algorithm [11]. One can also mention the Gustafson-Kessel fuzzy clustering algorithm [12] that is capable of detecting hyper-ellipsoidal clusters of different sizes and orientations by adjusting the covariance matrix of data.

1.2. Evolution of the partitional clustering methods

Clustering methods progressed considerably toward more realistic approaches of clusters' detection and separation. The membership restriction to exactly one cluster of hard clustering was overcome by fuzzy probabilistic clustering [11]. However, fuzzy clustering may be inaccurate in a noisy environment producing non intuitive high membership degrees for outliers. To improve this weakness, Krishnapuram and Keller [13] proposed a possibilistic approach to clustering that they called a possibilistic c-means (PCM) clustering. More recently, a more flexible concept based on belief functions

theory, called *credal partition*, was introduced in [14] as an extension of the existing concepts of hard, fuzzy and possibilistic partitions. In this approach, each data point is given a *mass of belief* regarding its membership to, not only to a single cluster, but also to any subset of clusters. This particular representation permits coding all the situations, from certainty to total ignorance of membership to clusters. In the Evidential c-Means (ECM) algorithm [15], the credal partition is in particular exploited for outliers detection. The experiments presented in [14, 15] show that meaningful and robust summaries of the data can be achieved, as it is possible to compute, for each cluster, a set of objects that surely belong to it, and a larger set of objects that possibly belong to it. Robustness is achieved by assigning outliers to the empty set.

1.3. Online clustering

Numerous techniques have been developed for clustering data in a static environment [1]. In classical data analysis it is usually assumed that a data set is first collected completely and then the analysis is carried out. However, it is very common that we do not have a fixed data set, but a constantly growing amount of data coming in. In many real-life applications, non-stationary data (i.e. with time-varying parameters) are commonly encountered. A possible way to analyze such data is to restart the corresponding algorithm completely, each time new data arrive. However, this approach is neither very efficient, nor suited to detect changes in the data. Online clustering is an important problem that frequently arises in many fields, such as pattern recognition and machine learning [16].

The task of online clustering is to group data into clusters as long as they arrive in a temporal sequence. Also called *incremental clustering* in machine learning [17], or sometimes *adaptive clustering*, online clustering is generally unsupervised and has to manage recursive training in order to gradually incorporate new information and to take into account model evolutions over time. The goal of online methods is to avoid storage of complete datasets by discarding each data point once it has been used. Online methods are required when: 1) it is necessary to respond in real time; 2) the input data set may be so huge that batch methods become impractical because of computation time or memory requirement; and 3) the input data come as continuous streams of unlimited length that make it impossible to apply batch methods.

Adaptive versions of the mentioned clustering methods were proposed for data processing in online mode. In [18], the authors proposed an online k-means algorithm for data streams clustering. the key aspect of their method is a preprocessing step, which includes an incremental computation of the distance between data streams, using a discrete Fourier transform (DFT) approximation of the original data. They also propose an incremental adaptation of the number of clusters k. At each iteration, the current number of clusters can increase or decrease by one according to an evaluation of the current clustering structure by a separation index quality measure.

An online version of the spherical k-means algorithm has been proposed in [19] based on the Winner-Take-All competitive learning. In this online algorithm, each cluster centroid is incrementally updated. The number of clusters K is fixed in advance and the method was proposed as a more efficient version than in the batch version giving better clustering results.

Examples of clustering algorithms that are used in model identification procedures include Angelov's work on fuzzy rule-based models identification [20, 21]. In [22], a recursive approach for the adaptation of a fuzzy rule-based model structure has been developed and tested using on-line clustering of the input-output data with a recursively calculated spatial proximity measure. Centers of these clusters are then used as prototypes of the centers of the fuzzy rules (as their focal points). The recursive nature of the algorithm makes it possible to design an evolving fuzzy rule-base in on-line mode, which adapts to the variations of the data pattern.

Various online extensions of fuzzy clustering algorithms have been developed during the last decade [23, 24, 25, 26]. Our particular interest goes to an online version of the Gustafson-Kessel fuzzy clustering algorithm (EGK), that was proposed in [27] and enables online partitioning of data streams based on a similar principle than the one used in the initial GK algorithm [12]. In particular, online updating of the fuzzy partition matrix relies on the same formula. Rules were then proposed to decide whether a new cluster has to be created or existing prototypes should evolve.

Finally, To our knowledge, only one *incremental* approach to clustering using belief functions has been proposed [28]. In this approach the data in the training set are considered uncertain. Moreover, each data is described by a given number of attributes, each labeled by a mass of belief provided by an expert, which are very difficult to obtain in real life application. In addition, this approach assumes that the number of clusters is known in advance and can not evolve.

1.4. Contribution

In this paper, we propose the Evidential Evolving Gustafson Kessel algorithm (E2GK) which permits to adapt a credal partition matrix as data arrive. This clustering method is introduced in the theoretical framework of belief functions, and more precisely of Smets' Transferable Belief Model (TBM, [29]). E2GK is composed of two main steps, both performed online:

1. Determination of clusters prototypes (also called centers), either by moving existing prototypes, creating new ones, or by removing existing ones. To do so, we adapt some results from the Evolving Gustafson-Kessel algorithm (EGK) proposed in [27].
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) [15].

E2GK benefits from two efficient algorithms: EGK and ECM, by dealing with doubt between clusters and outliers in an online manner. 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.

This paper is an extended version of a previous contribution [30]. In Section 2, we present GK and ECM algorithms as well as belief functions giving the necessary background for Section 3 in which we introduce E2GK. Results are finally presented in Section 7.

2. Background

Let the patterns to be clustered be in the form of a collection $X = \{x_1, \ldots, x_k, \ldots, x_N\}$ of feature vectors $x_k \in \Re^q$. Let c be the number of clusters characterized by a prototype (or a center) denoted $v_i \in \Re^q$.

2.1. Gustafson-Kessel Algorithm

Most of the prototype-based fuzzy clustering algorithms, like FCM, are based on an optimization scheme and aim at minimizing a suitable function 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 data point x_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,
- Parameter $\beta > 1$ is a weighting exponent that controls the fuzziness of the partition (it determines how much clusters may overlap).

The FCM algorithm uses point prototypes (cluster centers) v_i and the inner-product norm-induced metric given by:

$$d_{ik}^2 = \|x_k - v_i\|_S^2 = (x_k - v_i)^T S (x_k - v_i) \ , \tag{2}$$

as the distance measure, where the norm matrix S is a positive definite symmetric matrix. The above distance measure is meaningful only when all clusters are expected to be ellipsoids of the same orientation and size, as the norm matrix determines the size and shape of the points enclosed within a given distance of the center. The norm matrix is hard to specify in advance, thus it is usually taken to be the identity matrix and doing so reduces the distance measure to the Euclidean distance. The Euclidean distance gives good results only when all clusters are spheroids of the same size or when all clusters are well separated. To overcome the drawback due to the Euclidean distance one can use the Mahalanobis distance (MD) as the distance measure.

The Gustafson-Kessel algorithm (GK) associates each cluster with both a point and a matrix, respectively representing the cluster center and its covariance. While the original fuzzy c-means makes the implicit hypothesis that clusters are spherical, the Gustafson-Kessel algorithm can identify ellipsoidal clusters. Gustafson and Kessel [12] extended the standard fuzzy c-means algorithm by employing an adaptive distance norm, in order to detect clusters of different geometrical shapes in the data set. Each cluster has its own norm-inducing matrix S_i. In the GK and EGK algorithms, the fuzzy

covariance matrix F_i of the i-th cluster is used. 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 S_i defined by:

$$d_{ik}^2 = \|x_k - v_i\|_{S_i}^2 = (x_k - v_i)^T S_i (x_k - v_i) \quad . \tag{3}$$

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

$$S_i = [\rho_i \det(F_i)]^{1/q} F_i^{-1} \quad , \tag{4}$$

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

$$F_i = \frac{\sum_{k=1}^{N} (u_{ik})^\beta (x_k - v_i)(x_k - v_i)^T}{\sum_{k=1}^{N} (u_{ik})^\beta} \quad . \tag{5}$$

The minimization of the objective function $J(V,U)$ under the constraint $\sum_{i=1}^{c} u_{ik} = 1$, using an iterative algorithm, which alternatively optimizes the cluster centers and the membership degrees:

$$v_i = \frac{\sum_{k=1}^{N} (u_{ik})^\beta x_k}{\sum_{k=1}^{N} (u_{ik})^\beta}, \; i = 1 \ldots c, \; k = 1 \ldots N \quad . \tag{6}$$

and

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} (d_{ik}/d_{jk})^{2/\beta - 1}}, \; i = 1 \ldots c, \; k = 1 \ldots N \quad . \tag{7}$$

2.2. Belief Functions

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) [29], for uncertainty representation and combination of various pieces of information without additional priors. In this section, we present the main concepts oh this theory.

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*:

$$\begin{array}{rccc} m: & 2^\Omega & \rightarrow & [0,1] \\ & A & \mapsto & m(A) \end{array} \quad , \tag{8}$$

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 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 BBA m is said to be *normal* if $m(\emptyset) = 0$. A normalized BBA is such as:

$$m_*(A) = \begin{cases} \dfrac{m(A)}{1 - m(\emptyset)} \text{ if } A \neq \emptyset \\ 0 \text{ otherwise} \end{cases} \tag{9}$$

This process is called Dempster *normalization*. 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 value outside Ω. The conflict 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 [15].

Several functions - in *one-to-one correspondence* [29] - can be computed from a BBA. Among these functions, the *plausibility* function is defined by:

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B) \quad \forall A \subseteq \Omega \ , \tag{10}$$

where $pl(A)$ represents the maximal degree of belief supporting the subset A . It is important to note that pl boils down to a probability measure when m is a Bayesian BBA and to a possibility measure when the focal elements are nested [31]. Probability and possibility measures are thus recovered as special cases of belief functions.

