Granul. Comput. (2016) 1:1–11
DOI 10.1007/s41066-015-0012-z

ORIGINAL PAPER

DCC: a framework for dynamic granular clustering

Georg Peters[1,2] · **Richard Weber**[3]

Received: 23 July 2015 / Accepted: 17 November 2015 / Published online: 4 February 2016

Abstract Clustering is one of the most relevant data mining tasks. Its goal is to group similar objects in one cluster while dissimilar objects should belong to different clusters. Many extensions have been developed based on traditional cluster algorithms. Recently, approaches for dynamic as well as for granular clustering have been of particular interest. This paper provides a framework, DCC-Dynamic Clustering Cube, to categorize existing dynamic granular clustering algorithms. Furthermore, the DCC-Framework can be used as a research map and starting point for new developments in this area.

Keywords Dynamic clustering · Granular clustering · Granular computing

1 Introduction

The amount of data collected, their accessibility as well as the computational and methodological capabilities to analyze them have significantly increased in the past years. The term Big Data subsumes this trend and is often regarded as one of the key factors that determine an enterprise's competitiveness. According to the information technology research and advisory company Gartner (2011) Big Data is characterized by the following three Vs: volume, variety, and velocity.

In the context of our paper, velocity is of special interest. A high velocity of incoming data requires strategies to *dynamically* adapt the analytic system to cope with the changing behavior of the respective data structures.

Furthermore, due to the sheer amount of data, Big Data requires procedures to develop cost-efficient solutions that are simplified but still acceptable representations of the underlying complex patterns. As they are intended to address human decision-makers, they should be descriptively presented in human centered ways without jeopardizing too much of the content of the original data. These requirements are very similar to those that Yao (2005) postulates for *granular computing* (see Sect. 2.2).

Last but not least, one of the most popular methods in data mining and Big Data is *clustering* (Jain et al. 1999). The goal of clustering is to group similar objects into a cluster while dissimilar objects should be separated by assigning them to different clusters.

Putting these three keywords, *dynamic*, *granular computing*, and *clustering*, together we get *dynamic granular clustering*. The objective of the paper is to address these three keywords holistically by developing a framework for dynamic granular clustering: DCC—Dynamic Clustering Cube. Our framework helps to categorize established dynamic granular cluster approaches and discloses methodical gaps that should be considered to be filled. We study various approaches of granular computing for clustering rather than focussing on a particular granular clustering method.

The remainder is organized as follows. The next section, gives a brief introduction to the characteristics of dynamic

✉ Georg Peters
georg.peters@hm.edu

✉ Richard Weber
rweber@dii.uchile.cl

[1] Department of Computer Science and Mathematics, Munich University of Applied Sciences, Lothstrasse 34, 80335 Munich, Germany

[2] Australian Catholic University, Sydney, Australia

[3] Department of Industrial Engineering, FCFM, Universidad de Chile, República 701, Santiago, Chile

data, granular computing, and cluster analysis. In Sect. 3, we propose our DCC-Framework and discuss its three dimensions. In Sect. 4, we present selected dynamic granular clustering methods showing how each one fits into the DCC-Framework. In Sect. 5, we review selected application areas where dynamic granular clustering offers particular advantages. Section 6 concludes this paper and hints at future developments.

2 Dynamic data, granular computing, and cluster analysis

2.1 Dynamic data

Data to be analyzed can be dynamic in different ways, e.g., with respect to location or time, etc. For the purpose of this article, we understand *dynamic data* as any kind of data that take only time-dependent aspects into account and refrain from any other dynamic phenomena, like geographical movements and others.

Following Joentgen et al. (1999), we can distinguish two cases regarding time-dependent aspects: (1) objects whose feature vectors contain just values at a certain moment of time, i.e., snapshots of feature values and (2) objects that contain functions of feature values over time, i.e., feature trajectories. An example for the first case is a feature vector describing a certain customer with its current attribute values, such as, age and income, as used, e.g., for customer segmentation in the retail industry. An example for the second case is a feature vector describing a certain patient's blood pressure and heart rate during the past hours, as used for patient monitoring in health care.

Another important issue is whether observations can be identified over time or not. If objects are identifiable over time, their respective profile constitutes dynamic data, e.g., customers' buying behavior over time (see, e.g., Berkhin 2006). In the opposite case, i.e., objects are not identifiable, dynamic data provides information on the behavior of the entire set of analyzed objects, e.g., changing buying behavior of all customers from a customer base. In both cases, i.e., identifiable or non-identifiable objects, it is necessary to store a time stamp along with each newly generated feature value.

2.2 Granular computing

In the past decades, *granular computing* (Bargiela and Pedrycz 2003; Pedrycz 2007; Pedrycz et al. 2008) has emerged as a new archetype to simplify problems by dealing with information granules derived from underlying true but complex data. Granular computing is motivated by human problem solving strategies which are often based on information granules rather than on precise data (Pedrycz 2013). If one takes the present state of evolution of mankind, such a strategy seems to be more successful than alternative strategies that are directly based on the underlying real data.[1]

The original idea of granular computing goes back to the nineties of the last century. For example, Zadeh (1997) wrote: "Fuzzy information granulation underlies the remarkable human ability to make rational decisions in an environment of imprecision, partial knowledge, partial certainty and partial truth." In 2004, Yao (2004) stated that "the consideration of granularity is motivated by the practical needs for simplification, clarity, low cost, approximation, and the tolerance of uncertainty"; in 2005, Yao (2005) concluded that granular computing should be "1) Truthful representation of the real world [...] 2) Consistent with human thinking and problem solving [...] 3) Simplification of problems [...] 4) Economic and low cost solutions."

