Granular meta-clustering based on hierarchical, network, and temporal connections

Pawan Lingras¹ · Farhana Haider¹ · Matt Triff²

Received: 23 July 2015 / Accepted: 4 November 2015 / Published online: 5 January 2016
© Springer International Publishing Switzerland 2015

Abstract In granular computing, each object is represented as an information granule and an information granule can be connected to other granules through semantic relationships. These connections can lead to a granular hierarchy or a network. Data mining of one set of objects may not be able to capture information contained in granular connections. This paper describes a concept of meta-clustering that clusters a set of granules using clustering information from another or the same set of networked granules. Cluster membership of one granule can affect another granule's cluster membership, resulting in a recursive meta-clustering process. We illustrate the usefulness of such meta-clustering for a granular hierarchy consisting of sets of businesses and reviewers, a set of networked granules representing mobile phone users, and trading patterns of financial instruments that are linked to each other through a temporal dimension.

Keywords Meta-clustering · Granular computing · Iterative clustering · k-means · Fuzzy c-means · Social networks · Time series · Financial markets · Web mining

1 Introduction

The conventional data mining process begins with the representation of objects based on raw data from a dataset. More refined data mining activities use results from previous data mining activities to improve the quality of the results. Examples of such secondary data mining techniques include ensemble of classifiers or stacked regression, where the results of previous classifications/predictions are combined to produce more accurate results. The combination of the results can be based on a predetermined formulation or could use further machine learning techniques. This paper describes a novel set of integrated secondary data mining approaches to clustering. Clustering is one of the most frequently used unsupervised data mining techniques for grouping similar objects. It is used at various stages in data mining from preliminary exploration of a new dataset and identification of outliers to sophisticated analysis for decision making. The proposals presented in this paper enhance the conventional clustering techniques for granular, network, and temporal data.

Granular computing is an emerging area of research that provides an ability to create innovative representations of objects. Such representations can facilitate the development of new algorithms (Bargiela and Pedrycz 2002; Pedrycz and Kreinovich 2008; Yao 2010a; Zadeh 1979). In granular computing, a granule represents an object associated with a set of information. For example, a customer with certain purchasing patterns could represent an information granule. A granule can include a collection of finer granules. For example, a customer granule could include many visits, which are finer granules. A visit in turn can include the purchase of a number of products, which are even finer granules. This results in a hierarchy of customers–visits–products. Profiles of customers created by

✉ Pawan Lingras
pawan.lingras@gmail.com

Farhana Haider
farhanahdr@gmail.com

Matt Triff
matt.triff@gmail.com

¹ Saint Mary's University, Halifax, NS B3H 3C3, Canada

² Canada Border Services Agency, Ottawa, ON, Canada

clustering should also include the profiles of visits that these customers make. The profiling of visits should in turn include profiles of customers. Thus, profiles of products should be both influenced by, and should influence, profiles of customers and visits. Similar granular hierarchies exist in other datasets. For example, at yelp.com a business has multiple reviewers, while a reviewer reviews multiple businesses. The set of business granules are thus connected to the set of reviewer granules. We describe an iterative clustering technique that iterates back and forth through a granular hierarchy to obtain a stable set of profiles of objects at all levels of the hierarchy.

Similar interdependency can also be observed in a networked environment, where objects such as phone users are connected to other phone users. In such a case, the profile of a phone user should include the profiles of other users created by the same clustering process. These dependencies are applicable to any social network. This paper presents a recursive clustering technique for such networked environments. The recursive clustering developed for a networked environment can also be extended to temporal databases, where profiles of daily patterns of a quantity should include profiles of previous daily patterns, and in some cases profiles of future daily patterns. For example, let us assume that we need to profile a stock based on its volatility. The volatility in a stock price today should take into account volatility of stock prices in the immediate past and immediate future. One can look at such a daily pattern as an object that is connected to the past and future daily patterns. Extension of the recursive clustering for a simple network into temporal networks will be applicable in such a case.

This paper illustrates the versatility of the meta-clustering algorithm for different types of granular connections, namely, two-way hierarchical connections between businesses and reviewers at yelp.com, a social network of mobile phone users, and a temporal network of daily stock patterns. The proposed meta-clustering approach can be applied to both regular and fuzzy clustering. This fact is demonstrated through regular and fuzzy meta-clustering of the yelp.com dataset.

2 Review of literature

2.1 Review of clustering and fuzzy clustering

This section reviews conventional clustering with the help of a popular algorithm called *k*-means (MacQueen 1967). Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ be a finite set of objects, and we assume that the objects are represented by m -dimensional vectors. A clustering scheme groups n objects into

k clusters $C = \{\mathbf{c}_1, \dots, \mathbf{c}_k\}$. Here, C is the set of clusters. Each of the clusters $\mathbf{c}_i$ is represented by an m -dimensional vector, which is the centroid or mean vector for that cluster. Each cluster centroid $\mathbf{c}_i$ is also associated with a set of objects assigned to the i th cluster. We will use $\mathbf{c}_i$ for both the centroid vector or the set representation of i th cluster depending on the context.

2.1.1 Clustering using *k*-means

k-means clustering is one of the most popular statistical clustering techniques (Hartigan and Wong 1979; MacQueen 1967). The objective is to assign n objects to k clusters. The process begins by randomly choosing k objects as the centroids of the k clusters. The objects are assigned to one of the k clusters based on the minimum value of the distance $d(\mathbf{x}_i, \mathbf{c}_i)$ between the object vector $\mathbf{x}_i$ and the cluster vector $\mathbf{c}_i$.

After the assignment of all the objects to various clusters, the new centroid vectors of the clusters are calculated as:

$$\mathbf{c}_i = \frac{\sum_{\mathbf{x}_l \in \mathbf{c}_i} \mathbf{x}_l}{|\mathbf{C}_i|}, \quad \text{where } 1 \leq i \leq k.$$

Here $|\mathbf{C}_i|$ is cardinality of cluster $\mathbf{C}_i$. The process stops when the centroids of the clusters stabilize.

Several cluster validity indices have been proposed to evaluate cluster quality obtained by different clustering algorithms. An excellent summary of various validity measures can be found in Halkidi et al. (2002). Many of the cluster validity measures are functions of the sum of within-cluster scatter to between-cluster separation. The scatter within the i th cluster, denoted by S_i , and the distance between cluster $\mathbf{c}_i$ and $\mathbf{c}_j$, denoted by d_{ij} , are defined as follows:

$$S_i = \frac{1}{|\mathbf{C}_i|} \sum_{\mathbf{x} \in \mathbf{c}_i} \text{distance}(\mathbf{x}, \mathbf{c}_i) \quad (1)$$

$$d_{ij} = \text{distance}(\mathbf{c}_i, \mathbf{c}_j) \quad (2)$$

where $\mathbf{c}_i$ is the center of the i th cluster, $|\mathbf{C}_i|$ is the number of objects in $\mathbf{C}_i$, and $\text{distance}(\mathbf{x}, \mathbf{y})$ is the distance between two vectors. Depending upon the application, we can choose any distance function. Two popular distance functions are Euclidean distance and the inverse of cosine similarity function. This study uses Euclidean distance. However, it will also be interesting to experiment with other distance measures including the Mahalanobis distance that is particularly useful when we are working with a dataset that represents only a sample of the universe.

We can sum up the within-cluster scatter for all clusters in a clustering scheme C as:

$$S(C) = \sum_{i=1}^k S_i \quad (3)$$

Similarly, between-cluster distance for a clustering scheme can be summed as:

$$D(C) = \sum_{i=1}^k \sum_{j=1}^k d_{ij} \quad (4)$$

It is advisable to plot both of these measures for the datasets under study. Usually, the scatter within cluster starts rising rapidly, while distance between cluster starts falling rapidly when the number of clusters falls below a certain value. The knee of the curves (Lingras et al. 2014) can be used as the range for determining an appropriate number of clusters.

2.1.2 Fuzzy *c*-means clustering

Conventional clustering assigns various objects to precisely one cluster. A fuzzy generalization of clustering uses a fuzzy membership function to describe the degree of membership of an object to a given cluster (ranging from 0 to 1; where a greater value signifies a greater degree of membership) (Coppi and D'Urso 2002; Coppi and D'Urso 2003; Coppi et al. 2012; D'Urso and DeGiovanni 2014; D'Urso et al. 2014). There is a stipulation that the sum of fuzzy memberships of an object to all the clusters must be equal to 1.

The algorithm was first proposed by Dunn (1973). Subsequently, a modification was proposed by Bezdek (1981). The fuzzy *c*-means (FCM) algorithm is based on minimization of the following objective function:

$$\sum_{i=1}^n \sum_{j=1}^k u_{ij}^m d(\mathbf{x}_i, \mathbf{c}_j), \quad 1 < m < \infty \quad (5)$$

where n is the number of objects and each object is a d dimensional vector. A parameter m is any real number greater than 1, u_{ij} is the degree of membership of the i th object ($\mathbf{x}_i$) in the cluster j , and $d(\mathbf{x}_i, \mathbf{c}_j)$ is the Euclidean distance between the object and a cluster center $\mathbf{c}_j$.

The degree of membership given by a matrix $\mathbf{u}$ for objects on the edge of a cluster, may have a lesser degree than objects in the center of a cluster. However, the sum of these coefficients for any given object x_i is defined to be 1.

$$\sum_{j=1}^k u_{ij} = 1 \quad \forall i \quad (6)$$

The centroid of a fuzzy cluster is the weighted average of all objects, where the weights of each object is its degree of membership to a cluster:

$$\mathbf{c}_j = \frac{\sum_{i=1}^n u_{ij}^m \mathbf{x}_i}{\sum_{i=1}^n u_{ij}^m} \quad (7)$$

FCM is an iterative algorithm that terminates if

$$\max \left(\left| u_{ij}^{t+1} - u_{ij}^t \right| \right) < \delta \quad (8)$$

where δ is a termination criterion between 0 and 1, and t is the iteration step.

2.2 Review of granular computing

Granular computing encompasses multiple levels or layers of granularity in thinking, problem solving, and information processing (Yao 2010a). A granular set can be represented in the form of five-tuple array, (U, D, L, H, J) , where U is the universe of the problem discussed, D describes all the elements in U , L and H are the operators of the opposite direction, and J restricts the L and H (Li 2009). Initially, the main stream of granular computing research focused on fuzzy sets, rough sets, interval analysis and cluster analysis (Pedrycz and Kreinovich 2008; Yao 2007a, 2008b). In the context of fuzzy sets, Zadeh introduced the notion of information granulation in 1979 (Zadeh 1979). In 1982, Pawlak introduced the theory of rough sets using partitions induced by equivalence relations, which can be considered a specific type of granulation. Zadeh further elaborated on information granulation and its central role in human reasoning (Zadeh 1997), which provided new insights into granular computing. A more elaborate formulation of granular computing followed Zadeh's paper in the form of a book by early pioneers Bargiela and Pedrycz. They provided an elegant pyramidal information processing paradigm for granular computing (Bargiela and Pedrycz 2002).