Decision making in the TBM framework consists in the choice of the best hypothesis using the *pignistic probability* distribution [29] defined as:

$$\mathbf{BetP}(\omega) = \sum_{\omega \in A} \frac{m(A)}{|A|}, \quad \forall \omega \in \Omega \ . \tag{11}$$

where each mass of belief $m(A)$ is equally distributed among the elements of A. If the BBA is subnormal ($m(\emptyset) \neq 0$), a preliminary normalization step has to be performed (Eq. 9).

2.3. ECM: Evidential C-Means algorithm

Belief functions theory is largely used in clustering and classification problems [32, 33]. Recently (2003), T. Denoeux and M-H. Masson 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 A of Ω, and not only to singletons of Ω. Partial knowledge regarding the membership of a datum k to a class i is represented by a BBA m_{ik} on the set $\Omega = \{\omega_1, \ldots, \omega_c\}$. This particular representation makes possible the coding of all situations, from certainty to total ignorance. A $N \times 2^c$ partition matrix M is derived by

determining, for each data point k, the BBA's $m_{ik} = m_k(A_i)$, $A_i \subseteq \Omega$ such that m_{ik} is low (resp. high) when the distance d_{ik} between datum k and focal element A_i is high (resp. low).

The particularity of ECM is that only singletons focal elements (clusters ω_k) are associated with centroids. However, the distance between a data point and any non empty subset $A_i \subseteq \Omega$ is defined by computing the center of each subset A_i, the latter being the barycenter $\overline{v_i}$ of clusters centers composing A_i:

$$\overline{v_i} = \frac{1}{|A_i|} \sum_{l=1}^{c} s_{li}\, v_i, \tag{12}$$

with

$$s_{li} = \begin{cases} 1 \text{ if } \omega_l \in A_i, \\ 0 \text{ otherwise} \end{cases} \tag{13}$$

The distance d_{ik} between x_k and the focal set A_i may then be defined by:

$$d_{ik}^2 = \|x_k - \overline{v_i}\|^2 . \tag{14}$$

Therefore, ECM is not a FCM with 2^c classes.

ECM uses the classical Euclidean distance. Classes are thus supposed to be spherical. However, the use of a Mahalanobis distance may be interesting in case of elliptical clusters. A variant of ECM with an adaptive metric was proposed in [34]. To compute the distance between a data point x_k and a non singleton subset of clusters of Ω, A_i, the authors proposed to associate the matrix $\overline{S_i}$ with subset A_i by averaging the matrices associated to the classes $\omega_j \in A_i$:

$$\overline{S_i} = \frac{1}{|A_i|} \sum_{l=1}^{c} s_{li}\, S_l,\ \forall A_i \subseteq \Omega,\ A_i \neq \emptyset, \tag{15}$$

where, S_l denotes a $(q \times q)$ matrix associated to cluster ω_l, $l = 1 \cdots c$ inducing a norm $\|x\|_{S_l}^2 = x^T\, S_l\, x$. Matrix $\overline{S_i}$ may be seen as a kind of within-class covariance matrix of the clusters composing A_i [34]. The distance d_{ik}^2 between x_k and any bubset $A_i \neq \emptyset$, is then defined by:

$$d_{ik}^2 = \|x_k - \overline{v_i}\|_{\overline{S_i}}^2 = (x_k - \overline{v_i})^T\, \overline{S_i}\, (x_k - \overline{v_i}) \ . \tag{16}$$

The Evidential c-means algorithm (ECM) [15] is a clustering algorithm that uses a FCM-like algorithm to derive a credal partition by minimizing a suitable objective function. Our approach for developing E2GK (Evidential Evolving GK algorithm) is based on the concept of credal partition as described in ECM [15] where the objective function was defined as:

$$J_{ECM}(M,V) = \sum_{k=1}^{N} \sum_{\{i/A_i \neq \emptyset, A_i \subseteq \Omega\}} |A_i|^{\alpha}\, m_{ik}^{\beta} d_{ik}^2 + \sum_{k=1}^{N} \delta^2 m_k(\emptyset)^{\beta} \ , \tag{17}$$

subject to

$$\sum_{\{i/A_i \neq \emptyset, A_i \subseteq \Omega\}} m_{ik} + m_k(\emptyset) = 1 \quad \forall k = 1, \ldots, N \ , \tag{18}$$

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_{ik} denotes the Euclidean distance between data point k and subset A_i,
- δ controls the amount of data considered as outliers.

The matrix M is computed by the minimization of criterion (Eq. 17) and was shown to be [15], $\forall k = 1 \ldots N$, $\forall i / A_i \subseteq \Omega$, $A_i \neq \emptyset$:

$$m_{ik} = \frac{|A_i|^{-\alpha/(\beta-1)} \, d_{ik}^{-2/(\beta-1)}}{\sum\limits_{A_l \neq \emptyset} |A_l|^{-\alpha/(\beta-1)} \, d_{lk}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}} \quad , \tag{19}$$

and

$$m_k(\emptyset) = 1 - \sum_{A_i \neq \emptyset} m_{ik} \quad . \tag{20}$$

To use the adaptive distance (Eq. 16), the authors of [34] demonstrated that the covariance matrix, called Σ_l, associated to each cluster ω_l is obtained by minimizing (Eq. 17) with respect to S_l and can be seen as an analog in the evidential framework of the fuzzy covariance matrix. The expression of Σ_l is then given by[1] [34]:

$$\Sigma_l = \sum_{k=1}^{N} \sum_{A_i \ni \omega_l} (m_{ik})^2 \, |A_i|^{\alpha-1} \, (x_k - \overline{v_i})(x_k - \overline{v_i})^T \, , \; l = 1 \cdots c \quad . \tag{21}$$

From the credal partition, the classical clustering structures (possibilistic, fuzzy and hard partitions) can be recovered [15]. A possibilistic partition can be obtained by computing from each bba m_k for data point k, the plausibilities pl_k of the different clusters (Eq. 10). For example, $pl_k(\{\omega_i\})$ is the plausibility (or possibility) that object k belongs to cluster i. One can also obtain a probabilistic fuzzy partition by calculating the pignistic probability $\mathrm{BetP}_k(\{\omega_i\})$ (Eq. 11) induced by each bba m_k and interpreting this value as the degree of membership of the object k to cluster i. Assigning each object to the cluster with highest pignistic probability, or with highest plausibility, permits to obtain a hard partition.

One can also summarize the data by assigning each object to the set of clusters with the highest mass. One then obtains a partition of the points in at most 2^c groups, where each group corresponds to a set of clusters. This makes it possible to find the points that unambiguously belong to one cluster, and the points that lie at the boundary of two or more clusters. Moreover, points with high mass on the empty set may optionally be rejected as outliers.

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

[1] For optimization requirements, the authors set parameter β equal to 2 [34].

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

3. The Evidential Evolving Gustafson-Kessel algorithm

In [27], an online version of the Gustafson-Kessel clustering algorithm (EGK) was proposed. It enables online partitioning of data streams by adapting the fuzzy partition matrix defined in the original GK (Eq. 7). Data arrive in a temporal sequence and for each new incoming data point, rules are applied to decide whether a new cluster has to be created or existing prototypes must evolve.

In this section we propose an adaptation of this algorithm to the context of belief functions. The main idea is to derive, online, a credal partition matrix from the data. As shown in ECM [15], a credal partition offers a better interpretation of the data structure. In particular, doubt between clusters is useful for modeling transitions between clusters and high degrees of conflict generally reflect atypical data. The proposed algorithm is presented in Tab. 2.

3.1. Initialization

At least one cluster 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 algorithm or the adaptive distance version of ECM (section 2.3) can be applied to identify an initial partition matrix. The result of the initialization phase is a set of c prototypes v_l and a covariance matrix[2] F_l for GK (Eq. 5), and Σ_l for ECM (Eq. 21).

3.2. The updating procedure

For each new incoming data point x_k, the following steps are performed.

3.2.1. Determination of the existing clusters

The boundary of each cluster is defined by the cluster radius r_l, defined as the *median* distance between the cluster center v_l and the points belonging to this cluster with membership degree larger or equal to a given threshold u_h:

$$r_l = \underset{\forall x_k \in \, l\text{-th cluster and } P_{lk} > u_h}{\text{median}} \|v_l - x_k\|_{S_l} \quad . \tag{22}$$

[2]To obtain a covariance matrix from ECM, one can also use the Mahalanobis distance as proposed in [34].

where P_{lk} is the confidence degree that point k belongs to $\omega_l \in \Omega$ and can be obtained by three main process: either by using the belief mass $m_k(\omega_l)$, or the pignistic transformation [29] that converts a BBA into a probability distribution, or by using the plausibility transform (10) [35]. In section 7, BetP was used in most of the presented applications of E2GK.

Compared to EGK, where the *maximum* rule is used, we here apply the *median* value which is less sensitive to extreme values. Moreover, the minimum membership degree u_h, initially introduced in [27] and required 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.

3.2.2. Computation of the partition matrix

Starting from the resulting set of clusters at a given iteration, we need to build the partition matrix M as in ECM (Eq. 19).

In section 5, solutions are proposed to decrease memory consumption due to the storage of the partition matrix.

3.2.3. Adaptation of the structure

Given a new data point x_k, two cases are considered:

Case 1: x_k belongs to an existing cluster, thus an update of clusters 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 by applying the Kohonen rule [36]:

$$v_{p,new} = v_{p,old} + \theta \cdot \Delta \ , \tag{23}$$

where

$$\Delta = x_k - v_{p,old} \ , \tag{24}$$

and

$$\Sigma_{p,new} = \Sigma_{p,old} + \theta \cdot \left(\Delta \Delta^{\mathrm{T}} - \Sigma_{p,old}\right) \ , \tag{25}$$

where θ is a learning rate (and can be set in $[0.05, 0.3]$ as in [27]), $v_{p,new}$ and $v_{p,old}$ denote respectively the new and old values of the center, and $\Sigma_{p,new}$ and $\Sigma_{p,old}$ denote respectively the new and old values of the covariance matrix.