Bell et al. (1988) mentioned descriptive, normative, and prescriptive interactions that people are having when they are taking decisions. The descriptive analysis is of special importance in the context of our paper. It identifies ways and habits how people take decisions, putting special emphasis on decomposition of complex problems, cost-benefit analysis instead of optimal solutions, and simplification in case of uncertainties present in the environment. This underlines the importance of granular computing when it comes to develop systems for decision support in human-centric situations.

As can be seen, granular computing is not method, but goal-driven. Any method that helps to reach these goals should be considered as a substantial component of granular computing. This goal-driven definition of granular computing also implies that its techniques are not necessarily developed for granular computing. It also subsumes methods that were proposed long before the term granular computing itself was introduced (Yao 2008a, b). Bargiela and Pedrycz (2003) identify set theory and interval analysis (Kreinovich 2008), fuzzy (Zadeh 1965), rough (Pawlak 1982), and shadowed sets (Pedrycz 1998), probabilistic sets and probability-based granular constructs, and higher-level granular constructs as important methods within the portfolio of granular computing.

These techniques are used to aggregate detailed data towards information granules by applying different aggregation characteristics: e.g., probabilistic approaches use probabilities and fuzzy sets similarities as granulation

[1] Of course, one could also argue that evolution will strengthen the human ability to analyze complex data in the future. In this case, information granules are presently needed due to the underdeveloped state in human development.

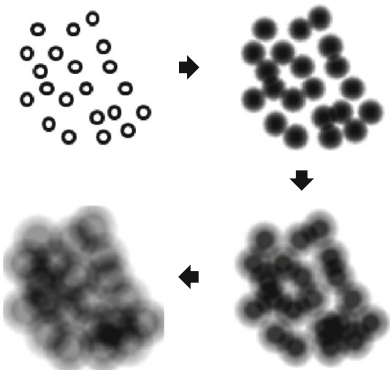

Fig. 1 Granulation of objects

criteria for building information granules. See Fig. 1 for an illustrative example for a three-step granulation.

2.3 Cluster analysis

Besides methods as discussed above, e.g., data mining techniques are needed to generate information granules. *Clustering* is one of the most popular techniques in data mining with a virtually unmanageable number of algorithms. See, e.g., Jain et al. (1999) or Xu and Wunsch (2005) for surveys on cluster analysis.

The goal of clustering is to group similar objects into the same cluster, while dissimilar objects should belong to different clusters. The degree of similarity of objects is calculated based on their respective feature values (see, e.g., Berkhin 2006).

In terms of granular computing, a cluster can be interpreted as an information granule that presents its objects on a coarser and more granular level (Gacek and Pedrycz 2015). Two important areas in clustering are hierarchical (Li et al. 2011) and partitive approaches (Xu and Wunsch 2005).

Hierarchical clustering is divided into divisive and agglomerative methods. In hierarchical divisive clustering, one starts with one cluster for all data. By splitting clusters, the representation of the data gets finer and finer until each object forms its own cluster. In hierarchical agglomerative clustering, one starts with each object forming its own cluster and move upwards merging clusters until all objects belong to the same cluster.

In partitive clustering, the objects are assigned to a (predefined) number of clusters based on similarities. The most popular partitive cluster approach is probably k-means (MacQueen 1967) and its extensions and derivatives (Peters et al. 2013). In the context of our paper, clustering algorithms based on soft computing approaches are of particular importance, such as, e.g., fuzzy c-means (Bezdek 1981), possibilistic c-means (Krishnapuram and Keller 1993), rough k-means (Lingras and West 2004; Peters 2014), and granular clustering (Pedrycz 2005).

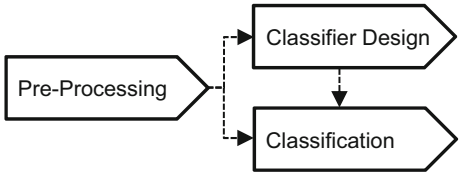

Fig. 2 Phases of clustering

The obtained clustering results can also be used to assign new observations to the established clusters. We call the first phase of clustering *classifier design* and the second phase *classification*.[2] Obviously, the structure of the data used for classifier design has to be at least similar, at best identical to the structure of the new data that are sent on the classifier to obtain reasonable results. Figure 2 shows the two phases of clustering enriched by an optional preprocessing phase.

3 The dynamic clustering cube

3.1 Foundations of the DCC-Framework

Static data characteristics are rather the exception than the rule in many real-life applications. Hence, dynamic approaches to clustering have become of rapidly increasing importance recently. They address the need to constantly adapt the clustering process to changes in the analyzed data domain.

To categorize algorithms in the field of dynamic granular clustering, we propose the DCC-Framework that consists of the three crucial dimensions of dynamic clustering (Fig. 3), i.e.,

- *Characteristics of change*,
- *Types of granulation*,
- *Clustering processes*.

While the two dimensions *Characteristics of Change* and *Clustering Processes* are of general nature, the third dimension, the *Types of* dimension can be specified context-dependently. In our paper, we deal with *Types of Granulation*; alternatively it could be, e.g., *Types of Uncertainty* according to Zadeh's Generalized Theory of Uncertainty (Zadeh 2006), or possibly a further characteristic.

[2] Classification, not to be confused with the common distinction between unsupervised learning (clustering) and supervised learning (classification or regression).

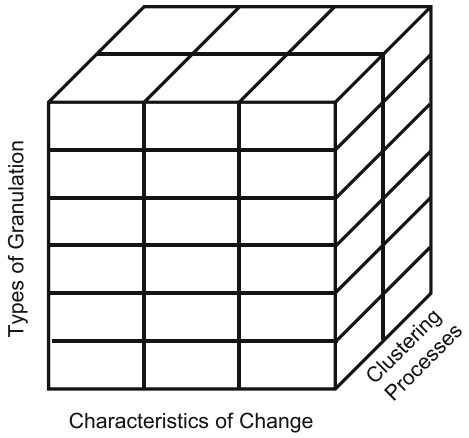

Fig. 3 The dynamic clustering cube

3.2 Dimensions of DCC

3.2.1 DCC: characteristics of change

DCC's *Characteristics of Change* dimension addresses kinds of change in the data domain to be analyzed. As already discussed above, *Characteristics of Change* can be observed in several circumstances, e.g., in spatial environment or regarding time, beside others.