Artificial intelligence (AI), fuzzy sets, and rough sets continue to be the primary motivations behind developments in granular computing (Yao 2008b, c). A comprehensive study of AI perspectives on granular computing is provided by Yao (2011). For example, some of the central concepts and strategies of granular computing can be seen in sub-areas of artificial intelligence, such as concept formation, categorization, learning, abstraction and reformulation, computer vision and image processing, planning, and hierarchical problem solving (Yao 2011). An overview of the current research in granular computing can be found in a handbook and a number of edited books (Pedrycz 2005; Pedrycz and Chen 2011; Pedrycz and Kreinovich 2008; Yao 2010a, 2009).

Yao's research expanded the granular computing perspective, with investigations of concrete theoretical foundations and resulting models that led to a general triarchic theory. The triarchic theory of granular computing provides the philosophical basis, followed by algorithmic developments leading to practical computational techniques (Yao 2007b, 2008a, b, 2009, 2010b). Yao's proposal provides a

conceptual framework and an architecture for developing granular computing as an independent area of research as opposed to a collection of related theories such as fuzzy and rough set theories. Each granular structure is a hierarchical construct that provides a multilevel representation and understanding of a problem. Such a structure forms the basis of the meta-clustering algorithm proposed in this paper. A family of hierarchical structures can help us understand a problem from multiple points of view (Yao 2009). The granular computing triangle consists of three essential elements of granular computing: philosophy, methodology/algorithms, and computational aspects. This new paradigm of granular computing will help us become better problem solvers by designing and implementing intelligent systems based on granular reasoning, representation, and processing (Yao 2010a).

Yao's investigations on granular computing also provide an insight into, a structured understanding of, and structured methods for solving, real-world problems from multiple views and at multiple levels of abstraction within each view (Chen and Yao 2008; Luo and Yao 2011, 2010a, 2007b, 2009). (Yao 2002, 2003a, b) demonstrated the potential value of granular computing in intelligent systems, as well as a test bed for playing with ideas from granular computing. Hoeber used a granular approach for studying retrieval support through visualization in web information retrieval support systems (Hoeber 2008). Yan et al. demonstrated the effectiveness of a granular information retrieval system by implementing and evaluating a prototype system (Yan et al. 2011). Xie et al. applied granular computing for a conceptual biology research supporting platform (Xie et al. 2008).

Gacek and Pedrycz (2015) developed a comprehensive and systematic approach to show an emergence of granular information of higher type, which are used to implement granular interval prototypes. They discussed a way of forming granular data in the context of representation of time series. Pedrycz and Bargiela (2012) considered a concept of granular prototypes that generalizes the numeric representation of the clusters and helps to capture more details about the data structure. Using a granulation-degranulation scheme, they designed granular prototypes being reflective of the structure of data to a higher extent than the representation that is provided by their numeric counterparts (prototypes).

2.3 Review of simultaneous and meta-clustering

Researchers have found it advantageous to cluster multiple objects from a dataset at the same time. Representing a dataset as a matrix, data miners normally consider rows to be the objects and columns as the attributes of these

objects. A good example of such a dataset is a document collection or corpus having n rows and m columns. Each row in the table corresponds to a document. Each column corresponds to a keyword. A cell in the i th row and j th column is the frequency of keyword _{j} in document doc _{i} . However, one can easily transpose this view and say that it is a collection of keywords that shows how often the keyword occurs in various documents in the collection given by the columns in the table. Slonim and Tishby (2000) proposed a two stage clustering method for this application. El-Yaniv and Souroujon (2001) extended this two stage approach with an iterative version where the resulting document clustering could be used to re-cluster the words and the process would continue. More generically, double clustering can be viewed as a dimensionality reduction technique that replaces the columns by groups of the columns. Castellano et al. (2002) further generalized double clustering using fuzzy set theory. Caruana et al. (2006) showed that the use of meta-clustering, meaning clusters of clusters, can make it easier for the users to see the more meaningful groupings. Ramirez-Cano et al. (2010) took meta-clustering to three levels for grouping players in a game based on three different criteria: skills, preferences while playing the game, and relationships with other players.

The generalization of meta-clustering can be found when bi-clustering. Bi-clustering was first introduced by Mirkin (1996). It was extended to tri-clustering and then more generally to n -clustering (Ignatov et al. 2012, 2013; Gnatyshak et al. 2012, 2013). These efforts were based on Formal Concept Analysis (FCA). In bi-clustering, clustering was done row-wise and column-wise simultaneously to determine the intersecting regions. The extension of the matrix to a three-dimensional cuboid gave rise to tri-clustering. Tri-clustering was shown to be useful for a number of applications including Bibsonomy and a crime dataset. Ignatov et al. (2013) described how the tri-clustering can efficiently handle large connected social graph involving three sets of objects.

3 Generalized and unified granular view of meta-clustering

The review of literature in the previous subsection shows that meta-clustering is an important and useful research direction. The majority of the research consists of analyzing matrices or cuboids of related objects. In the next section, we will briefly review a generalized view of meta-clustering that unifies the conventional clustering from static information in a dataset with dynamic information that is generated through simultaneous clustering of a

related set of objects. Static information is the original data used for the clustering. Dynamic information represents the relation of each of the static objects with the cluster result obtained. We have called this information dynamic as this data changes with respect to the cluster results in each iteration.

In granular computing, a granule represents an object associated with a set of information. For example, a customer with certain purchasing patterns could represent an information granule. A granule can include a collection of finer granules. For example, a customer granule could include finer granules containing information for many individual visits. A visit, in turn, can include the purchase of a number of products, which are even finer granules. This results in a hierarchy of customers–visits–products. Profiles of customers created by clustering should also include the profiles of visits that these customers make. The profiling of visits should in turn include profiles of customers. Similarly, profiles of products should be both influenced by, and should influence, profiles of customers and visits. Lingras et al. (2014) described how such a hierarchy can be clustered iteratively with the help of static information and the dynamically changing profiles of customers and products throughout the meta-clustering process. This approach unified the conventional static clustering with the simultaneous meta-clustering such as double clustering using granular computing.

The proposed granular meta-clustering does not require the presence of a matrix or cuboid consisting of multiple sets of objects. Similar interdependency can also be observed in a networked environment, where objects such as phone users are connected to other phone users within the same dataset. In such a case, the profile of a phone user should include the profiles of other users created by the same clustering process. These dependencies are applicable to any social network. Lingras and Rathinavel (2012) proposed a recursive clustering technique for such networked environments.

The recursive clustering method developed for a networked environment can also be extended to temporal databases, where profiles of daily patterns of a quantity should include profiles of previous daily patterns, and in some cases profiles of future daily patterns. For example, let us assume that we need to profile a stock based on its volatility. The volatility in a stock price today should take into account volatility of stock prices in the immediate past and immediate future. One can look at such a daily pattern as an object that is connected to the past and future daily patterns with different connection weights. The weight of the connection will decrease based on temporal differences between the daily patterns. Lingras and Haider (2014b) enhance the recursive clustering originally developed for a simple network for applications in weighted networks in

general, and temporal networks in particular. The basic technique of recursive meta-clustering algorithm is shown in Fig. 1. Figure 2 shows a simple illustration of the process using a similar type of data used in Sect. 4. However, for ease of demonstration only a two-dimensional dummy data set is used in this illustration.

The integrated meta-clustering of hierarchical, network, and temporal data show that the use of granular computing can help us visualize the profiling problem in a broader context and combine the conventional clustering with the emerging simultaneous clustering approaches such as double clustering, bi-clustering, tri-clustering, n -clustering. Moreover, the proposed approach is general enough to work with one, two, three, or n number of sets of objects.

The following sections describe the applications of this meta-clustering algorithm for different types of granular connections:

- two-way hierarchical connections between businesses and reviewers on yelp.com
- social network of mobile phone users
- temporal network of daily stock patterns

The proposed meta-clustering approach can be applied to both regular and soft clustering. This fact is demonstrated through regular and fuzzy meta-clustering of the yelp.com dataset.

Fig. 1 Flow of basic recursive meta-clustering

Fig. 2 Simple illustration of recursive meta-clustering using dummy data

Business b_i				
Reviewer r_i				
Business Data:		Reviewer Data:		
	*	**	*	**
b_1	2	15	r_1	20 5
b_2	12	3	r_2	10 6
...			...	
Step 1: Cluster Business Data. Clusters (bc_k) obtained are bc_1 and bc_2 .				
Step 2: Create dynamic part and concatenate with static part.				
Static Part:		Dynamic Data:		
(Actual data from Reviewer Data)		(Count of businesses in bc_k reviewed by r_j)		
	*	**	bc_1	bc_2
r_1	20	5	3	4
r_2	10	6	10	16
...			...	
Step 3: Cluster the concatenated static and dynamic part. We will have four cluster centers for each resultant cluster. Steps 2 and 3 are repeated until two consecutive dynamic parts of the cluster centers are very similar (the difference below a specified threshold). The centers of the final clusters may look like:				
	*	**	bc_1	bc_2
rc_1	1.5	0.1	3.4	1.2
rc_2	15.0	2.1	3.0	20.0
...				
Now we can create profiles for a reviewer that show their type and the business type they reviewed. For example, rc_1 are reviewers who reviews very rarely, they mostly rate one star and the majority of their reviews are for business type bc_1 .				

4 Meta-clustering in granular hierarchy using yelp.com data

Yelp.com is an online review and recommendation community. Yelp was founded in 2004, is available in 30 countries worldwide, and it currently has over 140 million unique monthly visitors. Yelp provides value to consumers by allowing users to research written reviews, ratings, business details such as business hours and whether or not a business has free WiFi, as well as pictures posted by other users of the business and its products. Yelp also provides a social platform for its users, allowing them to create events, lists of recommended businesses to share and comment on, and to message and become friends with other users.