In [27], the authors propose to recursively update the inverse of the covariance matrix and its determinant to improve the computational efficiency of the algorithm using an exponentially weighted moving average (EWMA) procedure. The resulting expressions are given in the following:

$$\Sigma_{p,new}^{-1} = \left(I - G_p \Delta^T\right) \Sigma_{p,old}^{-1} \frac{1}{1-\theta}, \tag{26}$$

where

$$G_p = \Sigma_{p,old}^{-1} \Delta \frac{\theta}{1 - \theta + \theta \Delta^T \Sigma_{p,old}^{-1} \Delta}, \tag{27}$$

and

$$\det\left(\Sigma_{p,new}\right) = (1-\theta)^{n-1}\det\left(\Sigma_{p,old}\right)\left(1-\theta+\theta\Delta^T\Sigma_{p,old}^{-1}\Delta\right), \tag{28}$$

with $0 < \theta < 1$ is the constant forgetting factor that controls the updating rate. This formulation requires that the covariance matrix is initialized as a diagonal matrix with sufficiently large elements [27].

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. In EGK [27], the covariance matrix Σ_{new} of this new cluster is initialized with the covariance matrix of the closest cluster $\Sigma_{p,old}$. To avoid introducing too much prior information related to previously created clusters, and to reduce the effect of these latter on the shape and orientation of the newly created ones, we propose to initialize the covariance matrix of each new cluster as a diagonal matrix corresponding to a reduced set of data points of circular shape. This matrix is then filled by Eq. 25.

In the initial EGK algorithm [27], a parameter P_i was introduced to assess the number of points belonging to the i-th cluster in order to quantify the credibility of the estimated clusters. 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.

An additional step is proposed in our adaptation of EGK. 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 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 it is not the case, then the cluster with the minimum number of points is removed. Compared to the initial EGK algorithm, in which the number of clusters only increases, E2GK is more flexible because the structure can change by increasing *or decreasing* the number of clusters.

The overall algorithm is summarized in Tab. 2.

4. An Example on Synthetic Data

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 $u_h = 0.5$, $P_{\text{tol}} = 20$ 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 of type G_2,

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 Σ_l, $l = 1 \cdots c$ 2. **Calculate $\overline{v_i}$, the barycenter of the cluster centers** composing $A_i \subseteq \Omega$ 3. **Calculate the credal partition M, using Eq. 19 and Eq. 20**
Updating	*Repeat* for every new data point x_k 4. Find the closest cluster p 5. Decision-making: Calculate the radius r_p of the closest cluster using Eq. 22 with the **median value** *If* $d_{pk} \leq r_p$ 6. Update the center v_p (Eq. 23) 7. Update the covariance matrix Σ_p (Eq. 26 and Eq. 28) *else* 8. Create a new cluster: $v_{c+1} := x_k$; $\Sigma_{c+1} := \Sigma_p$ *end* 9. **Recalculate the credal partition M using Eq. 19 and Eq. 20** 10. **Check the new structure: remove all the clusters with a number of data points $\leq P_{\mathbf{tol}}$**

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

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 location: $[10.1\ \ 3.2]$, $[10.1\ \ -3.2]$, $[-4.1\ \ -3.1]$, $[-2.3\ \ 8.3]$, $[6.2\ \ 9.2]$ and $[8.6\ \ -3.1]$.

E2GK parameters were set to: $P_{\text{tol}} = 20$, $\theta = 0.1$, $\delta = 10$, $\alpha = 1$ 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 red 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 black circles are those with the highest mass given to the empty set and considered as outliers. It can be

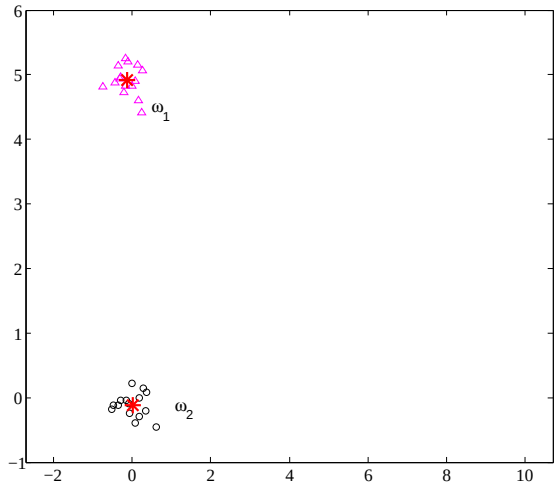

Figure 1: Initialization of E2GK algorithm using some data from two clusters. Centers are represented by stars.

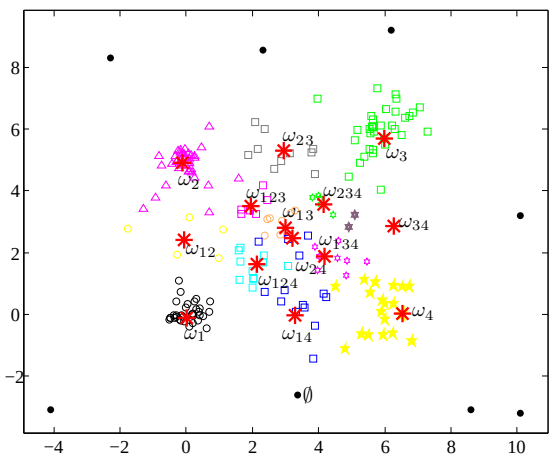

Figure 2: Credal partition with $\delta = 10$, $\alpha = 1$, $\beta = 2$, $\theta = 0.1$, $P_{\text{tol}} = 20$. Red big stars represent centers. We also displayed the centers corresponding to subsets, e.g. ω_{123}, and atypical data (black dots) are well detected.

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 cluster ω_1 and ω_2. Then from $t = 31$ to 45 points are assigned by E2GK to cluster ω_2. 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

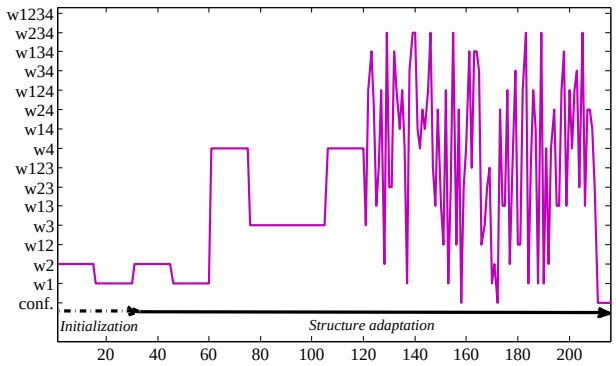

Figure 3: Structure adaptation: data arrived at each instant (x-axis) and is assigned to one of all possible subsets (y-axis). The set of possible subsets also evolves with the number of clusters.

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 begin of this section is recovered: at $t = 76$ data from group G_3 arrives but they are not enough to allow the creation of clusters 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 $t = 181$ and $t = 211$ do not affect the structure. This figure does not illustrate clusters' removing because this operation is made within the algorithm.

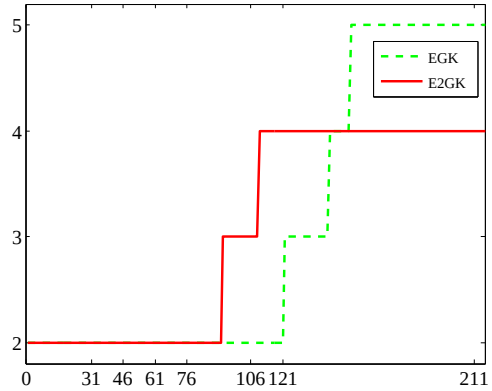

Figure 4: Structure evolution: the number of clusters at each instant varies as data arrive.

Figure 5 describes the dataset partitioning after decision making by applying the pignistic transformation (11) on the final credal partition matrix. Data tips 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 $P_{\text{tol}} = 20$, u_h=1/c and $\theta = 0.1$ (the same as in E2GK). EGK generates on this dataset too many (five) centers which are also misplaced.

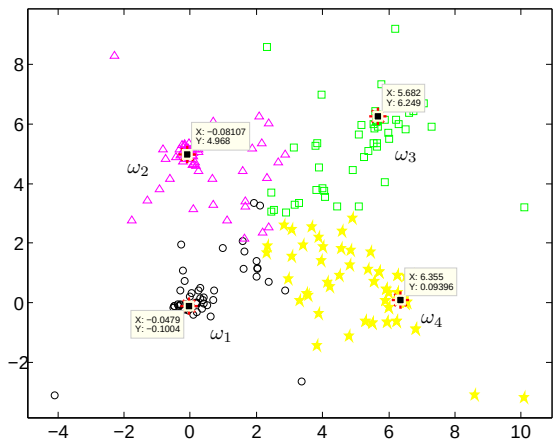

Figure 5: Decision on clusters for each point based on the pignistic probabilities obtained from the credal partition (Fig. 2) using E2GK algorithm. Also are displayed the coordinates of the centers found by E2GK.