We do not further investigate changes in spatial environment, etc., but concentrate on changes due to time only. We identify three different cases and examine them in greater detail. They are:

- No change,
- Cluster movements, and
- Changes in cluster structures.

No change. Obviously, the simplest situation to deal with, is when the data do not change at all (see Fig. 4, where the black arrows indicate time steps between the diagrams). In these cases, static cluster algorithms perform well. There is no need to implement components into the algorithms that address changing characteristics of the data domain.

Note, that static data domains are often considered in real-life projects even when data structures change over time. Reasons for neglecting changes include, e.g., that users are not aware of changing data structures. Users might also just ignore them since they do not have the resources to adapt their systems to the changes, or they consider them as marginal. Obviously, only the last reason, marginal changes, is acceptable while the first and second reasons may lead to misinformation and, therefore, contradict the objectives of cluster analysis.

Cluster movements. In the second case, cluster movements, we identify the following subcategories: seasonal changes, trend changes, and random movements.

(a) *Seasonal changes.* Seasonal changes are characterized by a sequence of distinct patterns that repeatedly occur: Season 1 → Season 2 → Season 3 → Season 4 → Season 1 → and so on (see Fig. 5). Seasonal patterns can be observed frequently. The four seasons of a year are prominent examples. Note, that the four seasons of a year are not crisply separated homogeneous seasons, but steadily moving patterns from winter-like, to spring-like, to summer-like to fall-like weather conditions. They include features like temperatures, rainfall, and others (see, e.g., the average monthly temperature of Berlin in Fig. 6). So, the four weather seasons (winter, spring, summer, and fall) can be interpreted as information granules that help to reduce complexity in the definition of our environment.

Further seasonal changes are, e.g., induced by cultural habits, for example religious seasons like Christmas or Eastern, or sport seasons, like, e.g., sport activities assigned to the Summer Olympics contrasting winter sports. Identifying and/or anticipating such seasonal changes provides benefits, e.g., for customer segmentation where some provider

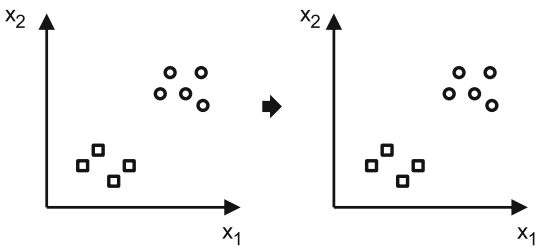

Fig. 4 No changes in the data domain

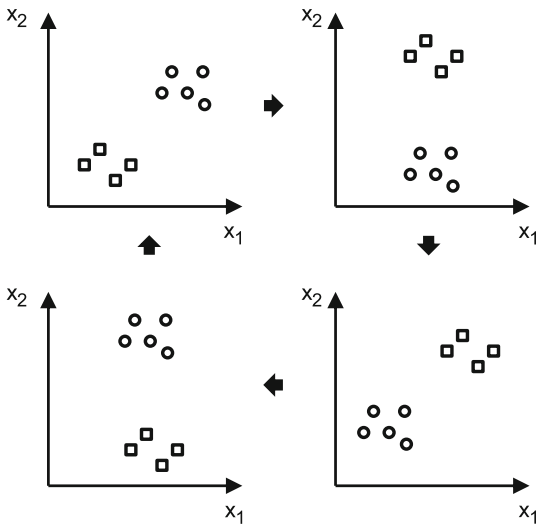

Fig. 5 Seasonal changes

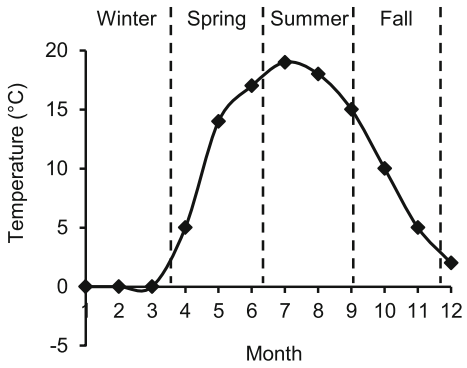

Fig. 6 Seasonal patterns: average temperature of Berlin (Klimatabelle 2015)

(e.g., retail, tourist industry) can offer the right product at the right time for a changing customer segment.

When we assume that each season is characterized by a stable data domain, we do not need to apply dynamic cluster algorithms again. In case we can identify the seasons, seasonal changes can be treated like a sequence of stable data domains. Then it is only of interest to detect deviations from the presumably stable seasonal changes.

(b) *Trend changes.* Trend changes are often of probabilistic nature, e.g., the means and/or the variances of data sets follow trends. For example, GDP (gross domestic product) or inflation may follow certain trends in some periods of time.

See, for example, Fig. 7 for trend movements of two clusters. The left cluster (squares) moves upwards while the second cluster (circles) moves downwards over time. In general, a trend can be non-linear. The oscillation of the temperature over a year could also follow a trend.

(c) *Random movements.* Different to the cluster movements discussed above, random movements of clusters are not predicable. Any new data must be tested for structural changes. If the changes are below a threshold they can be neglected, otherwise the cluster model has to be adapted to the new data.

In general, different effects (seasonal, trend, and random movements) can occur simultaneously. Such combinations, however, go beyond the scope of our paper.