In Spring 2013, Yelp released a large set of data, covering the entire Phoenix Metropolitan Area (PMA) as part of the Yelp Dataset Challenge. The Yelp Dataset Challenge was open-ended and aimed at finding innovative uses for the data Yelp collects. Yelp posed potential questions to answer, such as “What time of day is a restaurant busy, based on its reviews?”, “What makes a review funny or cool?”, “Which business is likely to be reviewed next by a user?”, and more. Yelp encouraged the submission in any form that entrants felt conveyed the appeal of their project,

which would later be judged for one of ten cash prizes. The data covering the PMA came as four separate files, one each for businesses, check-ins, users, and reviews. Business information included each businesses unique ID, name, neighborhoods that they are located within, full localized address, city, state, latitude, longitude, average star rating out of five (rounded to half stars) from reviewers, categories, and a variable set for whether the business is still active. Reviews contained the business ID of the business being reviewed, the ID of the reviewer writing the review, the number of stars the reviewer gave the business out of five (rounded to the half star), the text of the written review, the date the review was given, and the number of votes other users have given the review, in the categories of “Funny”, “Cool” and “Useful”. Reviewer data contained the unique user ID, first name, number of reviews they have given, average stars rated (as a floating point average of all the reviews they have made), and the total number of votes their reviews have received for the three categories previously mentioned. Finally, check-in data contained information of which business the check-in data related to, and the total number of people who had checked-in to the business on the mobile Yelp app or webpage for each hour of each day of the week (168 categories, or 24 categories for each day of the week).

4.1 Algorithm: meta-clustering in granular hierarchy

In this section, we use k -means and fuzzy c -means clustering algorithms to group similar objects. We ensure that our clusters are as compact as possible and well separated from each other. Since the k -means algorithm depends on randomly selected initial centroids of the clusters, we apply the algorithm multiple times and choose a clustering scheme that has the most compact clusters. Cluster compactness and manual inspection of cluster centroids were used to determine the optimal number of clusters.

On yelp.com, a business is reviewed by many reviewers and a reviewer reviews many businesses creating a bi-directional graph. Our clustering of reviewers uses profiles derived from the clustering of businesses, and vice versa. Let $R = \{r_1, r_2, \dots, r_{nr}\}$ be the set of reviewers and $B = \{b_1, b_2, \dots, b_{nb}\}$ be the set of businesses. Here, $nr = 43,873$ is the number of reviewers and $nb = 11,537$ is the number of businesses in the yelp.com dataset. Furthermore, let $RC = \{rc_1, rc_2, \dots, rc_{kr}\}$ be the clustering scheme of reviewers and $BC = \{bc_1, bc_2, \dots, bc_{kb}\}$ be the clustering scheme of businesses. After studying the compactness of the clusters and resulting centroids, it was decided that the number of reviewer clusters $kr = 7$. The knee of the curve for the within-cluster scatter showed that the scatter starts rising rapidly after $kr = 11$. The increase in the scatter intensifies after $kr = 7$. Similarly, the number of business clusters based on the knee of the curve for scatter within clusters was decided to be 7, i.e., $kb = 7$. The resulting cluster centroids were manually studied to ensure that they were sufficiently apart from each other. For the k -means

clustering, setting the number of clusters to 7 helped group some of the outlying objects into separate clusters. As we will show, using c -means clustering allows for objects to belong to multiple clusters resulting in more moderate centroids. For example, we had a regular cluster with only two prolific reviewers. They were outliers compared to the rest of the reviewers. We do not see such a single cluster with only two reviewers when using fuzzy c -means clustering.

The reviewer r_j is represented by a static data part sr_j and dynamic data part dr_j , i.e., $r_j = (sr_j, dr_j)$ as shown in Table 1. Here, sr_j are the data that are extracted from the raw dataset such as types of reviews (total, *, **, ***, ****, *****, votes). The dynamic part dr_j will be derived from the clustering of businesses. We will represent $dr_j = (m_{j1}, m_{j2}, \dots, m_{jkb})$, where m_{ji} is the normalized count of businesses that the reviewer r_j reviewed that falls in bc_i cluster of businesses, for regular clustering. For fuzzy clustering, m_{ji} is the average membership of businesses that the reviewer r_j reviewed that falls in bc_i cluster of businesses.

Similarly, the business b_i will be represented by a static data part sb_i and dynamic data part db_i , i.e., $b_i = (sb_i, db_i)$ as shown in Table 2. The static part sb_i will represent the number of reviews (total, *, **, ***, ****, *****) received for the business from the raw dataset, while the dynamic part db_i will be derived from the clustering of reviewers. For regular clustering, $db_i = (m_{i1}, m_{i2}, \dots, m_{ikr})$, where m_{ij} is the normalized count of reviewers that reviewed the business b_i that falls in rc_j cluster from the clustering of reviewers. For fuzzy c -means clustering, m_{ij} is the average memberships of reviewers that reviewed the business b_i that falls in rc_j cluster from the clustering of reviewers.

Table 1 Static and dynamic parts of reviewer data

Reviewer ID	Static representation (sr)							Dynamic representation (dr)						
	Total sr ₁	* sr ₂	** sr ₃	*** sr ₄	**** sr ₅	***** sr ₆	Votes sr ₇	bc ₁ dr ₁	bc ₂ dr ₂	bc ₃ dr ₃	bc ₄ dr ₄	bc ₅ dr ₅	bc ₆ dr ₆	bc ₇ dr ₇
r₁	6	0	0	17	33	50	11	0	17	17	17	17	0	33
r₂	9	0	22	0	33	44	6	22	0	67	11	0	0	0
r_i	–	–	–	–	–	–	–	–	–	–	–	–	–	–
r_{nr}	18	6	0	17	67	11	32	11	6	39	28	0	6	11

Table 2 Static and dynamic parts of business data

Business ID	Static representation (sb)						Dynamic representation (db)						
	Total sb ₁	* sb ₂	** sb ₃	*** sb ₄	**** sb ₅	***** sb ₆	rc ₁ db ₁	rc ₂ db ₂	rc ₃ db ₃	rc ₄ db ₄	rc ₅ db ₅	rc ₆ db ₆	rc ₇ db ₇
b₁	67	7	10	12	58	12	3	2	18	5	59	6	8
b₂	4	0	0	25	25	50	50	0	0	0	25	0	25
b_i	–	–	–	–	–	–	–	–	–	–	–	–	–
b_{nb}	18	0	11	11	61	17	13	0	0	0	73	0	13

Fig. 3 Regular and fuzzy meta-clustering algorithms in a granular hierarchy

Procedure: recursiveMetaCluster()

```

dyn ← Empty                                prevCenters.DynCls ← Empty
n ← Number of Records                      k ← Number of Clusters
b ← Dataset of businesses                  r ← Dataset of reviewers

1. for j= 1 to n do step 2
2.   static part  $sb_j = b_j$ , static part  $sr_j = r_j$ 
3.  $bc \leftarrow \text{cluster}(sb)$ ,  $rc \leftarrow \text{cluster}(sr)$ 
4. for j=1 to n do step 5 to step 7
5.   for i=1 to k do step 6
6.     if fuzzy  $mb_{ji}$ =average membership of businesses from  $bc_i$  who were reviewed by  $r_j$ 
7.     else  $mb_{ji}$ =count of businesses from  $bc_i$  who were reviewed by  $r_j$ 
8.     add  $mb_{ji}$  to  $dr_j$ 
9.  $statPlusDynB \leftarrow sb + dr$ 
10.  $centersB \leftarrow \text{cluster}(statPlusDynB)$ 
11. if  $centersB.DynClsPart == prevCenters.DynClsPart$  got to step 14
12.  $prevCenters.DynClsPart \leftarrow centersB.DynClsPart$ 
13. goto step 4
14. for j=1 to n do step 15 to step 17
15.   for i=1 to k do step 16
16.     if fuzzy  $mr_{ji}$ =average membership of reviewers from  $rc_i$  who reviewed businesses  $b_j$ 
17.     else  $mr_{ji}$ =count of reviewers from  $rc_i$  who reviewed businesses  $b_j$ 
18.     add  $mr_{ji}$  to  $db_j$ 
19.  $statPlusDynR \leftarrow sr + db$ 
20.  $centersR \leftarrow \text{cluster}(statPlusDynR)$ 
21. if  $centersR.DynClsPart == prevCenters.DynClsPart$  Stop
22.  $prevCenters.DynClsPart \leftarrow centersB.DynClsPart$ 
23. goto step 15

```

Since we do not have any clustering results at the beginning of the iterative process, we will cluster businesses based solely on the static part. The subsequent clustering of both businesses and reviewers will use the static and dynamic representations. The iterations will stop when values of the dynamic parts dr_j and db_i for all objects stabilize. Figure 3 provides the formal description of the proposed iterative meta-clustering algorithm.

4.2 Comparison of regular and fuzzy static clustering on yelp.com

Table 3 shows the results of clustering applied to the static information about the businesses. We can describe the resulting static profiles of business clusters as:

- sbc₁** *Sparsely but very well rated* Fewest number of reviews, mostly five stars.
- sbc₂** *Sparsely and lowly rated* Fewest number of reviews, mostly one and two stars.
- sbc₃** *Well rated* Modest number of reviews, mostly five and four stars.
- sbc₄** *Ambivalently rated* Modest number of evenly spread reviews.
- sbc₅** *Reasonably rated* Modest number of reviews, mostly four and five stars.
- sbc₆** *Well rated* Large number of reviews, mostly four and five stars with noticeable three stars.

Table 3 Centroids from regular static clustering of business data

Cluster ID	Total	*	**	***	****	*****	Size
sbc₁	5.75	4.99	2.18	2.74	3.92	86.14	2073
sbc₂	5.29	55.80	11.44	7.80	11.13	13.82	1221
sbc₃	11.10	6.45	7.10	11.70	59.22	15.50	2212
sbc₄	12.42	12.61	20.05	34.21	22.61	10.50	2301
sbc₅	13.64	7.04	5.83	9.24	31.04	46.83	2782
sbc₆	101.39	5.92	9.43	16.04	37.45	31.13	844
sbc₇	334.85	3.18	6.43	13.64	38.59	38.13	104

- sbc₇** *Reasonably rated* Largest number of reviews, mostly four and five stars with noticeable three stars.

Table 4 shows the results of the corresponding fuzzy clustering based on the static information about the businesses. One of the major advantages of the fuzzy clustering is the fact that the businesses can belong to multiple clusters. Another interesting feature of fuzzy clustering is that the resulting centroids tend to be less extreme, better separated, and the cluster sizes are more uniformly distributed. The overall clustering profiles match the regular clustering and are given below:

- sbcf₁** *Sparsely but very well rated* Fewest number of reviews, mostly five stars.