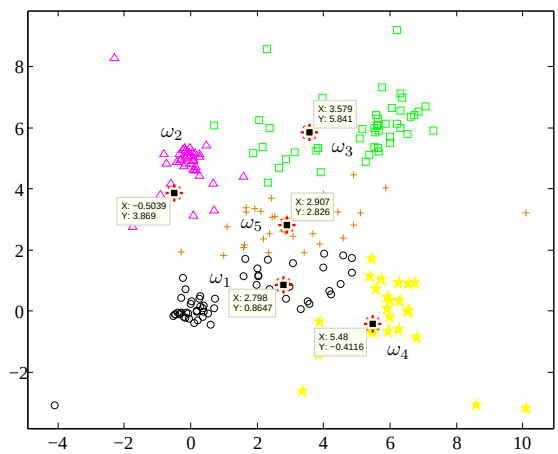

Figure 6: Decision on clusters for each point based on the maximum of degree of membership from the fuzzy partition using GK algorithm. Also are displayed the coordinates of the centers found by EGK. The parameter u_h was set to $1/c$ and the other parameters are the same as in E2GK ($\theta = 0.1$ and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

5. Limiting the Complexity

As mentioned in section 3, one has to consider complexity issues due to the computation of the credal partition. Indeed, for each data point in the partition matrix, the belief mass potentially has $2^{|\Omega|}$ elements. Storing a belief mass for all data points may require a lot of memory resources (and becomes intractable for $|\Omega| \geq 12$). In [15, 14], the authors proposed to reduce the complexity of the method by considering only a subclass of BBA's with a limited number of focal sets. They proposed for instance, to limit the focal sets to be either Ω, or to be composed of at most two classes. Follow-

ing a similar idea, we propose to use the concept of k-additive belief function in order to decrease memory consumption. We also propose solutions to improve speed and memory consumption.

5.1. k-additive partition matrix

In the context of discrete fuzzy measures (also called non-additive measures or capacities), M. Grabisch proposed the concept of k-additive fuzzy measure in order to face with complexity issues [37]. Considering the fact that a belief function is a particular fuzzy measure for which the Möbius Transform is non-negative, we give the following definitions:

Definition 1. A *belief function* on Ω is a function $Bel : 2^{\Omega} \rightarrow [0,1]$ generated by a BBA m as follows:

$$Bel(A) = \sum_{B \subseteq A} m(B) \quad \forall A \subseteq \Omega \ , \tag{29}$$

Note that m is actually the Möbius transform of Bel. We will refer to the BBA inducing a k-additive Bel as a k-additive belief mass (or BBA).

Definition 2. The belief mass is said to be k-additive if the cardinality of focal sets is less or equal to k, i.e. $m(A) = 0$ if $|A| > k$ and thus it exists at least one element $A \subset \Omega$ containing k elements with $m(A) > 0$.

The k-additive belief mass is thus here an approximation of a belief mass and parameter k sets the complexity. In section 7, we give an illustration of the influence of parameter k.

5.2. Improving speed and memory consumption

This approximation facilitates decision-making concerning the belonging to a cluster of each data point. Indeed, if one uses the pignistic probability distribution (10) or the plausibility transform (11) on singletons, then the number of computations is reduced since the belief mass has less elements.

For each data point, one needs to compute the belief mass of each element A (one by one) with cardinality greater or equal to k and for each one of them has to:

- For the pignistic probability: transfer the mass $\frac{m(A)}{|A|}$ on $m(\omega_i)$ if $\omega_i \in A$;
- For the plausibility transform: transfer the mass $m(A)$ on $m(\omega_i)$ if $\omega_i \cap A \neq \emptyset$.

Once all subsets have been treated for the current point, one can decide which cluster it belongs to. Therefore, it is not required to store the partition matrix, which drastically reduces time and memory consumption.

5.3. Re-interpreting the partition matrix

In some applications, the partition matrix has to be stored for further analysis of the data structure. In this case, the end-user can set the value of k:

- For $k = 1$, only masses on singletons are computed. In this case, one obtains a probabilistic partition matrix (after normalisation).
- For $k = |\Omega|$, masses on all subsets are computed. In this general case, one obtains an evidential partition matrix as in ECM [15].
- For $k \in]1, |\Omega|[$, one can compute the plausibility of each cluster to obtain a possibilistic partition matrix. An other option consists in computing masses on subsets with cardinality less or equal to k to obtain a *k-additive partition matrix*.

Considering the example presented in section 4, E2GK was applied on the same dataset, with the same parameters. Figures 8, 9, 10 and 11 shows the influence of parameter k on the final results of the algorithm. The decision on clusters is obtained by applying the pignistic transformation on each final k-additive credal matrix. For each value of k, data tips show the coordinates of the centers discovered by E2GK. One can see that the number of clusters increases when k is small. For example, for $k = 1$ i.e., only singletons are considered for the computation of the credal partition, E2GK finds 6 clusters. This can be justified by the fact that, in this case, the credal partition is actually a probabilistic partition. Doubt between clusters has not been taken into account leading to the creation of clusters within data of type "noise". When $k = 2$, five clusters are found by E2GK. Here, only doubt between two clusters is considered. Finally, a value of $k = 3$ leads to a partitioning of the dataset into 4 clusters, as in the general case ($k = \Omega$). However one can notice a slight difference for the center values, closer to real values for the general case. The Mahalanobis distance enables E2GK to adapt the clusters to their real shape. The shape of the clusters found by E2GK is illustrated in Fig. 7. This figure shows that the clusters shape correspond to the ground truth. Indeed, the clusters whose variance is small (left) are clearly identified despite the presence of clusters with much higher variance (right). Another important result is the form of the cluster of noisy data (center) that appears spherical reflecting the fact that the noisy points are almost evenly distributed throughout the baseline.

6. Performance and Parameter Sensitivity

Cluster validity methods aim at giving the experts some quantitative tools to evaluate the quality of the resulting partitioning. There exist numerous methods in the literature to discuss issues in cluster analysis [38, 39]. A fundamental issue faced in clustering is to decide the optimal number of clusters that best fits the data set. An improper choice of the clustering algorithm parameters leads to a resulting partitioning that is not optimal for the specific data set. It is very common that visualization of the clustering results on a 2D data set experiments is used to verify how meaningful the provided partitioning is. However, this verification can be a very difficult task in the case of large multidimensional data sets. Research efforts have been made to address the problem of the evaluation of the clustering results, and the reader can refer to a

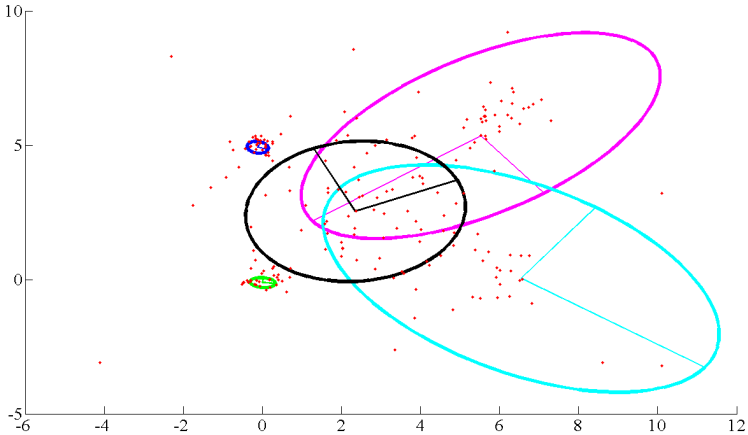

Figure 7: Shape of clusters found for $k = 2$

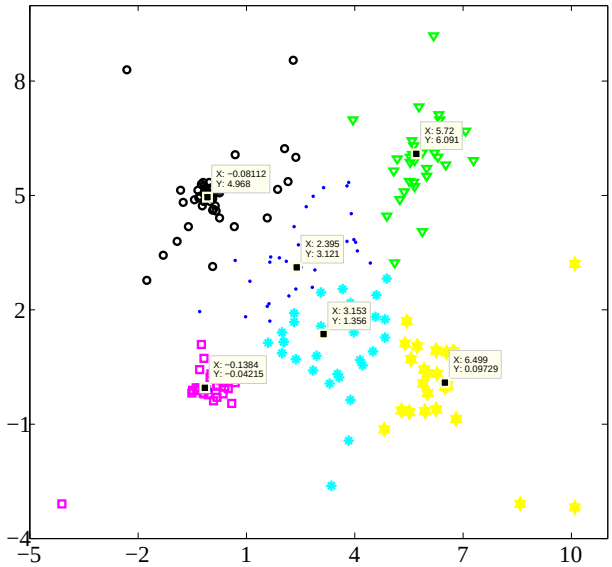

Figure 8: Final decision on clusters when $k = 1$.

review of these methods in [40, 41]. Among the various validity indices that exist in the literature, the external validity indices are used to evaluate the results of the clustering algorithm based on a predefined structure on a data set that reflects the intuition of the user. The resulting clustering structure $C = \{C_1, \ldots, C_m\}$ is thus compared to an

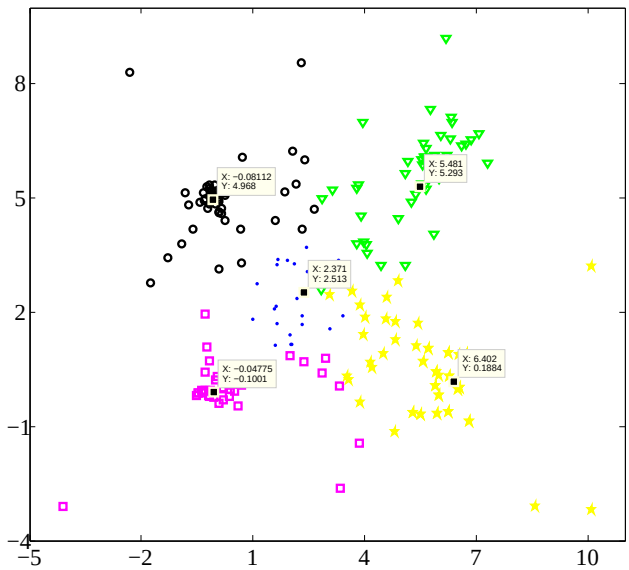

Figure 9: Final decision on clusters when $k = 2$.

independent partition $P = \{P_1, \ldots, P_l\}$ of the same data set built according to the user's intuition. Examples of these indices are the *Rand index* [42], the *Folkes and Mallows index*, *Hubert's* Γ *statistic*, etc. [40].

In the present paper, we consider the *Jaccard coefficient* [43] defined as follows:

$$J = \frac{a}{a+b+c} \tag{30}$$

where:

- a: the number of pairs of points for which both points belong to the same cluster of the clustering structure C and to the same group of partition P.
- b: the number of pairs of points for which the points belong to the same cluster of C and to different groups of P.
- c: the number of pairs of points for which the points belong to the same cluster of P and to different groups of C.