Changes in cluster structure. In contrast to the cases discussed so far, we now investigate changes in the cluster structure itself. Crespo and Weber (2005) identified two such cases:

– Emerging clusters and
– Dying clusters.

So far, we have assumed that clusters move over time, but the number of clusters remains unchanged. However, in general, new clusters may emerge and existing clusters may disappear over time.

In the upper part of Fig. 8, we observe that a new cluster emerges. The two clusters (squares and circles) of the original data set are accompanied by new separated data (diamonds) that eventually form a new cluster.

The lower part of Fig. 8 shows an example for a dying cluster. While two clusters, the diamonds cluster and the circles cluster, are refreshed by new objects, the number of objects in the squares cluster remains unchanged over time. Due to its decreasing relative importance, it can be

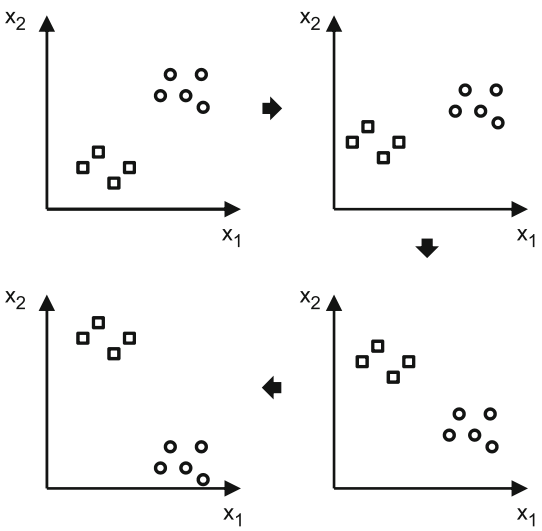

Fig. 7 Trend changes

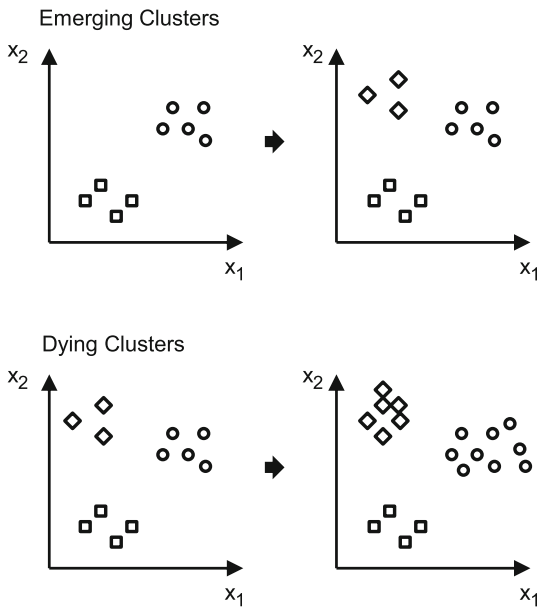

Fig. 8 Emerging and dying clusters

considered as dying. In the long run, it is a possible candidate for removal from the data set. The reader is referred to Crespo and Weber (2005), Peters and Weber (2012), and Peters et al. (2012) for a more detailed discussion on possible criteria to identify emerging and dying clusters.

3.2.2 DCC: types of granulation

As already mentioned in Sect. 2.2, granular computing is a general concept for information processing rather than a specific method or algorithm. For the purpose of this paper, however, we concentrate on its use for clustering objects that are described by features. Since clustering can be considered as one step within the KDD (Knowledge Discovery in Databases) process (Fayyad et al. 1996), we follow the respective methodology and discuss the *Types of Granulation* dimension for the following three elements separately: input data, preprocessing, and cluster approach.

Input data. On the one hand, numeric input values can be treated as crisp numbers, e.g., a customer's age is 23 years, which corresponds to a medium degree of granulation; a finer one would be, e.g., 23 years, 2 months, 5 days, and 10 h. Information granules with a higher degree of granulation are, e.g., intervals of real-valued components of a feature vector (continuing our example: the customer is a twen). On the other hand, numeric information can also be granularized by, for example, fuzzy or rough concepts. If input information is not numeric, as is the case, e.g., with text, the subsequent preprocessing steps convert this non-numeric information into numeric values.

Preprocessing. Following the before-mentioned KDD process, input data should be preprocessed. Here, we concentrate on preprocessing that generates information granules, instead of analyzing all relevant techniques in this step. The least degree of granulation is simply to refrain from any kind of preprocessing. If we have an input vector with real-valued attributes, intervals could be built, such as range of age or range of income as is often the case in polls. Another technique to pre-process numeric feature values is principal component analysis (PCA) (Jolliffe 2002), which is used to compress original information in higher aggregated information granules, so-called principal components [or factors in the case of factor analysis (Mulaik 2009)]. If, however, input is non-numeric, we often use transformations to represent the original information by vectors of numeric-valued features. Exemplarily, we would like to mention the case of text mining, where text is transformed into real-valued feature vectors using the TFIDF transformation (Salton and McGill 1986) and the so-called vector space model; see Kroha et al. (2006). The same idea has been applied to clustering of images, music, web pages, among others.

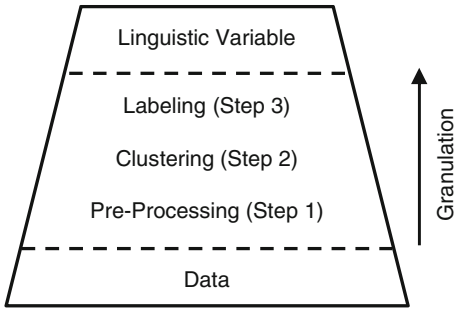

Fig. 9 Granulation steps of clustering

Cluster approach. Finally, the cluster approach itself exhibits different *Types of Granulation*. Crisp clustering, i.e., constructing clusters without modeling uncertainty as part of the cluster result, is already one way of aggregating information contained in a set of feature vectors, thus establishing information granules (Gacek and Pedrycz 2015). Other types of granulation can be obtained by considering different types of uncertainty modeling, arriving at, e.g., probabilistic clustering, fuzzy clustering, possibilistic clustering, or rough clustering, etc. (Pedrycz and Bargiela 2012).