Table 4 Centroids from fuzzy static clustering of business data

Cluster ID	Total	*	**	***	****	*****	Size
sbcf₁	5.47	1.83	0.92	1.13	2.36	93.73	1799
sbcf₂	8.77	36.66	14.52	16.19	17.77	14.84	1948
sbcf₃	10.74	6.88	4.89	7.67	29.05	51.47	2246
sbcf₄	12.15	7.74	7.88	14.49	49.96	19.89	2229
sbcf₅	13.43	12.90	14.26	25.06	31.29	16.46	2115
sbcf₆	76.85	6.87	9.73	16.40	37.63	29.34	1032
sbcf₇	262.98	3.83	6.96	14.23	38.72	36.25	168

- sbcf₂** *Sparsely and lowly rated* Few reviews, majority are one and two stars.
- sbcf₃** *Well rated* Modest number of reviews, mostly five and four stars.
- sbcf₄** *Reasonably rated* Modest number of reviews, mostly four and five stars.
- sbcf₅** *Ambivalently rated* Modest number of evenly spread reviews.
- sbcf₆** *Reasonably rated* Large number of reviews, mostly four and five stars with noticeable three stars.
- sbcf₇** *Reasonably rated* Largest number of reviews, mostly four and five stars with noticeable three stars.

Table 5 shows the results of clustering applied to the static information about the reviewers. We can describe the resulting static profiles of reviewer clusters as:

- src₁** *Infrequent and hard* Very few and mostly one and two star reviews.
- src₂** *Infrequent and soft* Very few and mostly five and four star reviews.
- src₃** *Infrequent and very soft* Very few and almost exclusively five star reviews.
- src₄** *Infrequent and balanced* Very few and mostly five star reviews, with noticeable two and three stars reviews as well.
- src₅** *Somewhat prolific and balanced* Modest number of reviews and votes, mostly four, five, and three stars.
- src₆** *Prolific and balanced* Large number of reviews and votes are mostly four, three, and five stars.

Table 5 Centroids from regular static clustering of reviewer data

Cluster ID	Total	*	**	***	****	*****	Votes	Size
src₂	1.60	87.33	0.69	2.29	0.85	8.82	2.39	5154
src₅	2.47	1.55	1.22	3.06	3.65	90.50	3.06	16,569
src₆	2.94	5.64	65.53	6.94	10.10	11.77	4.18	4581
src₄	4.75	3.43	2.70	8.49	69.60	15.75	6.26	14,321
src₃	22.19	2.63	4.34	63.06	17.01	12.94	85.01	3171
src₁	183.04	4.50	9.43	23.14	39.48	23.41	2278.42	75
src₆	442.5	2.27	5.65	26.29	46.04	19.73	13,073.5	2

- src₇** *Extremely prolific and balanced* This group of two is essentially an outlier with a large number of reviews and votes; these users should be treated separately as extremely prolific reviewers.

Table 6 shows the results of fuzzy clustering applied to the static information about the reviewers. The moderating effect of fuzzy *c*-means is more pronounced for the reviewer dataset. The last two clusters **src₆** and **src₇** consisted of a total of 77 reviewers with extremely high values for total reviews and votes. The corresponding fuzzy clusters, **srcf₆** and **srcf₇**, have more moderate centroids and represent more than 14,000 reviewers. The outlying reviewers have been essentially absorbed in the cluster **src₅**. These moderate profiles are possible because these reviewers can belong to multiple clusters. Due to the more pronounced effect of fuzzy clustering, the reviewer fuzzy profiles are somewhat different from their regular clustering counterparts and can be described as:

- srcf₁** *Infrequent and hardest* Very few and mostly one star reviews.
- srcf₂** *Infrequent and soft* Very few and mostly five star reviews.
- srcf₃** *Infrequent and very soft* Very few and almost exclusively five star reviews.
- srcf₄** *Infrequent and hard* Very few and mostly two star reviews.
- srcf₅** *Infrequent middle of the road* Very few and mostly three star reviews.
- srcf₆** *Frequent and somewhat soft* Modest number of reviews and votes.
- srcf₇** *Prolific and balanced* Large number of evenly spread reviews.

In summary, the comparison of regular clustering and fuzzy profiles shows that the fuzzy clustering allows objects to belong to multiple clusters. This assignment to multiple clusters leads to centroids that are well separated and less extreme. Moreover, the objects are more uniformly distributed among all the clusters. However, our profiles provide rather limited information. For example, **sbcf₁** is labeled as *Sparsely but very well rated*, which means the

Table 6 Centroids from fuzzy static clustering of reviewer data

Cluster ID	Total	*	**	***	****	*****	Votes	Size
srcf₆	1.46	97.77	0.40	0.36	0.57	0.88	3.79	4079
srcf₄	1.52	0.34	0.24	0.26	0.39	98.75	2.20	13,842
srcf₃	1.71	0.33	0.44	0.75	97.72	0.74	3.13	6740
srcf₂	1.83	0.76	95.69	0.79	1.36	1.37	5.82	2274
srcf₅	2.12	0.76	0.91	94.47	2.09	1.74	6.40	2223
srcf₆	6.14	4.23	4.41	5.89	40.15	45.30	10.70	7032
srcf₁	11.09	8.99	11.82	18.24	33.88	27.04	22.64	7683

businesses in this cluster have good reviews. But we do not know the type of reviewers. If the reviewers were soft (meaning they tend to grant restaurants inflated ratings), then these high ratings should be somewhat discounted. In the following section, we will use our proposed meta-clustering approach that will provide more enhanced meta-profiles.

4.3 Regular meta-clustering in a granular hierarchy

The meta-clustering algorithm was implemented using a UNIX bash script that iteratively called an R program for clustering and a Python program for creating dynamic

representations of clustering. *k*-means clustering used 1000 iterations and was applied 10 times to get compact clusters. The dynamic representation seemed to stabilize after 21 iterations of indirectly recursive meta-clustering. On a high-performance computing cluster provided by ace-net.ca the meta-clustering took only one minute.

The parallel meta-clustering, described in Fig. 3, created the meta-centroids for various categories of businesses and reviewers. The business clusters obtained from the regular meta-clustering algorithm are shown in Table 7. The meta-centroids of the reviewer clusters obtained from the regular meta-clustering algorithm are shown in Table 8. The last

Table 7 Centroids from regular meta-clustering of business data

Cluster ID	Static representation (sb)						Dynamic representation (db)							Size
	Total sb₁	* sb₂	** sb₃	*** sb₄	**** sb₅	***** sb₆	rc₁ db₁	rc₂ db₂	rc₃ db₃	rc₄ db₄	rc₅ db₅	rc₆ db₆	rc₇ db₇	
bc₁	4.99	6.08	2.56	2.38	6.09	82.89	6.90	1.47	62.52	5.75	20.29	2.67	4	1449
bc₂	5.27	50.80	11.02	9.47	13.64	15.07	42.15	1.69	10.95	5.15	33.74	5.69	0.63	1214
bc₃	10.95	5.69	4.65	6.97	21.73	60.95	5.74	1.42	21.20	4.95	59.59	6.37	0.69	2644
bc₄	11.79	15.55	19.22	31.08	22.30	11.85	10.05	1.17	5.62	3.18	68.81	10.49	0.67	2510
bc₅	13.34	6.08	6.84	12.33	55.02	19.74	4.61	0.98	7.98	3.24	74.85	7.84	0.50	2764
bc₆	100.98	5.92	9.43	15.98	37.39	31.28	3.66	1.81	3.04	20.22	64.67	6.25	0.36	852
bc₇	334.86	3.19	6.43	13.65	38.60	38.13	1.70	21.76	3.80	7.39	59.70	5.38	0.27	104

Table 8 Centroids from regular meta-clustering of reviewer data

Cluster ID	Static representation (sr)							Dynamic representation (dr)							Size
	Total sr₁	* sr₂	** sr₃	*** sr₄	**** sr₅	***** sr₆	votes sr₇	bc₁ dr₁	bc₂ dr₂	bc₃ dr₃	bc₄ dr₄	bc₅ dr₅	bc₆ dr₆	bc₇ dr₇	
rc₁	1.83	62.86	17.70	11.55	2.87	5.02	2.48	3.86	22.65	12.28	26.37	11.76	21.15	1.93	6908
rc₂	2.09	6.37	7.71	9.81	29.08	47.03	3.11	0.53	0.74	2.44	2.40	2.38	7.29	84.22	5064
rc₃	2.14	1.79	1.51	1.56	3.87	91.28	2.45	28.98	3.33	34.09	8.73	15.83	5.89	3.16	8922
rc₄	2.70	3.63	13.08	12.23	6.31	64.75	3.63	0.99	0.67	3.13	2.45	3.26	84.01	5.49	8805
rc₅	9.57	3.19	6.14	11.70	64.20	14.77	24.93	1.21	1.77	11.33	12.41	21.41	42.16	9.71	14,094
rc₆	180	4.54	9.46	22.71	39.44	23.84	2235	1.24	2.28	11.64	17.28	18.17	36.13	13.26	78
rc₇	442	2.28	5.66	26.29	46.05	19.73	13073	2.12	4.14	17.35	18.76	16.41	31.57	9.64	2

column in each table shows the size of each cluster. The resulting business profiles are more refined than conventional clustering process as they use associations with the profiles of the reviewers that reviewed the business. We can describe these enhanced profiles obtained from regular meta-clustering as follows:

- bc₁** *Sparsely but very well rated by softies* Fewest number of reviews mostly five stars, most coming from *rc₃* and *rc₅*, which tend to give mostly four and five stars reviews.
- bc₂** *Sparsely and lowly rated by both hard and soft groups* Fewest number of reviews, mostly one and two stars, most coming from *rc₁* (gives one stars reviews) and *rc₅* (gives four stars reviews).
- bc₃** *Well rated by softies* Modest number of reviews, mostly five and four stars, most coming from *rc₅* and *rc₃*, which tend to give mostly four and five stars reviews.
- bc₄** *Ambivalently rated even by softies* Modest number of evenly spread reviews, most coming from *rc₅*, which tends to give mostly four stars reviews.
- bc₅** *Reasonably rated by mostly softies* Modest number of reviews, mostly four and five stars, most coming from *rc₅*, which tends to give mostly four stars reviews.
- bc₆** *Well rated by balanced reviewers* Large number of reviews, mostly four and five stars with noticeable three stars, most coming from *rc₅* (gives mostly four stars) and *rc₄* (capable of giving two and three stars)
- bc₇** *Reasonably rated by many softies* Largest number of reviews, mostly four and five stars with noticeable three stars, most coming from *rc₅* (gives mostly four stars) and *rc₂* (gives mostly five and four stars).

The association of reviewer information with a business cluster is inversely applicable to the reviewer profiles, which are refined using the profiles of the businesses who are reviewed by these reviewers. These augmented reviewer profiles can be described as:

- rc₁** *Infrequent and hard, cover the spectrum* Very few and mostly one and two star reviews. The reviews are spread evenly across most business clusters.
- rc₂** *Infrequent and soft, cover popular and favorites* Very few and mostly five and four star reviews. Almost all reviews are for *bc₇* which has very large number of four and five stars.
- rc₃** *Infrequent and very soft, cover the spectrum* Very few and almost exclusively five star reviews, spread evenly across most business clusters.