The Jaccard coefficient has been commonly applied to assess the similarity between different partitions of the same dataset, the level of agreement between a set of class

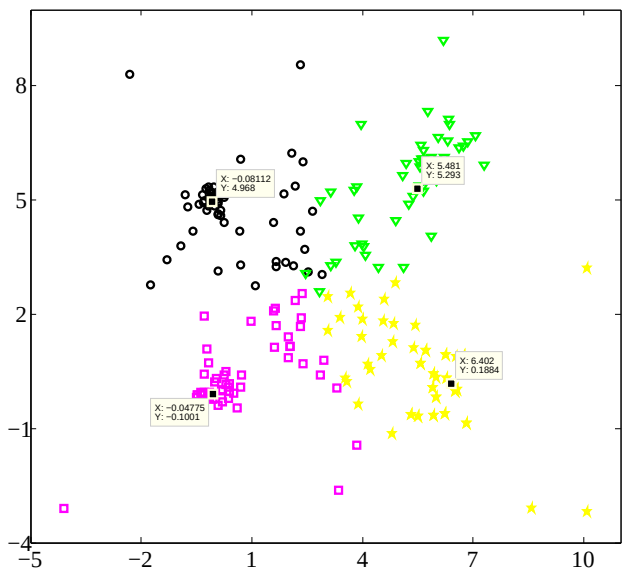

Figure 10: Final decision on clusters when $k = 3$.

labels P and a clustering result C is determined by the number of pairs of points assigned to the same cluster in both partitions. It produces a result in the range $[0,1]$, where a value of 1 means that C and P are identical. The higher the value of J is, the more similar C and P are.

To show the performance of E2GK, we consider the example of section 4. We recall that the dataset is composed of four groups of data points generated by bivariate Gaussian distributions, in addition with a fifth group representing noisy data, whose parameters are given in Tab. 3. To compute the Jaccard coefficient J, the four groups of data points represent the partition P to be compared with the results of the clustering algorithm (ground truth). For the sake of comparison, J is calculated for both EGK and E2GK algorithms, with parameters $u_h = 0.5$, $P_{\text{tol}} = 20$ and $\theta = 0.1$ for EGK, and $P_{\text{tol}} = 20$, $\theta = 0.1$, $\delta = 10$, $\alpha = 1$ and $\beta = 2$ for E2GK. With these settings, $J_{EGK} = 0.62612$ and $J_{E2GK} = 1.00000$. Thus, E2GK outperforms EGK as $J_{E2GK} = 1$ means that the partition discovered by E2GK perfectly matches the partition P considered as ground truth. This conclusion fits with the analysis of the results in section 4 illustrated in Fig.5 and Fig.6. It was shown that the centers of the clusters discovered by E2GK are very close to the real values, whereas EGK generates too many clusters with misplaced centers.

In the following, a study of the sensitivity of E2GK to parameters P_{tol} and θ is

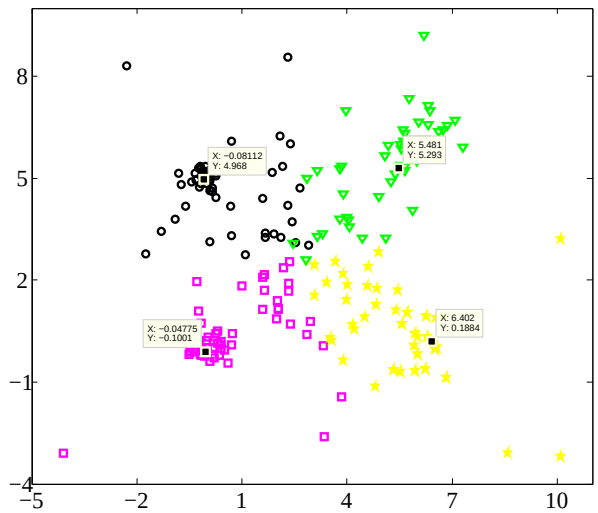

Figure 11: Final decision on clusters when $k = |\Omega|$.

provided on the same data set.

6.1. Influence of parameter P_{tol}

A parameter of particular interest is the P_{tol}, defined as the desired minimum amount of data points within a cluster [27]. The choice of this parameter is very dependent of the considered set of data. In EGK, the decision whether to create a new cluster or not is based on the value of P_{tol}. A major difference introduced in E2GK compared to the original method is that the evolution of the data structure after the creation of a new cluster is taken into account, and clusters that may have been valid before the creation of this new cluster could evolve to smaller clusters containing less than P_{tol} points. E2GK performs an additional step to remove these clusters making it possible not only to increase, but also to decrease the number of clusters.

To illustrate the influence of P_{tol} on the performance of E2GK, we conducted experiments on the same data set for different values of P_{tol}, the remaining parameters were kept unchanged. The Jaccard coefficient was calculated for each value of P_{tol} for both E2GK ($\theta = 0.1$, $\delta = 10$, $\alpha = 1$, $\beta = 2$) and EGK ($u_h = 0.5$, $\theta = 0.1$) algorithms. Considering P_{tol} taking the values 8, 9, 10, 12, 13, 15, 18, 20, 23, 25, 26, 27, 28, 30, 33, 35, 38, 40, 42, 44, 46, 48, 50, the comparative results are given in Fig. 12.

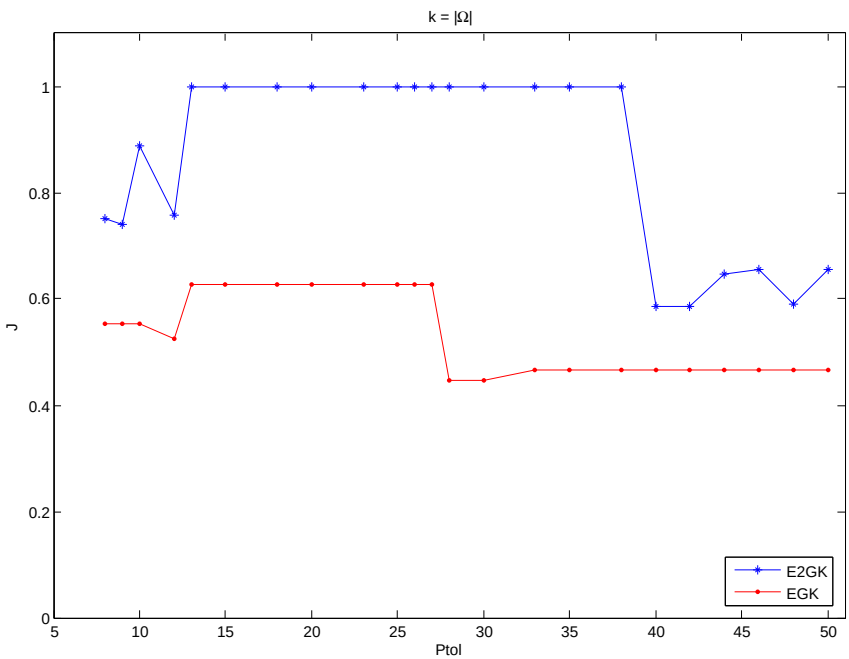

Figure 12: Evolution of the Jaccard Coeffitient with P_{tol} ($u_h = 0.5$, and $\theta = 0.1$)

For any value of P_{tol}, E2GK remains more efficient than EGK. One can notice that the variation of P_{tol} affects both algorithms in a similar way. As expected, too small or too high values of P_{tol} lead to a less meaningful partition of the data set and thus to a smaller J. Starting from $P_{\text{tol}} = 13$, and for both E2GK and EGK, the value of J remains constant at its maximum value. One notable difference is that this stable phase, of optimal value of J, is higher and larger in the case of E2GK. At $P_{\text{tol}} = 27$, J_{EGK} suddenly decreases, whereas a similar abrupt fall in the value of J_{E2GK} occurs at $P_{\text{tol}} = 38$. This can be explained by looking at the partition of data considered as ground truth. The considered partition is composed of 30 points each of type G_1, G_2, G_3 and G_4 to which 90 points of type noise were added to test the updating procedure. The influence of P_{tol} in the case of E2GK is reduced due to the fact that doubt between clusters has been taken into account in the computation of the credal partition, doubt mainly being added by the noisy data.

In section 5, we provided some solutions to reduce the complexity of E2GK through the concept of k-additive belief mass. Figures 13 and 14 shows the comparative results for $k = 2$ and $k = 3$. Basically the same conclusion as for the general case ($k = |\Omega|$) can be made. It is also to mention that the optimal value J_{E2GK} is reached at $P_{\text{tol}} = 15$ for $k = 2$ and $k = 3$. This reflects that E2GK is more sensitive to small values of P_{tol}, but also that the restriction to case of a 2 or 3-additive belief mass still permits a better stability to noisy data.

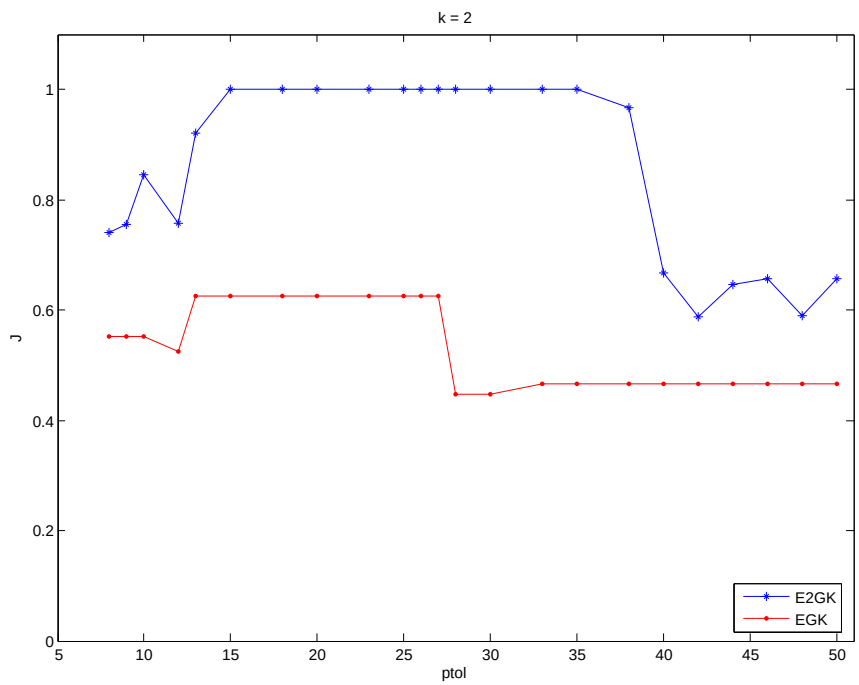

Figure 13: Evolution of the Jaccard Coeffitient with P_{tol} when $k = 2$ ($u_h = 0.5$, and $\theta = 0.1$).