Figure 9 depicts the granulation steps from input data via preprocessing techniques to clustering that eventually ends with the definition of linguistic variables.

Generally, in any of these steps granular objects could be observed, i.e., the input data can be information granules already, as suggested, e.g., by Gacek and Pedrycz (2015), who propose clustering of granular data and its application to time series clustering. In each single step, representations of the data on higher levels of granulation are possible, e.g., a cluster can be considered as granular representation of its members (Pedrycz 2013).

Any combination of 'input data', 'preprocessing', and 'cluster approach' is potentially possible as an instantiation of the dimension *Types of Granulation* of the proposed DCC-Framework. An example is crisp clustering of crisp input data without preprocessing; another example would be rough clustering of documents (e.g., newspaper articles) containing text that has been preprocessed using TFIDF and the vector space model.

Last but not least, please note, that the DCC-Framework itself can be regarded as a kind of granular categorization of clustering algorithms.

3.2.3 DCC: clustering processes

In the *Clustering Processes* dimension, different types of algorithmic structures are identified and categorized. For example, the classic k-means and its derivatives and extensions form a family of clustering algorithms

addressing static data sets. The basic structure is similar for all k-means-like algorithms, including classic k-means, fuzzy c-means, rough k-means, among others and contains basically the following four steps:

1. Initialization,
2. Calculation of the means,
3. Assignment of objects to clusters, and
4. Termination or going back to step 2.

In the dynamic case, the process of clustering is determined by several dimensions that may depend on each other. In the context of our paper, we propose the following dimensions that are discussed in more detail in the next paragraphs.

– Type of cluster algorithm,
– Flow of data, and
– Implemented dynamics.

Type of cluster algorithm. Most cluster algorithms can be categorized into partitive and hierarchical approaches. Although both have the objective of grouping similar objects in the same cluster and dissimilar objects in different clusters, their philosophies are different. Algorithmically, this leads to very different cluster processes. Even within, e.g., partitive clustering a diverse range of approaches can be observed. For example, one direction is the classic k-means family of algorithms, as another partitive approach support vector clustering has emerged (see Ben-Hur et al. 2001). Both directions have motivated the development of different types of cluster algorithms: see Peters et al. (2013) for a survey on soft k-means clustering and Saltos and Weber (2015) for a rough-fuzzy version of support vector clustering that detects outliers based on the information granules generated during clustering.

Flow of data. Data sent on a classifier can be treated object by object. Alternatively, they can be received in sets of objects, so-called batches. Hence, the dynamic classifier process is different when it is updated for each new object or after a certain number of objects arrived.

Implemented dynamics. When a dynamic component is required, two different implementations are possible. On the one hand, the static clustering algorithm remains unchanged. It is nested into a shell that monitors dynamic changes within the arriving data and triggers the static clustering algorithm to update its classifier when significant changes are observed. On the other hand, the dynamic component can be implemented in the core of the clustering algorithm itself.

4 Selected methods

The DCC-Framework provides a scheme to present existing techniques in a structured way. At the same time, it is a rich starting point to stimulate the development of new methods for dynamic granular clustering. In this section, we present exemplarily some already existing methods that belong to particular cells of the DCC, i.e., dynamic fuzzy c-means and rough k-means, evolving DDAA clustering, functional fuzzy c-means, and CluStream.

4.1 Dynamic fuzzy c-means and dynamic rough k-means

Since dynamic fuzzy c-means (Crespo and Weber 2005) and dynamic rough k-means (Peters et al. 2012) are similar in many of the cube's dimensions, they will be discussed together. We only hint at differences where these are relevant.

Dimension 1: Characteristics of change. After new data have been received, both methods first identify if the current cluster solution has to be modified or not. In the affirmative case, the respective base algorithm's parameters are updated. Among these parameters is the cluster number which allows to detect newly emerging or dying clusters. Applying the base algorithm (fuzzy c-means or rough k-means, respectively) provides the new cluster structure. In the case of dynamic rough k-means, Fig. 10 shows the respective clustering cycle; dynamic fuzzy c-means (Crespo and Weber 2005) has a similar structure.

Dimension 2: Types of granulation. As will be shown next, no advanced structure of input data or sophisticated preprocessing approaches are required.

– *Input data*. Input data are real-valued components of feature vectors as is the case of, e.g., traditional k-means.
– *Preprocessing*. No particular preprocessing is necessary in order to granularize the input data. Of course, some other kinds of preprocessing, such as normalization could be applied, that are not in the focus of this paper.
– *Cluster approach*. The cluster approach employs fuzzy or rough clustering, respectively.

Subsequently, labels, i.e., linguistic variables, are assigned to the clusters found. This goes along with the human-centric nature of granular computing as discussed in Sect. 2.2.

Dimension 3: Clustering processes. Both methods work with generalized versions of classical k-means and do not require specific clustering approaches.

– *Type of cluster algorithm*. We use an extension of k-means clustering; fuzzy c-means or rough k-means, respectively.
– *Flow of data*. New data arrive in batches where the periodicity or batch size can be determined given the data's structure or have to be set context-dependently.