- rc₄** *Infrequent and balanced, cover favorite businesses* Very few and mostly five star reviews, with noticeable two and three stars reviews as well. Most reviews are for *bc₆*, which has large number of four and five stars.
- rc₅** *Somewhat prolific and balanced, cover almost all the spectrum* Modest number of reviews and votes, mostly four, five, and three stars. They do not have too many reviews for *bc₁* and *bc₂* (which do not receive too many reviews).
- rc₆** *Prolific and balanced, cover popular places* Large number of reviews and votes are mostly four, three, and five stars. They do not have too many reviews for *bc₁* and *bc₂* (which do not receive too many reviews).
- rc₇** *Extremely prolific and balanced, do not cover most and least popular* This group of two is essentially an outlier with a large number of reviews and votes, and these users should be treated separately as prolific reviewers. Their reviews are mostly four, three, and five stars. They do not have too many reviews for *bc₁* and *bc₂* (which do not receive too many reviews), and *bc₇* which receives most reviews.

4.4 Fuzzy meta-clustering in a granular hierarchy

The business clusters obtained from the fuzzy meta-clustering algorithm are shown in Table 9. The meta-centroids of the reviewer clusters obtained from the fuzzy meta-clustering algorithm are shown in Table 10. The last column in each table shows the size of each cluster. One of the first differences between regular and fuzzy meta-clustering of businesses is the fact that they are somewhat more uniformly distributed among seven clusters. The last two clusters with a large number of reviews are larger in size with more moderate values for total number of reviews. The resulting business meta-profiles are somewhat different from the corresponding regular meta-clustering profiles. We can describe these enhanced profiles as follows:

- bcf₁** *Sparsely but very well rated by softies* Fewest number of reviews, mostly five stars, most coming from softies from *rcf₂*.
- bcf₂** *Sparsely and lowly rated by both hard and soft groups* Few reviews, majority are one and two stars, very few from somewhat hard reviewers *rcf₄*, and *rcf₅*.
- bcf₃** *Well rated by softies* Modest number of reviews, mostly five and four stars, noticeable reviews from the softest reviewers *rcf₂*.

Table 9 Centroids from fuzzy meta-clustering of real business data

Cluster ID	Static representation (sb)						Dynamic representation (db)							Size
	Total sbf ₁	* sbf ₂	** sbf ₃	*** sbf ₄	**** sbf ₅	***** sbf ₆	rcf ₁ dbf ₁	rcf ₂ dbf ₂	rcf ₃ dbf ₃	rcf ₄ dbf ₄	rcf ₅ dbf ₅	rcf ₆ dbf ₆	rcf ₇ dbf ₇	
bcf₁	5.49	1.86	0.93	1.14	2.40	93.66	0.06	0.43	0.06	0.05	0.05	0.19	0.16	1803
bcf₂	8.80	36.51	14.55	16.25	17.84	14.85	0.17	0.13	0.11	0.08	0.08	0.19	0.23	1950
bcf₃	10.78	6.93	4.93	7.72	29.05	51.37	0.09	0.22	0.11	0.07	0.07	0.21	0.21	2244
bcf₄	12.16	7.77	7.90	14.52	49.89	19.92	0.09	0.14	0.15	0.08	0.09	0.21	0.23	2229
bcf₅	13.42	12.93	14.24	25.00	31.32	16.50	0.11	0.13	0.13	0.09	0.10	0.21	0.24	2108
bcf₆	76.34	6.90	9.74	16.41	37.63	29.32	0.08	0.15	0.13	0.08	0.08	0.22	0.24	1034
bcf₇	262.17	3.84	6.97	14.24	38.73	36.22	0.07	0.16	0.13	0.07	0.08	0.23	0.24	169

Table 10 Centroids from fuzzy meta-clustering of real reviewer data

Cluster ID	Static representation (sr)							Dynamic representation (dr)							Size
	Total srf ₁	* srf ₂	** srf ₃	*** srf ₄	**** srf ₅	***** srf ₆	Votes srf ₇	bcf ₁ drf ₁	bcf ₂ drf ₂	bcf ₃ drf ₃	bcf ₄ drf ₄	bcf ₅ drf ₅	bcf ₆ drf ₆	bcf ₇ drf ₇	
rcf₁	1.47	97.75	0.41	0.37	0.58	0.90	3.82	0.07	0.20	0.13	0.14	0.17	0.21	0.07	4079
rcf₂	1.53	0.35	0.25	0.27	0.41	98.72	2.23	0.19	0.08	0.16	0.11	0.11	0.22	0.14	13,839
rcf₃	1.72	0.34	0.45	0.77	97.67	0.76	3.17	0.05	0.09	0.12	0.16	0.14	0.27	0.17	6740
rcf₄	1.84	0.77	95.63	0.81	1.39	1.40	5.89	0.06	0.12	0.12	0.14	0.16	0.27	0.13	2275
rcf₅	2.14	0.78	0.93	94.38	2.13	1.78	6.47	0.05	0.11	0.11	0.14	0.17	0.27	0.15	2224
rcf₆	6.21	4.32	4.50	6.01	39.94	45.23	10.85	0.09	0.09	0.13	0.13	0.13	0.27	0.17	7059
rcf₇	11.07	8.96	11.78	18.15	33.98	27.13	22.60	0.07	0.10	0.13	0.14	0.14	0.27	0.16	7657

- bcf₄** *Reasonably rated by everyone* Modest number of reviews, mostly four and five stars from evenly spread spectrum of reviewers.
- bcf₅** *Ambivalently rated by everyone* Modest number of evenly spread reviews from evenly spread spectrum of reviewers.
- bcf₆** *Reasonably rated by hard reviewers* Large number of reviews, mostly four and five stars with noticeable three stars, few coming from hardest reviewers *rcf₁*, *rcf₄*, and *rcf₅*.
- bcf₇** *Reasonably rated by everyone* Largest number of reviews, mostly four and five stars with noticeable three stars from evenly spread spectrum of reviewers.

Similar to regular meta-clustering, the association of reviewer information with a business cluster is inversely applied to the reviewer profiles. The resulting fuzzy meta-profiles are refined using the profiles of the businesses who are reviewed by these reviewers. The moderating effect of fuzzy *c*-means is more pronounced for the reviewer dataset. Similar to when we clustered on the static representations above, the last two clusters **rc₆** and **rc₇** consisted of a total

of 80 reviewers with extremely high values for total reviews and votes. The corresponding fuzzy clusters, **rcf₆** and **rcf₇**, have more moderate centroids and represent more than 14,000 reviewers. The outlying reviewers have been essentially absorbed in the cluster **rc₅**. These moderate profiles are possible because these reviewers can belong to multiple clusters. The augmented reviewer fuzzy meta-profiles are somewhat different from their regular clustering counterparts and can be described as:

- rcf₁** *Infrequent and hardest, cover low rated businesses* Very few and mostly one star reviews, noticeable reviews for *bcf₂*.
- rcf₂** *Infrequent and soft, cover popular and favorites* Very few and mostly five star reviews, noticeable reviews for well-reviewed business clusters *bcf₁* and *bcf₃*.
- rcf₃** *Infrequent and very soft, cover the spectrum* Very few and almost exclusively five star reviews, spread evenly across clusters *bcf₃* to *bcf₇*.
- rcf₄** *Infrequent and hard, cover favorite businesses* Very few and mostly two star reviews, more of them for well-reviewed business cluster *bcf₆* and evenly reviewed business cluster *bcf₅*.

- rcf₅** *Infrequent middle of the road, cover popular places* Very few and mostly three star reviews more of them for well-covered business clusters bcf_6 and bcf_7 .
- rcf₆** *Frequent and somewhat soft, cover popular places* Modest number of reviews and votes, mostly five and four stars, more of them for well-covered business clusters bcf_6 and bcf_7 .
- rcf₇** *Prolific and balanced, cover popular places* Large number of evenly spread reviews, more of them for well-covered business clusters bcf_6 and bcf_7 .

4.5 Comparison of static and meta-regular/fuzzy clustering

In this section we first analyzed experimental results of regular and fuzzy clustering with static data from the database. The regular clustering is especially important to identify the initial clustering parameters such as the number of clusters. However, in most real-life applications the clusters are not very well defined and the boundaries between the clusters are often fuzzy. Therefore, in most practical applications it may be necessary to refine our clustering using fuzzy set theory. The application of fuzzy c -means to our static representation showed that the fuzzy clustering refines the results in two aspects:

1. Fuzzy memberships of the clusters allow an object to be defined as a member of multiple clusters with varying membership.
2. Fuzzy centroids tend to be more moderate because a large number of objects contribute to the centroid calculations to a varying degree.

Parallel meta-clustering of reviewers and businesses influence each other's profiling during the clustering process itself. The influence of the business profiles on reviewer representation and vice versa can be controlled using different weighting strategies. As we were primarily interested in the static representations of both the businesses and reviewers, we chose to emphasize this aspect in our weighting strategy. As a result, centroids from the static clustering are reasonably similar to the static parts of the meta-clustered centroids. This means that the addition of the dynamic part does not radically change the original profiles or object memberships. However, subtle changes to the object memberships, especially when using fuzzy meta-clustering help us better describe profiles of individual objects. The meta-clustering helps us enhance these profiles.

5 Meta-clustering in granular networks using mobile phone data

In a networked environment information granules can be dependent on each other for a more complete representation. For example, in a mobile phone network, granules such as phone users are connected to other phone users. In such a case, the profile of a phone user should include the profiles of other users created by the same clustering process. The information granule in the proposed meta-clustering algorithm is represented by a static and dynamic portions (s_j, d_j) . The static portion s_j consists of attributes derived from the primary source of data such as the average duration of a phone call, number of phone calls, type of phone calls, and temporal information about the phone calls. The dynamic portion d_j consists of the profiles of the destination phone users. These profiles of other phone users need to be recursively derived from the clustering process itself. For the first meta-clustering iteration, we only use the static representation. From second iteration onwards, the dynamic part is derived from the cluster memberships of the previous iteration. As the clustering scheme evolves, the dynamic part of the information granule keeps on changing through every iteration. The meta-clustering process stops when values of dynamic representation d_j for all objects stabilize. While the proposed recursive meta-clustering is applicable to any social network, the effectiveness of the proposed recursive meta-clustering algorithm is demonstrated here for a network of phone users. Mobile phone call mining is an emerging area of research. Phone call records of a group of customers is a great source of information for user modeling. Calling patterns can be useful for suggesting appropriate modifications to the customer's calling plan. The proposed integrated analysis of all the users in a granular network can be useful for the phone companies to better manage their resources as well as to maximize utilization and profits.