6.2. *Influence of parameter θ*

Parameter θ is the learning rate of the updating procedure as mentioned in section 3.2. in Eq. 23 ans Eq. 25. It takes its values in $[0.05, 0.3]$ and determines the step of searching in the updating rule. Large values of θ guarantee sparsely selected clusters. The choice of parameter θ remains difficult as in some cases, a large step leads to large changes that could ignore valuable clusters [27].

Considering the same example as before, we provide comparative results of EGK and E2GK when parameter θ varies. For this test, P_{tol} was chosen equal to 20 as it was previously shown to lead to optimal values of J_{EGK} and J_{E2GK} (Fig. 12). Values of θ range between 0.05 and 0.3 with a step of 0.05. The results are depicted in Fig. 15 and underline the fact that the performance J_{E2GK} remains optimal for any value of θ whereas J_{EGK} varies between 0.48 and 0.76 until $\theta = 0.28$. At $\theta = 0.12$, $J_{EGK} = 1$ meaning that The resulting partition of EGK perfectly matches the considered partition P. Additional experiments by looking at the Jaccard coeffients values when both θ and P_{tol} vary show the same trend, i.e. while EGK is very influenced by the variations of θ, E2GK show few changes due to θ. This might indicate that parameter P_{tol} is a crucial parameter that deserves to be paid a particular attention.

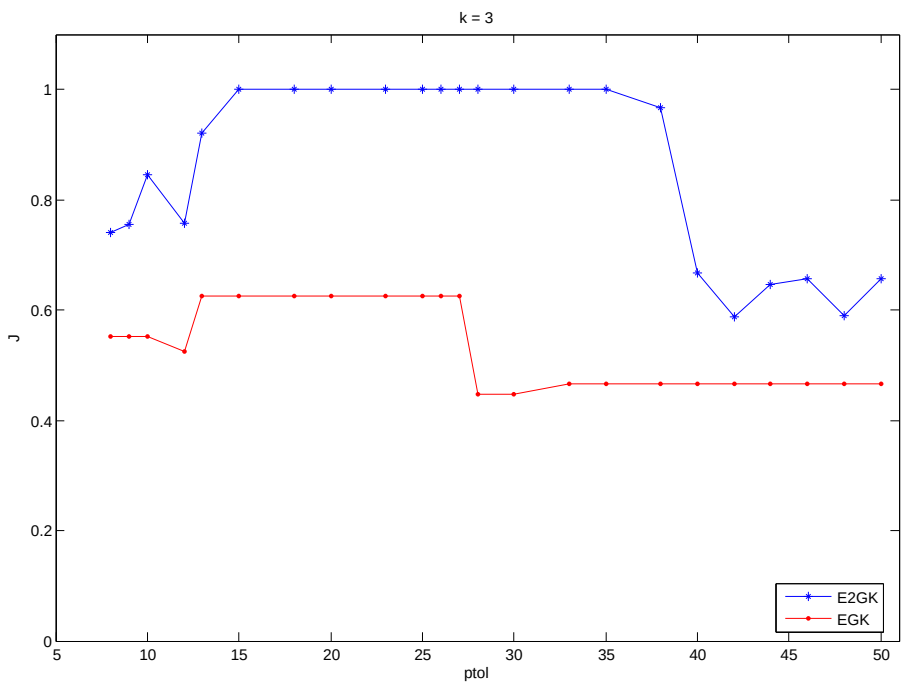

Figure 14: Evolution of the Jaccard Coeffitient with P_{tol} when $k = 3$ ($u_h = 0.5$, and $\theta = 0.1$).

6.3. Convergence issues

The fact that the complete dataset is not available makes difficult the proof of optimality of the centers and covariances estimates in online methods. In EGK, updating is based on static GK equations, and a kind of convergence can generally be observed in an experimental way with appropriate θ. The learning rate θ determines the step of searching and is generally made small to avoid large changes that could miss a valuable center. In E2GK, we use a similar principle by assuming it inherits these convergence properties. In experiments, this scheme demonstrated a greater robustness than EGK with respect to θ and P_{tol}.

7. Application of E2GK

7.1. A benchmark 1-D problem

Let consider the Mackey-Glass chaotic time series defined as follows:

$$x(t) = \frac{0.2 \cdot x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1 \cdot x(t) \quad , \tag{31}$$

with $\tau = 17$ and $x_0 = 1.2$. A total of 270 samples were generated. E2GK parameters were set to $\delta = 10$, $\alpha = 1$, $\beta = 2$, $\theta = 0.01$ and $P_{\text{tol}} = 10$.

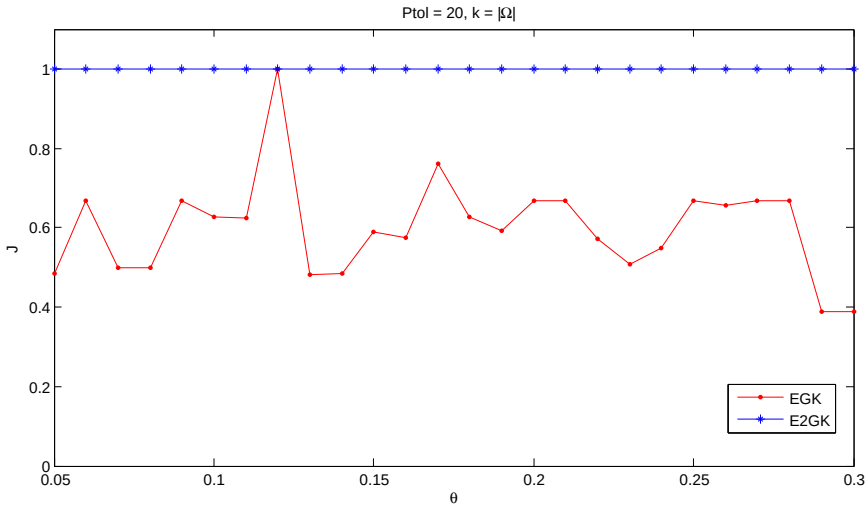

Figure 15: Evolution of the Jaccard Coeffitient with θ ($P_{\text{tol}} = 20$, $u_h = 0.5$)

The obtained series is depicted in Figure 16 as well as the resulting segmentation by E2GK (using $[t \;\; x(t)]$ as inputs). Figure 17 shows the number of clusters evolving along time. The online segmentation provides 10 segments well located on the curve.

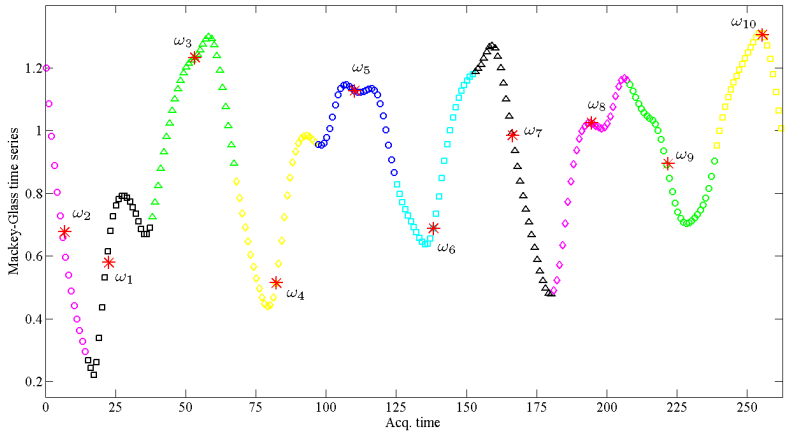

Figure 16: The Mackey-Glass time series and its online segmentation. Prototypes appear in red stars.

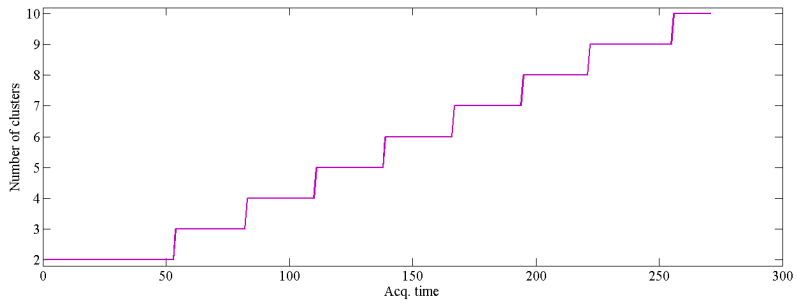

Figure 17: Number of clusters along time for the first application (Mackey-Glass).

7.2. Square data

As a second application, we propose to test the ability of E2GK on a particular dataset that we call *square data*. The dataset is composed of 10 blocks, each composed of 50 data points as depicted in figure. We first generate 10 centers $\tilde{c}_i, i = 1 \dots 10$ uniformly distributed in $[-8, 8]$, around which 50 data points are uniformly drawn in $[\tilde{c}_i - 0.5, \tilde{c}_i + 0.5]$.