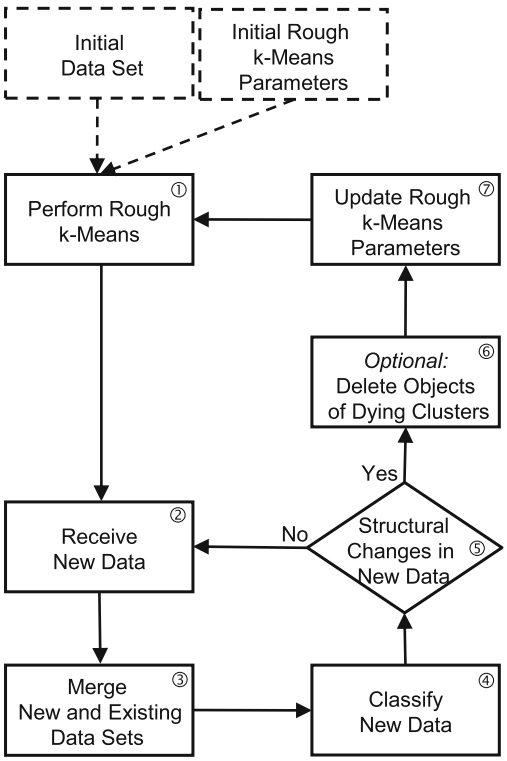

Fig. 10 Dynamic rough clustering cycle (Peters et al. 2012)

– *Implemented dynamics*. The respective base algorithm is nested into an overall updating scheme, as shown, e.g., in Fig. 10.

A main advantage of both, dynamic fuzzy c-means as well as dynamic rough k-means, is that they are very flexible and do not require additional concepts that would go far beyond basic k-means. In both cases, the respective classic version, i.e., fuzzy c-means and rough k-means, are nested into a dynamic shell.

4.2 Evolving DDAA clustering

Despite its name, Dynamic Data Assigning Assessment (DDAA) (Georgieva and Klawonn 2008) has been proposed to cluster static data sets. Its dynamism corresponds to the process how clusters are found in such static data sets. An enhancement, the evolving DDAA clustering method, however, has been proposed for dynamic data sets. Its categorization into the DCC is described next.

Dimension 1: Characteristics of change. The proposed evolving DDAA clustering algorithm can detect the following changes in streaming data:

– Movement of existing clusters,
– Creation of new clusters, and
– Merger of clusters.

Dimension 2: Types of granulation With respect to this dimension, we obtain the following details.

– *Input data*. In general, numeric input data are used that allow calculating distances.
– *Preprocessing*. No particular preprocessing for granulation purposes is performed.
– *Cluster approach*. Crisp or fuzzy clustering has been proposed by Georgieva and Klawonn (2008).

Dimension 3: Clustering processes. Incoming data streams are treated with a crisp, respectively, a fuzzy clustering method.

– *Type of cluster algorithm*. The static version of the DDAA clustering algorithm is based on a crisp, respectively, fuzzy objective function, similar to the corresponding version of the c-means algorithm. The difference, however, is that for the given number of clusters (denoted c) only $c - 1$ are used to detect 'good' clusters while the final cluster candidate contains so-called 'noise points'.
– *Flow of data*. The evolving DDAA clustering algorithms treat incoming data streams, i.e., single newly arriving data points.
– *Implemented dynamics*. Both static versions, crisp as well as fuzzy DDAA clustering algorithm are nested into an evolving clustering procedure where the respective model parameters are iteratively updated.

4.3 Functional fuzzy c-means

Joentgen et al. (1999) proposed functional fuzzy c-means (FFCM) to cluster trajectories of feature values, i.e., the dynamism is considered via feature functions instead of feature values. Such feature functions describe the development of a feature's value during a 'relevant' past. It has to be decided context-dependently what 'relevant' means in a particular application.

Dimension 1: Characteristics of change. Functional fuzzy c-means is a clustering algorithm that is built to cluster a data set where each object is described by feature trajectories rather than feature values. Data are acquired once, i.e., no newly arriving data are considered. Changes that might have occurred prior to data acquisition could be identified by analyzing the respective trajectories.

Dimension 2: Types of granulation. Clustering feature trajectories rather than feature vectors has several consequences for granulation.

– *Input data*. Same as in fuzzy c-means, functional fuzzy c-means needs numeric input values for each object-describing feature. The difference, however, is that it does not only take the most recent snapshot data, but

the trajectory of each feature value over the relevant past.
- *Preprocessing*. Granulation takes place during preprocessing where similarities of trajectories are established by the corresponding fuzzy sets that determine how similar two trajectories are.
- *Cluster approach*. Based on the before-mentioned similarity measure a distance is calculated that is used in FFCM to cluster objects that are described by trajectories. Each cluster is characterized by its center which again is a vector of feature trajectories instead of feature values.

Dimension 3: Clustering processes. Similar to Dimension 2, having trajectories as observations requires also specific characteristics of the clustering process as will be shown next.

- *Type of cluster algorithm*. The cluster algorithm is inspired by standard fuzzy c-means, but applied to feature trajectories instead of feature values.
- *Flow of data*. FFCM takes a static set of objects as input. In its basic version no newly arriving data are considered.
- *Implemented dynamics*. The way how dynamic elements are treated in FFCM is via trajectories of feature values instead of static snapshots. This is treated explicitly in the above described preprocessing step where a fuzzy set determines similarity among trajectories.

4.4 CluStream

CluStream (Aggarwal et al. 2003) is an extension of the BIRCH algorithm (Gama 2012) which has been designed to cluster data streams. It stores the relevant information in so-called cluster features (CF) or micro-cluster, which are compact representations of a set of points.

Dimension 1: Characteristics of change. CluStream recognizes changes in data streams by comparing incoming data points (observations) with a solution that has initially been determined using k-means. Based on the distances to the respective centroids of existing CF, a new data point is absorbed by already established CF or builds a new micro-cluster.

Dimension 2: Types of granulation. The second dimension of DCC addresses the types of granulation. For CluStream we identify the following characteristics.