5.1 Algorithm: meta-clustering in granular networks

We will describe our proposal using phone numbers as our objects. However, the algorithm can be applied to any network of objects. Let pn_j be the j th phone number and let us represent pn_j by a static data part s_j and dynamic data part d_j , i.e., $pn_j = (s_j, d_j)$. If k is the number of clusters, then $d_j = (m_{j1}, m_{j2}, \dots, m_{jk})$, where m_{jk} is the normalized count of phone numbers that pn_j is calling, which falls in the k th cluster from the previous iteration. The normalization for the $m_{j,k}$ is done by the sum of all the $m_{j,k}$'s. As

the clustering scheme evolves, d_i keeps on changing through every iteration. For the i th iteration, let us represent the corresponding quantities by a superscript i .

Then,

$$s_j^i = s_j^{i-1} = s_j^0 \quad (9)$$

$$d_j^i = (m_{j1}^{i-1}, m_{j2}^{i-1}, \dots, m_{jk}^{i-1}) \quad (10)$$

where m_{jk}^{i-1} is the normalized count of destination numbers belonging to the k th cluster called by pn_j in $(i-1)$ th iteration.

$$pn_j^i = (s_j^i, d_j^i) = (s_j^0, d_j^i) \quad (11)$$

Figure 4 provides a more formal and concise description of the proposed recursive meta-clustering algorithm. The following is a brief explanation of the application of the algorithm to the mobile phone call data set.

We first normalize the data by dividing the values of each of the seven attributes by the mean value for that attribute. This ensures that all the seven attributes are more or less equally weighted. Determining the appropriate number of clusters is one of the major challenges in any clustering analysis. This issue is resolved by applying k -means clustering to the normalized phone number data for different values of k . We used the minimization of within-cluster scatter and maximizing between-cluster separation as competing objectives to determine the optimum number of phone number clusters using Davies–Bouldin (DB) index (Davies and Bouldin 1979). We found the optimum number of phone number clusters to be five.

As the first step in our recursive meta-clustering algorithm, we apply the k -means clustering algorithm to the phone numbers with seven static attributes (as described in the next section). After the initial clustering with static representation involving seven attributes, we introduce five additional columns to the raw data set. These five attributes

will hold the number of destination phone calls that fall into each of the five clusters. Let us elucidate this further. For an origin phone number, we have a set of destination phone numbers. It is possible that some of the destination phone numbers also belong to cluster 1. Similarly, we will have destination phone numbers belonging to clusters 2, 3, 4 and 5. We count the number of such destination phone numbers belonging to cluster 1, 2, 3, 4 and 5 for each of the origin phone numbers. This new information is first normalized by their sum and entered in the five new columns we added to the raw data set. We perform the next iteration of clustering using the new augmented data set. The process is repeated until the five columns corresponding to the normalized cluster membership stabilize.

5.2 Mobile phone network data

This section describes data preparation applied to the original data set provided by Eagle (2010) followed by the design of the experiment.

The objective of the present study is to use recursive clustering to converge to a set of user profiles. The data set comprises of 182,208 phone calls data collected from about 102 users over a period of 9 months. Based on a previous study by Lingras and Butz (2011), we choose the following variables to represent a phone call:

1. Average duration of phone calls
2. Average number of weekend/weekday
3. Average number of daytime/night-time
4. Average number of outgoing/incoming
5. Average number of SMS
6. Average number of voice calls
7. Average number of long duration calls.

5.3 Results of meta-clustering in granular networks

The clustering results can be analyzed in two parts—static and dynamic. The static results are the results derived from the matrix D^0 and the dynamic results are derived from the information added with each iteration. The dynamic results are derived from $[m_1^i | m_2^i | \dots | m_k^i]$.

5.3.1 Static results

The static results correspond to the clustering analysis based on the static part of the data as described earlier. These results are derived from the centers and the sizes of the clusters. The centers of the clusters for the static variables and their sizes are tabulated in Table 11. From the results of the recursive clustering, we can make the following observations.

Procedure: recursiveMetaCluster()
 dyn ← Empty prevCenters.DynCls ← Empty
 n ← Number of Records pn ← dataset of phone numbers
 k ← Number of Clusters

1. for j= 1 to n do step 2
2. static part $s_j = pn_j$
3. $c \leftarrow \text{cluster}(s)$
4. for j=1 to n do step 5 to step 7
5. for i=1 to k do step 6
6. $m_{ji} = \text{count of phone numbers from } c_i \text{ called by } pn_j$
7. add m_{ji} to d_j
8. statPlusDyn ← s + d
9. centers ← cluster(statPlusDyn)
10. if centers.DynClsPart == prevCenters.DynClsPart
11. Stop
12. prevCenters.DynClsPart ← centers.DynamicClsPart
13. goto step 4

Fig. 4 Meta-clustering algorithm in a granular network

Table 11 The cluster centers and sizes corresponding to the static part of the data at the 2000th iteration

Cluster no	NumCalls	Duration	Weekend 1 weekday 0	Day 1 night 0	Out 1 in 0	SMS1	Voice call 1	Long duration 1	Cluster size
1	0.598	0.724	0.881	1.154	1.070	0.547	1.102	0.333	34
2	2.367	0.799	0.881	0.797	0.961	1.280	0.936	0.630	17
3	0.561	1.290	1.166	0.995	0.981	0.836	1.037	1.560	24
4	1.079	0.658	1.130	0.923	0.863	2.461	0.669	0.541	11
5	1.013	2.912	1.116	0.824	1.043	0.696	1.068	5.109	5

Note that the values have been rounded off to 3 decimal places

In each iteration, the cluster sizes were approximately 5, 24, 11, 17 and 34 with slight variations of plus or minus 4. Even though the combination of the clusters having these sizes change from iteration to iteration, the pattern of the above-mentioned cluster sizes does not change. This indicates that the clustering is fairly strong and the group of numbers that are in a particular cluster tend to move together into a different cluster in a different iteration.

Profile of Cluster 1 These users make low number of calls, low average duration calls, low weekend calls, highest daytime calls, highest outgoing calls, least SMS calls, highest voice calls and the fewest long duration calls.

Profile of Cluster 2 This cluster is made up of phone numbers which make the highest number of calls, low average duration of calls, low weekend calls, least daytime calls, low number of outgoing calls, moderate SMS calls and moderate number of voice calls.

Profile of Cluster 3 This cluster is made up of phone numbers that make the least number of calls, low average duration of calls, low weekend calls, moderate daytime calls, moderate outgoing calls, moderate SMS calls and high number of voice calls.

Profile of Cluster 4 This cluster is made up of phone numbers that make moderate number of calls, least average duration of calls, high number of weekend calls, moderate daytime calls, least outgoing calls, highest SMS calls, least voice calls and the least number of long duration calls.

Profile of Cluster 5 This cluster is made up of phone numbers that made moderate number of calls, highest

average duration of calls, high weekend calls, low daytime calls, high outgoing calls, low SMS calls, high voice calls and the highest number of long distance calls.

The above inferences have been summarized in Table 12.

5.3.2 Dynamic results

The dynamic results correspond to the clustering analysis based on the dynamic part of the data as described earlier. These results are derived from the centers and the sizes of the clusters. The results of this section are given below.

The result of the clustering is an augmented table of the seven static attributes from the raw data and the five normalized $m_{j,k}^i$ that provide the dynamic attributes. The normalization is such that the $m_{j,k}^i$ gives a measure of the probability of the destination numbers belonging to a particular cluster (1, 2, 3, 4 or 5) for each of our 91 phone numbers.

The observations from the resulting five dynamic attributes are as follows:

1. The phone numbers which have a closer social circle are the phone numbers which have a high $m_{j,k}^i$ and the phone numbers with a more diverse social group have a low $m_{j,k}^i$.
2. Most of the numbers have very less probability that their calls will belong to a particular cluster formed by the given 91 phone numbers. This is an expected result

Table 12 The summary of static profiles

Parameter	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Number of calls	Low	Highest	Lowest	High	High
Average duration	Low	Low	Moderate	Low	Highest
Weekend calls	Low	Low	Highest	High	High
Daytime calls	Highest	Low	Moderate	Moderate	Low
Outgoing calls	High	Moderate	Moderate	Moderate	High
SMS calls	Lowest	Moderate	Moderate	Highest	Low
Voice calls	Highest	Moderate	High	Lowest	High
Long duration calls	Lowest	Low	Moderate	Low	Highest

Table 13 The cluster centers corresponding to the dynamic part of the data at the 2000th iteration

Cluster number	$m_{j,1}^i$	$m_{j,2}^i$	$m_{j,3}^i$	$m_{j,4}^i$	$m_{j,5}^i$
1	0.008	0.017	0.047	0.002	0.010
2	0.005	0.010	0.022	0.000	0.005
3	0.011	0.001	0.012	0.001	0.005
4	0.023	0.008	0.011	0.000	0.009
5	0.003	0.000	0.000	0.000	0.001

Note that the values have been rounded off to 3 decimal places

because the given data set is not a closed group of phone number. This means that most phone calls are made mostly to phone numbers that fall outside the given 91 phone numbers.

- However, we see spikes at some phone numbers. These are the phone numbers that make and receive a good number of calls from phone numbers which belong to a cluster composed of phone numbers from the data set. The biggest spike is the second phone number—this phone number shows a probability of 1 for its calls being from a single cluster. The corresponding phone logs for this number were checked and it is indeed true that all calls are being made to a particular cluster.
- Overall the basic structure of the graph does not change much over the iterations.

The cluster centers for the dynamic part of the data can also be similarly analyzed. The cluster centers at the end of the 2000 iterations are given in the Table 13.

From Table 13, it can be seen that the centers show some differentiation between the clusters even for a recursive clustering scheme. Each row gives us the cluster center for each cluster and each column gives us a dimension. The following general remarks can be made with regard to the above table.

Sociability or Row-wise Analysis Since the rows are actually cluster centers, each of the values in the row is the value of the cluster center along a particular dimension. If a cluster center of cluster l is high with respect to a particular dimension $m_{j,k}^i$, it means that phone numbers belonging to cluster l frequently communicate with the phone numbers of cluster k . We can extend this concept to mention that for cluster l , if the values along most dimensions are high (i.e., most of the values are high along the row), then cluster l is a very social cluster and it is in contact with most of the clusters. It should be noted that if the value of a cluster center along dimension l (i.e., $m_{j,l}^i$) is very high and the value along dimension k (i.e., $m_{j,k}^i$) is very low, it only means that the data set corresponding to this cluster is

comprised of a majority of high dimension l (i.e., $m_{j,l}^i$) values and has a minority of high dimension k (i.e., $m_{j,k}^i$) values. It does not exclude the existence of high values of dimension 2 from the data set categorically. Hence, the row-wise values reflect a comparative profile of the calls belonging to the cluster.