E2GK algorithm is randomly initialized using 2 centers and 30 data points. Parameters were set to: $\delta = 100$, $\alpha = 1$, $\beta = 2$, $\theta = 0.01$ and $P_{\text{tol}} = 40$.

The remaining data points are gradually (one by one) given to E2GK and the resulting clusters are shown in Fig. 18. One can see that E2GK perfectly recognizes the square blocks whereas EGK fails to properly locate the centers (Fig. 19).

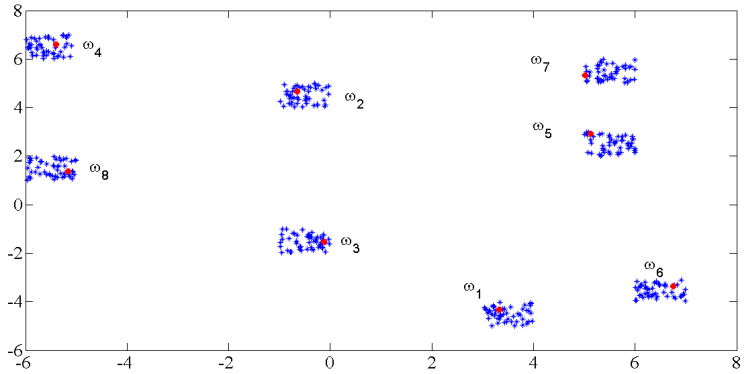

Figure 18: Square data : Decision on clusters for each point based on the pignistic probabilities obtained from the credal partition using E2GK algorithm. Also are displayed the coordinates of the centers found by E2GK.

7.3. A multidimensional real problem on PRONOSTIA platform

To illustrate our results, a dataset provided by an experimental platform called PRONOSTIA is used. This platform is dedicated to bearing prognosis. PRONOS-

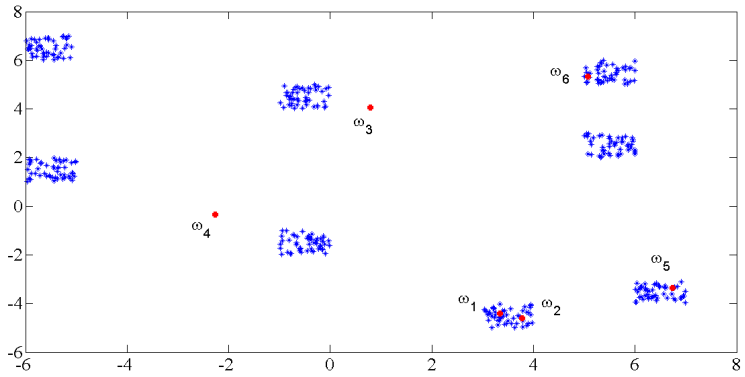

Figure 19: Square data : Decision on clusters for each point based on the maximum of degree of membership from the fuzzy partition using GK algorithm. Also are displayed the coordinates of the centers found by EGK. $u_h = 0.7$ and the other parameters are the same as in E2GK ($\theta = 0.01$ and $P_{\text{tol}} = 40$).

TIA is developed within the Department of Automatic Control and Micro-Mechatronic Systems (AS2M) of FEMTO-ST institute[3] for the test and validation of bearing prognostics approaches. The originality of this experimental platform lies in the characterization of both the bearing functioning and its degradation and also in the possibility to make the operating conditions of the bearing vary during its useful life.

Pronostia (Fig. 20) is an experimentation platform dedicated to the tests and validation of the machinery prognosis approaches, focusing on bearing prognostics. The main objective of Pronostia is to provide real experimental data that characterize the degradation of a ball bearing along its whole operational life (until fault/failure). Vibration and temperature measurements of the rolling bearing during its functioning mode are collected by sensors.

As prognosis algorithms need statistical data, it is necessary to conduct an experiment in just a few hours, and so collect a large amount of data in a few weeks. To do so, we developed a device that is able to maintain the bearing under study into hard operating conditions.

A data acquisition system was developed to ensures the visualization of the signals provided by the different sensors and sampled in a specific manner. Thus, all data can be monitored in real time on scrolling graphs. The raw signals provided by the sensors are processed in order to extract relevant information concerning bearings states. Several techniques have been implemented and gathered in a signal processing toolbox with Matlab: time-domain methods (RMS, skewness and kurtosis, crest factor, K-factor, Peak-to-Peak), frequency-domain methods (spectral and cepstrum analysis, envelope detection), time-frequency domain (short-time fourier transform) and discrete

[3]FEMTO-ST stands for “Franche-Comté Electronics, Mechanics, Thermal Processing, Optics - Science and Technology”. The platform was developed in AS2M department (Automatic control and Micro-Mechatronic Systems).

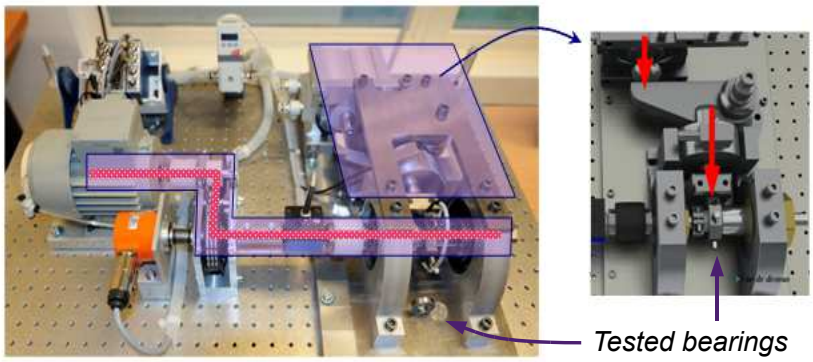

Figure 20: PRONOSTIA platform.

wavelets.

Figure 21 illustrates the power spectral density of the vertical acceleration sensor computed during the last half of the test period. It shows a growing amplitude at the end of the experiment when the bearing is gradually degrading. Various other data processings are possible to provide the necessary tools for bearing prognostics.

We consider here the power spectral density made of 512 points at each time slice. This huge dataset is then post-processed by a principal components analysis in order to automatically select the 6 most representative frequencies. These 6 features are used as inputs of E2GK (with 250 points).

We here applied E2GK in order to automatically find a partitioning (online) of the data streams. E2GK algorithm is initialized randomly using 2 centers and 20 data. Parameters were set to: $\delta = 20$, $\alpha = 1$, $\beta = 2$, $\theta = 0.01$ and $P_{\text{tol}} = 15$.

The first 20 data correspond roughly to the first 0.5 hour of the experiment where the bearing does not present any default. Then, data arrive sequentially and make the clustering structure possible to evolve. E2GK adapted its structure until obtaining 7 clusters as pictorially described in Fig. 22. First of all, a third cluster is obtained into the cloud around the initialization points. This shows that the bearing only degrades from about the third hour. Then 4 clusters are gradually added according to the degradation. Cluster ω_4 represents a *transition* between the *normal modes* (ω_1, ω_2 and ω_3) towards the *degrading modes* (ω_5 and ω_6). Finally the *fault mode* is detected with ω_7. Figure 23 shows the assignments, i.e. cluster chosen for each data point (according to the maximum of belief mass).

The application of E2GK on PRONOSTIA's data presents a realistic experimental case study of a common practical problem, that is online fault detection of bearing degradation. The application demonstrates that the health state of the bearing can be represented by evolving clustering and associating the clusters with the main states. The association between the clusters provided by E2GK and the health states is of a crucial importance in the context of novelty detection and prognostics applications. It is rationale to model the operating modes of a bearing as a set clusters [44, 45], and the lack of well defined boundaries between operating modes and the gradual transitions

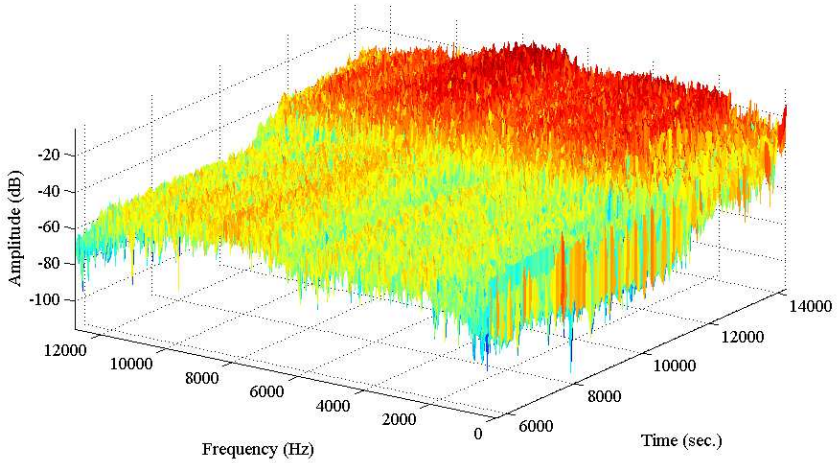

Figure 21: Power spectral density of the vertical acceleration sensor.

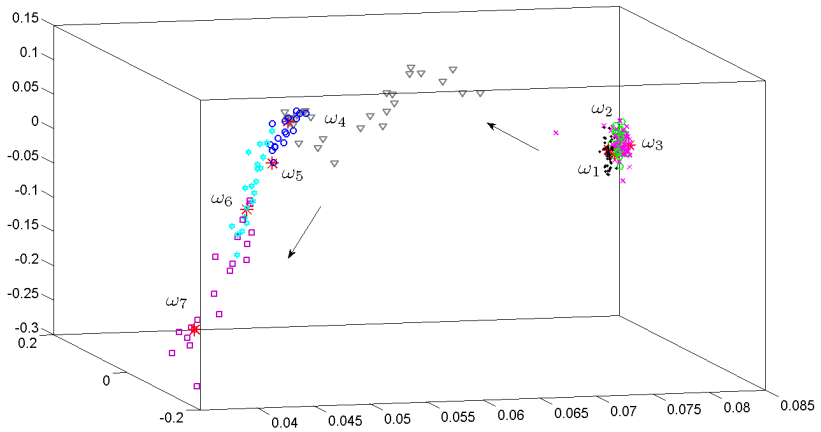

Figure 22: Online segmentation into clusters for PRONOSTIA's data. Centers are depicted with red crosses and arrows represents the order of arrival of the data.

between them makes E2GK of particular interest. Indeed, as shown in section 3, the concept of credal partition enables the explicit representation of doubt between clusters. Given this segmentation of PRONOSTIA's data streams into meaningful clusters and based on the evolution of these clusters, one can be interested in predicting a potential occurrence of a degradation. In [44], the authors discussed a similar topic using a fuzzy clustering method and a different decision making approach. Using E2GK, we are currently developing a prognostics approach exclusively based on belief functions to be compared with Angelov's work on fuzzy evolving systems [20, 21].