- *Input data*. CluStream receives numerical inputs via streaming data.
- *Preprocessing*. Granulation is defined by the user who determines the time intervals for updating the stored information regarding a particular solution.
- *Cluster approach*. The distances between a new object and centroids of existing micro-clusters are calculated efficiently based on the particular tree structure used to store the relevant information.

Dimension 3: Clustering processes. Finally, we address DCC's third dimension, the clustering processes.

- *Type of cluster algorithm*. The cluster algorithm is based on standard k-means and uses a tree structure to store and manage incoming data efficiently.
- *Flow of data*. The algorithm has been developed especially to treat data streams where single observations arrive with very high velocity.
- *Implemented dynamics*. CluStream applies an efficient tree structure to administer incoming observations.

Despite the fact, that CluStream does not use any particular method from granular computing, it offers various opportunities to apply some of the respective techniques to create information granules. Examples are the determination of the time intervals used and/or the decision to update a given solution which is based on distances to the centroids of existing micro-cluster.

The examples above show that there already exists a considerable number of methods for dynamic granular clustering. The proposed DCC-Framework helps to detect gaps and, therefore, has the potential to support researchers working in this area.

5 Selected areas of application

In this section, we present selected applications of dynamic granular clustering that have been reported in literature.

5.1 Dynamic clustering of supermarket transactions

An ever increasing, tough competition in the retail industry calls for anticipation of customers' needs and requirements. Clustering the respective transactions provides important insights into consumer behavior (Lingras et al. 2003; Peters et al. 2012). While static analyses can often explain past purchases, dynamic clustering has the potential to uncover changing demand pattern which lead to modified purchase pattern. If, e.g., in a supermarket customer behavior changes during a day, data gathered at the point-of-sales system (POS) reflect these drifts. An initial solution obtained, e.g., based on transactions during the morning hours could be updated as new transactions occur during the day. Changes in a cluster solution, such as moving, emerging, or dying clusters reveal modified consumer preferences and thus give hints on how to adapt advertisements dynamically in order to provide customers

with the most suited information in each moment. Experiments with real-life retail transactions have underlined this potential for the case of supermarkets (Lingras et al. 2003; Peters et al. 2012).

5.2 Clustering messages in social media

Social media generate continuously new data, e.g., in short text messages which reflect what certain community members are concerned about: e.g., Papadopoulos et al. (2012) propose methods to analyze such messages in the context of marketing or crime detection, to name just a few. While these analyses are mostly static, it can be interesting to apply dynamic concepts as presented in the DCC-Framework. In such cases, messages need to be transformed from text to numeric vectors which represent the respective information granules. Then clustering can be applied to group similar messages together. Updating the identified clusters can provide important information to the respective decision-makers, such as changing consumer behavior in marketing applications or 'dynamic hot-spots' (Herrmann 2015) in crime analytics.

5.3 Clustering gene expression data

Information from genes can be employed to synthesize a functional gene product, e.g., a protein. The process by which such information is used for a synthesis is called gene expression. The respective data sets are characterized by typically few observations and many attributes. Ben-Hur and Guyon (2003) suggested to cluster such genes after feature extraction via principal component analysis which constitute the information granules. They have shown that posterior clustering provides important insights to better understand the information describing the respective genes. While this application has been performed on a static data set, applying the appropriate cluster methods on dynamic gene expression data can capture evolving phenomena.

6 Conclusion

Dynamic aspects and granular information processing have received a lot of attention recently, both in the scientific community as well as in industry. Clustering is one of the most important tasks in data mining with a long and successful record of real-life applications. Today, clustering comprises of many different methods and extensions. Consequently, *dynamic methods*, *granular computing*, and *clustering* constitute important techniques in data mining. What was still missing, however, was a structured presentation of existing approaches in the area that merges these three aspects of data mining, i.e., dynamic granular clustering.

To fill this gap, we proposed the DCC-Framework. The Dynamic Clustering Cube can be regarded as an information granule itself that helps to make dynamic granular clustering more accessible and transparent by categorizing this field in an illustrative way. The analysis of the cube's three dimensions—(1) *Characteristics of Change*, (2) *Types of Granulation*, and (3) *Clustering Processes*—provides valuable insights into the corresponding phenomena. The DCC-Framework is not only a scheme for presenting and structuring already existing dynamic granular clustering approaches. Even more importantly, it helps to identify 'white spots' in research in the area of dynamic granular clustering that need to be filled.

Our analysis of selected methods and the discussion of exemplary applications underline the increasing potential of dynamic granular clustering. The DCC-Framework may support to further methodically enhance this field and may also motivate to use dynamic granular clustering in new application areas.

Acknowledgments Support from the Millennium Science Institute on Complex Engineering Systems (http://www.isci.cl; ICM: P-05-004-F, CONICYT: FBO16) and Fondecyt (1140831) is gratefully acknowledged

References

Aggarwal C, Han J, Wang J, Yo P (2003) A framework for clustering evolving data streams. In: Proceedings of the international conference on very large data bases. Morgan Kaufmann, Berlin, pp 81–92

Bargiela A, Pedrycz W (2003) Granular computing: an introduction. Kluwer Academic Publishers, Boston

Bell DE, Raiffa H, Tversky A (1988) Decision making: descriptive, normative, and prescriptive interactions. Cambridge University Press, New York

Ben-Hur A, Guyon I (2003) Chapter: Detecting stable clusters using principal component analysis. In: Functional genomics: methods and protocols. Humana Press, New York, pp 160–189

Ben-Hur A, Horn D, Siegelmann HT, Vapnik V (2001) Support vector clustering. J Mach Learn Res 2(12):125–137

Berkhin P (2006) A survey of clustering data mining techniques. In: Kogan J, Nicholas C, Teboulle M (eds) Grouping multidimensional data. Springer, Berlin, pp 25–71

Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York

Crespo F, Weber R (2005) A methodology for dynamic data mining based on fuzzy clustering. Fuzzy Sets Syst 150(2):267–284

Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery in databases. AI Mag 17(3):37–54

Gacek A, Pedrycz W (2015) Clustering granular data and their characterization with information granules of higher type. IEEE Trans Fuzzy Syst 23(4):850–860

Gama J (2012) A survey on learning from data streams: current and future trends. Prog Artif Intell 1(1):45–55

Gartner Inc., Gartner says solving 'Big Data' challenge involves more than just managing volumes of data (Press Release) (2011).