Popularity or Column-wise Analysis The popularity values (column-wise values) are indicative of how important a dimension (i.e., $m_{j,l}^i$) is for the cluster centers for all the clusters. As seen in the Table 13, $m_{j,2}^i$ (i.e., the column of cluster 2) has high values for clusters 1, 2 and 4. It means that $m_{j,2}^i$ is an important dimension for clusters 1, 2 and 4. However, all values along the column $m_{j,1}^i$ are high. This means that $m_{j,1}^i$ is an important dimension for all the clusters making cluster 1 a very popular and important cluster for all other clusters.

Please note that sociability and popularity are independent of each other. Hence, it is possible that a cluster l is social with cluster k but is not popular with cluster k . The results of the clustering as shown in Table 13 in terms of sociability and popularity are summarized below.

- Cluster 1 is the most popular cluster because the dimension $m_{j,1}^i$ has none of its values close to 0. Cluster 1 is also a very social cluster as its row-wise values are high.
- Cluster 2 is very popular among the phone numbers from the clusters 1, 2 and 4. Cluster 2 is also fairly social except with phone numbers from cluster 4.
- Cluster 3 is very popular with all the clusters except cluster 5. Cluster 3 is also social with all clusters.
- Cluster 4 is the least popular cluster because the dimension $m_{j,4}^i$ is nearly 0 for all clusters. Also, cluster 4 phone numbers are social with all the clusters except their own cluster. These phone numbers indicate people who are very selective of the people they communicate with.
- Cluster 5 phone numbers are the least social phone numbers. They socialize only with themselves and with phone numbers from cluster 1. But they are popular phone numbers and all clusters communicate with them.

From the above observations, we can conclude that while certain phone users tend to concentrate their destination numbers to a particular group of people (who fall within the same cluster because of their inherent calling behavior), others are more diversely networked. We can also distinguish between their sociability and popularity characteristics which can help to build a sophisticated model of the social network represented by the data set.

6 Meta-clustering in granular temporal environment

This paper proposes a novel recursive approach to temporal clustering in a granular environment. An information granule represents an object. For example, a stock in a financial market may be interesting to a trader if it is volatile on a given day. In that case, we can represent the stock using the price distribution during the day. In such a granular temporal environment, a daily pattern is connected to historical and future daily patterns. Traditionally, clustering of granules is done in isolation without any information on clustering of the connected granules. Such a clustering will only allow us to create the profile of a stock based on daily volatility. However, a trader will typically want to know how long the stock has been volatile to figure out where the stock is in terms of its volatility cycle. The proposed recursive temporal meta-clustering algorithm enhances the representation of a daily pattern with the clustering information of the daily patterns of the same stock from recent history. The clustering of such an enhanced representation is iterative. Each iteration uses results of the previous clustering of historical temporal patterns, until a stable clustering of all the patterns is achieved. These repeated applications of clustering are called meta-clustering because we use clustering information from the previous iteration to modify the representation of the granules. We demonstrate the application of the proposed approach using daily patterns of a set of 223 financial instruments (stocks/bonds/commodities) tracked over a period of 121 days. The resulting meta-profiles augment the daily volatility profile with historical volatility for the same financial instrument.

6.1 Algorithm: meta-clustering in granular temporal environment

The algorithm for the proposed Recursive Temporal Meta-clustering is represented in Fig. 5. The primary objective of the temporal meta-clustering is to recursively provide a historical perspective of the clustering. For example, let us consider daily patterns of a number of stocks that are being traded in a financial market. We want to create a profile of the daily pattern of volatility using clustering. However, profiling a stock only on 1 day's daily price pattern will not tell the trader if the stock is in an early or late stage of an unusual interest in the market. Therefore, we want to use volatility profiles of the recent history of a stock for creating a volatility profile of the stock on a given day. However, these historical volatility profiles require the clustering of the stock's pattern for the recent days in the same clustering algorithm, leading to a recursive profiling.

Procedure: recursiveMetaCluster()

```

dyn ← Empty          prevCenters.DynCls ← Empty
m ← Period of Dynamic Part  noCls ← No of Clusters
1. stat ← createStaticPart(m+1)
2. statPlusDyn ← stat + dyn
3. centers ← cluster(statPlusDyn)
4. rankedCentersStat ← rank(centers.StatClsPart)
5. if centers.DynClsPart == prevCenters.DynClsPart
6.   Stop
7. prevCenters.DynClsPart ← centers.DynamicClsPart
8. dyn ← createDynamicPart(rankedCentersStat, m, noCls)

```

Procedure: createStaticPart(mPlus1)

```

1. for each i = mPlus1 to (NumOfRecords.Item) do step 2
2.   add record[i] to statProfile
3. return statProfile

```

Procedure: rank(centers)

```

1. sort centers in ascending order
2. return centers

```

Procedure: createDynamicPart(rankedCenters, m, noCls)

```

1. for each i = 1 to (NumOfRecords.Item-m) do step 2 to 4
2.   for dy = i to (i+(m-1)) days do step 3
3.     rank[dy]=rankDay(dy,rankedCenters,noCls)
4.   add rank to dynProfile
5. return dynProfile

```

Procedure: rankDay(dy,rankedCenters,noCls)

```

1. Initialize min with -99 for each of (m+1) cells
2. for k = 1 to noCls do step 3 to 6
3.   dis=abs(rankedCenters[k]-record[dy])
4.   if dis<min do step 5 to 6
       [i.e. if most of the values of dist < values of min]
5.     min=dis
6.     rank =k
7. return rank

```

Fig. 5 Recursive temporal meta-clustering algorithm in a granular environment

The two key steps in the algorithm are creation of the static and dynamic parts. A daily pattern of a stock is naturally connected to the daily patterns of the same stock from previous days. It is fair to assume that a sustained activity in a stock does not last for more than 2 weeks (10 trading days). Based on this assumption, we can create a graph where each daily pattern is connected to the daily patterns of the same stock from the previous 10 days. This means representation of a daily pattern has data from that day (obtained statically from the database). This static part consists of a natural logarithm of five percentile values, 10, 25, 50, 75 and 90 % as described in the data processing section. The choice of a natural logarithm is based on the Black Scholes index, which also uses a natural logarithm to moderate the variation in the values. The historical volatility of the same stock over the last ten trading days constitutes the dynamic part of the representation of a daily pattern. More specifically, the dynamic part will use the volatility ranking of the last ten trading days for the same stock based on meta-clustering information. To have

10 days of history available in representation of a daily pattern, our dataset consists of patterns starting from the 11th trading day onwards.

6.2 Financial market data processing

Volatility of Financial Data Series is an important indicator used by traders. The fluctuation in prices creates trading opportunities. Volatility is a measure of the variation in price of a financial instrument over time. Distribution of prices during the day can provide an elaborate description of price fluctuations. A trader finds a daily price pattern interesting when it is volatile. The higher the fluctuations in prices, the more volatile the pattern. Nobel prize winning research introduced us to the concept of Black Scholes index to quantify volatility of a pattern. We can segment daily patterns based on values of the Black Scholes index. This segmentation is essentially a clustering of one dimensional representation (Black Scholes index) of the daily pattern. Black Scholes index is a single concise index to identify volatility in a daily pattern. However, a complete distribution of prices during the day can provide more elaborate information on the volatility during the day. While a distribution consisting of frequency of different prices is not a concise description for a single day, it can be a very useful representation of daily patterns for clustering based on volatility. Lingras and Haider (2014a) described how to create a rough ensemble of clustering using both of these representations. In this paper, we only use the daily price distribution to demonstrate the recursive temporal meta-clustering. However, the proposed approach can use either of the two representation or even an ensemble of the two clustering methods.

Following Lingras and Haider (2014a), we use five percentile values; 10, 25, 50, 75 and 90 % to represent the price distribution. 10 % of the prices are below the 10th percentile value, 25 % of the prices are below the 25th percentile value and so on. Our data set contains average prices at 10 min interval of 223 instruments transacted on 121 days comprising a total of 27,012 records. Each daily pattern has 39 intervals. This data set is used to create a five dimensional pattern, which represents 10, 25, 50, 75 and 90 percentile values of the prices. The prices are normalized by the opening price so that a commodity selling for \$100 has the same pattern as the one that is selling for \$10. Afterwards, natural logarithm of the five percentiles are calculated.

The representation is clustered 10 times using the k -means clustering algorithm. The optimal number of clusters is found by plotting the Davies–Bouldin (DB) index and the scatter in clusters. The aim is to use the number of clusters corresponding to the lowest DB index and the knee of curve of the within-cluster scatter. Based on these two

criteria, we choose five as a reasonable number of clusters (Lingras and Haider 2014a).

Since the natural logarithm values are very small, they are multiplied by 100. This weighting ensures that the small values of the static part of our representation are not dominated by the large values of dynamic part. Examples of the static parts of some of the daily patterns are shown in Table 14.

To create the first dynamic part, we cluster the daily patterns using these static parts. The resulting five clusters are ranked based on their volatility. The higher values of 90th percentile tend to suggest higher volatility. The cluster having lowest volatility is ranked 1, the cluster having next lowest valued centroid is ranked 2 and so on. Ranked clusters with centroid values of the percentiles from the static part are shown in Table 15.

The dynamic part of a daily pattern is created by assuming that the daily pattern is related to the last ten daily patterns for the same stock. We use the last ten volatility rankings for the same stock to make up the dynamic part of the representation of the daily pattern. The dynamic part puts the volatility of stock in historical perspective. We form the dynamic part using the percentile value representation of each of the $m = 10$ days and ranking with the ranks of static part clusters that are closest to them. Examples of dynamic parts created after first clustering with static parts for some of the daily patterns is shown in Table 16.