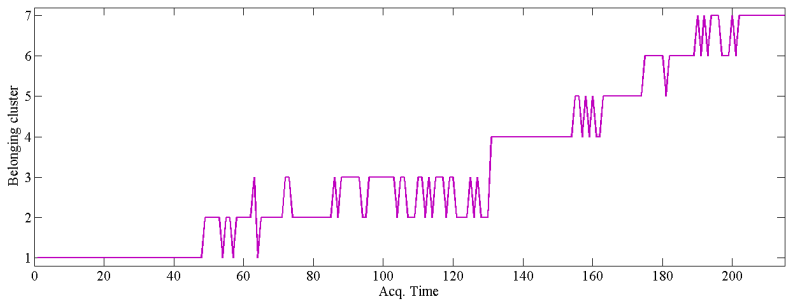

Figure 23: Number of clusters along time for the second application.

8. Conclusion

To our knowledge, only one *incremental* approach to clustering using belief functions has been proposed [28]. In this approach the data in the training set are considered uncertain. Moreover, each data is described by a given number of attributes, each labeled by a mass of belief provided by an expert. Also, this approach assumes that the number of clusters is known in advance.

E2GK algorithm is an evolving clustering algorithm using belief function theory, which relies on the credal partition concept. This type of partition permits 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 [27], initially based on a fuzzy partition, to which we bring some modifications:

- use the median operator to calculate cluster radius, which is more robust than the maximum rule,
- use of credal partitioning for a better representation of the data structure and an improved robustness to cluster evolution,
- change the structure of partitioning by adding or removing clusters (vs. adding only in EGK).

Solutions have been proposed to limit complexity issues, based on the concept of k-additive belief functions. The study of the influence of parameter k shows the importance of doubt representation in the clustering process to limit the number of clusters in the final partition matrix. Simulation results show that E2GK discovers relatively well the changes in the data structure. An analysis of parameters sensitivity (P_{tol} and θ) has also been carried out and demonstrated that E2GK is more robust than EGK.

Acknowledgment

We greatly thank Professor Thierry Denœux for fruitful discussions concerning the complexity issues discussed in this paper, as well as the French Ministry of Advanced

Education and Research for partially supporting this work, and Patrick Nectoux for his implication in the platform PRONOSTIA, and the anonymous reviewers for their helpful comments.

References

[1] C. M. Bishop, Neural networks for pattern recognition, Oxford University Press, 1995.

[2] M. R. Anderberg, Cluster Analysis for Applications, Academic Press, New York, NY, 1973.

[3] A. K. Jain, R. C. Dubes, Algorithms for Clustering Data, Prentice-Hall advanced reference series, Prentice-Hall, Inc., Upper Saddle River, NJ., 1988.

[4] E. Diday, J. C. Simon, Clustering analysis, Digital Pattern Recognition (1976) 47 – 94.

[5] A. Jain, M. Murty, P. Flynn, Data clustering: A review, ACM Computing Surveys, Vol. 31, No. 3, September 1999 31 (1999) 264 – 323.

[6] R. Xu, D. Wunsch II, Survey of clustering algorithms, IEEE Transactions on Neural Networks 16 (2005) 645 – 678.

[7] V. Brailovsky, A probabilistic approach to clustering, Pattern Recogn. Lett. 12 (1991) 193 – 98.

[8] C. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comput. 20 (1971) 68 – 86.

[9] J. McQueen, Some methods for classification and analysis of multivariate observations, in: Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[10] L. A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338 – 353.

[11] J. C. Bezdek, Pattern Recognition with fuzzy objective function algorithms, Plenum Press, New York, 1981.

[12] E. Gustafson, W. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in: IEEE Conference on Decision and Control, 1978.

[13] R. Krishnapuram, J. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Syst. 1 (1993) 98 – 110.

[14] T. Denoeux, M.-H. Masson, EVCLUS: Evidential clustering of proximity data, IEEE Transactions on Systems, Man and Cybernetics Part B 34(1) (2004) 95 – 109.

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

[16] R. Duda, P. Hart, D. Stork, Pattern Classification, Wiley, New york, 2001.

[17] S. Janichen, P. Perner, Acquisition of concept description by conceptual clustering, Perner, P., Imiya, A. (Eds.), Machine Learning and Data Mining in Pattern Recognition. Lecture Notes in Artificial Intelligence 3587 (2005) 153 – 163.

[18] J. Beringer, E. Hüllermeier, Online clustering of parallel data streams, Data & Knowledge Engineering 58 (2006) 180 – 204.

[19] S. Zhong, Efficient online spherical k-means clustering, in: IEEE International Joint Conference on Neural Networks, 2005.

[20] P. Angelov, E. Lughofer, X. Zhou, Evolving fuzzy classifiers using different model architectures, Fuzzy Sets and Systems 159 (2008) 3160 – 3182.

[21] P. P. Angelov, D. P. Filev, An approach to online identification of Takagi-Sugeno fuzzy models, IEEE Trans Syst Man Cybern B Cybern 34 (2004) 484 – 98.

[22] P. Angelov, An approach for fuzzy rule-base adaptation using on-line clustering, International Journal of Approximate Reasoning 35 (2004) 275 – 289.

[23] D. Park, I. Dagher., Gradient based fuzzy c-means (GBFCM) algorithm, in: IEEE Int. Conf. on Neural Networks, Orlando, FL, 1994, pp. 1626 – 1631.

[24] Y. Bodyanskiy, Computational intelligence techniques for data analysis, Lecture Notes in Informatics 72 (2005) 15 – 36.

[25] P. Hore, L. Hall, D. Goldgof, W. Cheng, Online fuzzy c-means, in: Annual Meeting of the North American Fuzzy Information Processing Society, 2008.

[26] O. Georgieva, F. Klawonn, Dynamic data assigning assessment clustering of streaming data, Applied Soft Computing 8 (2008) 1305 – 313.

[27] D. Filev, O. Georgieva, An extended version of the Gustafson-Kessel algorithm for evolving data stream clustering, in: Evolving Intelligent Systems, John Wiley ans Sons, New York, 2010, pp. 293 – 315.

[28] S. Ben-Hariz, Z. Elouedi, IK-BKM: An incremental clustering approach based on intra-cluster distance, in: Eighth ACS/IEEE International Conference on Computer Systems and Applications, AICCSA, 2010.

[29] P. Smets, R. Kennes, The Transferable Belief Model, Artificial Intelligence 66 (1994) 191 – 234.

[30] L. Serir, E. Ramasso, N. Zerhouni, E2GK: Evidential evolving gustafsson-kessel algorithm for data streams partitioning using belief functions, in: W. Liu (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Vol. 6717 of Lecture Notes in Computer Science, Springer Berlin, 2011, pp. 326 – 337.

[31] D. Dubois, H. Prade, P. Smets, New semantics for quantitative possibility theory, in: 2nd International Symposium on Imprecise Probabilities and Their Applications, Ithaca, New York, 2001.

[32] T. Denoeux, A k-nearest neighbor classification rule based on Dempster-Shafer theory, IEEE Trans. on Systems, Man and Cybernetics 25 (5) (1995) 804 – 813.

[33] H. Kim, P. H. Swain., Evidential reasoning approach to multisource-data classification in remote sensing, IEEE Transactions on Systems, Man and Cybernetics 25(8) (1995) 1257 – 1265.

[34] V.Antoine, B. Quost, M.-H. Masson, T. Denoeux, CECM: Constrained evidential c-means algorithm, Computational Statistics & Data Analysis-doi:10.1016/j.csda.2010.09.021.

[35] B. Cobb, P. Shenoy, On the plausibility transformation method for translating belief function models to probability models, Int. journal of approximate reasoning 41 (3) (2006) 314 – 330.

[36] T. Kohonen, Self Organization and Associative Memory, Springer-Verlag, Berlin, 1989.

[37] M. Grabisch, k-order additive discrete fuzzy measures and their representation, Fuzzy Sets and Systems 92 (1997) 167 – 189.

[38] R. N. Dave, Validating fuzzy partitions obtained through c-shells clustering, Pattern Recogn. Lett. 17 (1996) 613 – 623.

[39] I. Gath, A. Geva, Unsupervised optimal fuzzy clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (1989) 773 – 780.

[40] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Cluster validity methods: part I, SIGMOD Rec. 31 (2002) 40 – 45.

[41] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Clustering validity checking methods: part II, SIGMOD Rec. 31 (2002) 19 – 27.

[42] W. Rand, Objective criteria for the evaluation of clustering methods, Journal of the American Statistical Association 66 (336) (1971) 846 – 850.

[43] P. Jaccard, The distribution of flora in the alpine zone, New Phytologist 11 (1912) 37 – 50.

[44] D. Filev, R. Chinnam, F. Tseng, P. Baruah, An industrial strength novelty detection framework for autonomous equipment monitoring and diagnostics, IEEE Transactions on Industrial Informatics 6 (2010) 767 – 779.

[45] L. Serir, E. Ramasso, N. Zerhouni, Time-sliced temporal evidential networks: the case of evidential HMM with application to dynamical system analysis, in: IEEE Int. Conf. on Prognostics and Health Management, Denver, CO, USA, 2011.