(http://www.gartner.com/newsroom/id/1731916, retrieved April 21, 2015)
Georgieva O, Klawonn F (2008) Dynamic data assigning assessment clustering of streaming data. Appl Soft Comput 8:1305–1313
Herrmann ChR (2015) The dynamics of robbery and violence hot spots. Crime Sci 4(33):1–14
Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323
Joentgen A, Mikenina L, Weber R, Zimmermann H-J (1999) Dynamic fuzzy data analysis based on similarity between functions. Fuzzy Sets Syst 105(1):81–90
Jolliffe IT (2002) Principal component analysis. Springer, New York
Klimatabelle.info. Klimatabelle Deutschland (2015). (http://www.klimatabelle.info/europa/deutschland, retrieved April 19, 2015)
Kreinovich V (2008) Interval computation as an important part of granular computing: an introduction. In: Pedrycz W, Skowron A, Kreinovich V (eds) Handbook of granular computing. Wiley, Chichester, pp 1–31
Krishnapuram R, Keller JM (1993) A possibilistic approach to clustering. IEEE Trans Fuzzy Syst 1(2):98–110
Kroha P, Baeza-Yates R, Krellner B (2006) Text mining of business news for forecasting. In: 17th International workshop on database and expert systems applications, 2006. DEXA '06, pp 171–175
Li XY, Sun JX, Gao GH, Fu JH (2011) Research of hierarchical clustering based on dynamic granular computing. J Comput 6(12):2526–2533
Lingras P, West C (2004) Interval set clustering of web users with rough k-means. J Intell Inf Syst 23:5–16
Lingras P, Hogo M, Snorek M, Leonard B (2003) Clustering supermarket customers using rough set based Kohonen networks. In: 14. International symposium on methodologies for intelligent systems (ISMIS (2003) LNCS, vol 2871. Springer, Berlin, pp 169–173
MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: Fifth Berkeley symposium, pp 281–297
Mulaik SA (2009) Foundations of factor analysis. Statistics in the social and behavioral sciences, 2nd edn. Chapman & Hall/CRC, Boca Raton
Papadopoulos S, Kompatsiaris Y, Vakali A, Spyridonos P (2012) Community detection in Social Media—performance and application considerations. Data Min Knowl Discov 24:515–554
Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11:341–356
Pedrycz W (1998) Shadowed sets: representing and processing fuzzy sets. IEEE Trans Syst Man Cybern Part B: Cybern 28:103–109
Pedrycz W (2005) Knowledge-based clustering: from data to information granules. Wiley, Hoboken
Pedrycz W (2007) Granular computing—the emerging paradigm. J Uncertain Syst 1(1):38–61
Pedrycz W (2013) Granular computing: analysis and design of intelligent systems. CRC Press/Francis Taylor, Boca Raton
Pedrycz W, Bargiela A (2012) An optimization of allocation of information granularity in the interpretation of data structures: toward granular fuzzy clustering. IEEE Trans Syst Man Cybern Part B: Cybern 42(3):582–590
Pedrycz W, Skowron A, Kreinovich V (eds) (2008) Handbook of granular computing. Wiley, Chichester
Peters G (2014) Rough clustering utilizing the principle of indifference. Inf Sci 277:358–374
Peters G, Weber R (2012) Dynamic clustering with soft computing. WIREs Data Min Knowl Discov 2(3):226–236
Peters G, Weber R, Nowatzke R (2012) Dynamic rough clustering and its applications. Appl Soft Comput 12:3193–3207
Peters G, Crespo F, Lingras P, Weber R (2013) Soft clustering—fuzzy and rough approaches and their extensions and derivatives. Int J Approx Reason 54(2):307–322
Salton G, McGill MJ (1986) Introduction to modern information retrieval. McGraw-Hill, New York
Saltos R, Weber R (2015) Rough-fuzzy support vector domain description for outlier detection. In: 2015 IEEE international conference on fuzzy systems (FUZZ-IEEE), Istanbul. IEEE
Xu R, Wunsch D (2005) A survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678
Yao YY (2004) Granular computing. Comput Sci (Ji Suan Ji Ke Xue) 31:1–5
Yao YY (2005) Perspectives of granular computing. In: Proceedings 2005 IEEE international conference on granular computing (GrC 2005), vol 1, pp 85–90
Yao YY, (2008a) Granular computing: past, present, and future. In: Rough set and knowledge technology (RSKT 2008). LNAI, vol 5009. Springer, Berlin, pp 27–28
Yao YY (2008b) A unified framework of granular computing. In: Pedrycz W, Skowron A, Kreinovich V (eds) Handbook of granular computing. Wiley, Chichester, pp 401–410
Zadeh L (1965) Fuzzy sets. Inf Control 8(3):338–353
Zadeh L (1997) Information granulation and its centrality in human and machine intelligence. In: Grahne G (ed) Proceedings of the 6. Scandinavian conference on artificial intelligence (SCAI'97). Frontiers in artificial intelligence and applications, vol 40. IOS Press, Amsterdam, pp 26–27
Zadeh L (2006) Generalized theory of uncertainty (GTU)—principal concepts and ideas. Comput Stat Data Anal 51(1):15–46