Table 14 Static part of percentile data

Day:instrument	p10	p25	p50	p75	p90
2011-08-16:3_1	0	1.33	2.61	3.09	3.60
2011-08-17:3_1	0	0	0.17	0.43	0.52
:					
2012-01-31:3_1	0	0	0.67	1.33	2.06
:					
2011-08-16:Z_2	0	0.12	0.20	0.24	0.24
:					
2012-01-31:Z_2	0	0.04	0.08	0.14	0.50

Table 15 Ranked clusters for percentile data after first iteration

Rank	Cluster	Centers				
		p10	p25	p50	p75	p90
1	C2	0	0.09	0.19	0.33	0.46
2	C1	0	0.28	0.60	0.94	1.24
3	C5	0	0.64	1.32	1.94	2.37
4	C4	0	1.31	2.68	3.84	4.55
5	C3	0	3.65	10.21	11.13	11.61

Table 16 Dynamic part after first iteration

Day _{m+1} :instrument	d _{m-9}	d _{m-8}	d _{m-7}	d _{m-6}	d _{m-5}	d _{m-4}	d _{m-3}	d _{m-2}	d _{m-1}	d _m
2011-08-16:3_1	2	2	1	1	2	1	3	1	1	2
2011-08-17:3_1	2	1	1	2	1	3	1	1	2	4
:										
2012-01-31:3_1	1	2	3	1	1	3	4	1	1	3
:										
2011-08-16:Z_2	2	1	1	1	1	1	1	1	1	2
:										
2012-01-31:Z_2	1	2	2	1	3	1	2	3	1	2

Table 17 Concatenated static part (SP) and dynamic part (DP) after first iteration

Day:instrument	SP					DP									
	p10	p25	p50	p75	p90	d _{m-9}	d _{m-8}	d _{m-7}	d _{m-6}	d _{m-5}	d _{m-4}	d _{m-3}	d _{m-2}	d _{m-1}	d _m
2011-08-16:3_1	0	1.33	2.61	3.09	3.60	2	2	1	1	2	1	3	1	1	2
2011-08-17:3_1	0	0	0.17	0.43	0.52	2	1	1	2	1	3	1	1	2	4
:															
2012-01-31:3_1	0	0	0.67	1.33	2.06	1	2	3	1	1	3	4	1	1	3
:															
2011-08-16:Z_2	0	0.12	0.20	0.24	0.24	2	1	1	1	1	1	1	1	1	2
:															
2012-01-31:Z_2	0	0.04	0.08	0.14	0.50	1	2	2	1	3	1	2	3	1	2

Table 18 Cluster centers after clustering with concatenated profile

Rank	Cluster	SP					DP									
		p10	p25	p50	p75	p90	d _{m-9}	d _{m-8}	d _{m-7}	d _{m-6}	d _{m-5}	d _{m-4}	d _{m-3}	d _{m-2}	d _{m-1}	d _m
1	C1	0	0.14	0.30	0.48	0.64	1.34	1.32	1.23	1.20	1.32	1.32	1.32	1.31	1.32	1.35
2	C5	0	0.20	0.43	0.68	0.91	1.67	1.69	2.67	1.53	1.71	1.52	1.80	1.69	1.51	1.60
3	C3	0	0.20	0.43	0.68	0.91	1.51	1.69	1.37	2.65	1.61	1.78	1.56	1.68	1.68	1.51
4	C4	0	0.22	0.47	0.74	0.99	2.15	1.98	1.54	1.52	2.08	2.07	1.99	2.02	2.14	2.20
5	C2	0	0.91	1.91	2.69	3.21	1.68	1.72	1.74	1.72	1.71	1.75	1.78	1.737	1.79	1.77

Static and dynamic parts of Table 14 and Table 16 are concatenated as Table 17 for the next step of the clustering. The resultant profile of clustering the concatenated profile with 15 attributes (5 percentiles and ranks of last 10 days) is shown in Table 18. After every clustering, the dynamic part is updated and the clustering is repeated until the dynamic part converges or until the maximum number of iterations is reached.

6.3 Results of granular temporal meta-clustering

The recursive temporal meta-clustering process described in the previous section was executed for a maximum of 65 iterations. The rounded values of the dynamic parts were compared to test the convergence. While we did not get a

100 % match between two subsequent dynamic parts, a maximum of 98 % match was found for the results of iterations 9 and 10.

Based on these observations on the convergence, we will use the cluster results of the 9th iteration as the final clustering. The final cluster centroids and their ranks are shown in Table 19.

Now that we have the static and dynamic profiles for the days of all the financial instruments (stocks), we can create the meta-profiles of each cluster as follows:

Cluster C4 (Rank 1): Least volatile The stocks in this cluster are not volatile today nor have they shown any volatility for last 2 weeks (10 trading days).

Cluster C3 (Rank 2): Low volatility today, but volatile over last week The stocks in this cluster are not volatile

Table 19 Final ranked centers for percentile data

Rank	Cluster	p10	p25	p50	p75	p90	d _{m-9}	d _{m-8}	d _{m-7}	d _{m-6}	d _{m-5}	d _{m-4}	d _{m-3}	d _{m-2}	d _{m-1}	d _m
1	C4	0	0.17	0.37	0.58	0.78	1.04	1.02	1.05	1.04	1.03	1.04	1.03	1.03	1.06	1.10
2	C3	0	0.20	0.42	0.66	0.87	1.27	1.29	1.46	1.67	1.97	2.35	2.70	2.98	3.01	2.93
3	C2	0	0.20	0.43	0.68	0.90	3.31	3.17	2.71	2.35	1.95	1.56	1.33	1.21	1.20	1.23
4	C5	0	0.25	0.52	0.80	1.04	2.92	3.05	3.18	3.31	3.39	3.34	3.19	2.99	2.86	2.71
5	C1	0	0.91	1.92	2.72	3.24	1.27	1.21	1.25	1.22	1.23	1.19	1.19	1.17	1.16	1.21

today. However, they were volatile last week (5 trading days). The volatility in these stocks may be subsiding and it may be relatively safer to sell them.

Cluster C2 (Rank 3): Moderate volatility today and last week, but volatile 2 weeks ago The stocks in this cluster are somewhat volatile today. They have not shown much volatility last week (5 trading days) either. However, they were quite active 2 weeks ago. The volatility in these stocks has definitely subsided and it may be better to sell them as they are unlikely to rise in the next little while.

Cluster C5 (Rank 4): Moderate volatility today, but volatile for last 2 weeks The stocks in this cluster are not volatile today. However, they were volatile over the last 2 weeks (10 trading days). The volatility in these stocks seems to have come to a screeching halt. It may be good idea to study the news on these stocks and trade accordingly.

Cluster C1 (Rank 5): High volatility today, but was not volatile for last 2 weeks The stocks in this cluster are attracting interest from traders. They may be in the early phase of activity and potential buying opportunities.

In conventional clustering, we can only represent the daily volatility of a stock. The profiles created using only a single days' volatility do not provide a convenient historical perspective of the volatility for a given stock. Since a daily pattern of a stock is naturally related to the daily patterns of the same stock from previous days, a historical cluster representation based on this relation may be more convenient. Thus, the resulting meta-profiles created here, on the basis of such relations, not only describe the current volatility of the stock but whether the stock is at the beginning or the end of a volatility cycle. This may allow traders to look at stocks differently, leading to a more informed decision.

7 Computational requirements for the meta-clustering algorithm

The primary objective of the proposed meta-clustering algorithm is to generate semantically more meaningful profiles based on connections between granules. It is necessary to strike a balance between reliable and useful

profiles versus computational efficiency. The proposed meta-clustering algorithm has inherent opportunities for parallel processing. Therefore, while it will require significant computational resources, they can be distributed among multiple processors resulting in a reasonable chronological time requirement. In this section, we discuss the computational requirements and describe how the algorithm can be parallelized. The implementation of parallel meta-clustering is a separate research topic in itself, and is being investigated as part of our ongoing research.

The problem of obtaining an optimal clustering scheme is NP-hard. Let us assume that there are n objects that need to be grouped into k clusters. Each object can be assigned to any one of the k clusters, resulting in $k \times k \times \dots \times k = k^n$ possible clustering schemes. The clustering scheme that provides minimum scatter within clusters and maximum separation between clusters will then be selected as the optimal one. Therefore, finding the optimal clustering scheme will require $O(k^n)$ calculations of cluster quality. Calculation of cluster quality will require $O(n^2)$ distance calculations.

One of the popular iterative clustering algorithms, k -means, has been found to provide reasonable clustering schemes. Each iteration in k -means requires $O(k \times n)$ distance calculations. Therefore, k -means time requirements are $O(k \times n \times \text{iter})$, where iter is the number of iterations. However, the clustering scheme resulting from k -means depends on the initial choice of cluster centers. Typically, one needs to apply k -means multiple times and choose a clustering scheme that provides minimum scatter within clusters and maximum separation between clusters. Similar observations can be made about the fuzzy c -means algorithm.

The proposed meta-clustering algorithm uses multiple applications of a conventional clustering algorithm such as the k -means or fuzzy c -means. In addition, the resulting clustering schemes will be used to create the dynamic representations for each object. The creation of a dynamic representation will require $O(n_1 \times n_2)$ computations, where n_1 is the number of objects from one granular level (e.g., customers) and n_2 is the number of objects from the other granular level (e.g., products). One set of experiments

reported in this paper used 11,537 businesses and 43,837 reviewers. These experiments used a linear application of the clustering algorithms. The linear implementations will require significant chronological time when the values of n_1 and n_2 are of the order of millions or billions. It is possible to reduce the chronological time in a distributed environment as follows:

1. Apply k -means or fuzzy c -means algorithm in parallel on multiple nodes and choose the clustering scheme with the best quality.
2. While the proposed meta-clustering algorithm waits for clustering of objects from granular level 1 to create dynamic representations for objects from granular level 2, before grouping objects from granular level 2, it is possible to run the clustering schemes from both levels in parallel. The clustering at level 1 will use the currently available clustering scheme from cluster 2. If the clustering scheme from level 2 has not changed, then the clustering at level 1 will wait for the new clustering scheme from level 2. This parallel processing scheme can be implemented through message passing.
3. The creation of dynamic profiles involves sorting and searching lists. There are many parallel implementations of sorting and searching that can be used to facilitate faster computations.

8 Summary and conclusions

Granular computing provides a more meaningful representation of objects that can lead to interesting knowledge discoveries. This paper describes how representing objects and their relationships with each other as a granular network can be used to create recursive meta-profiles using a meta-clustering process. These meta-profiles recursively refer to each other. The approach is demonstrated using a granular hierarchy for businesses who are reviewed by multiple reviewers on a website called yelp.com, for a social network of mobile phone users, and for daily trading patterns of stocks that are linked to each other through time dimension. In addition, the meta-clustering process is shown to be applicable for both regular and fuzzy clustering.

The meta-clusters created for granular hierarchy are represented using meta-profiles. The meta-profiles for yelp.com help us understand the businesses in relation to the associated reviewers and vice versa. The profiles created using regular and fuzzy clustering are somewhat different due to the difference of the object assignment nature of these two algorithms. Fuzzy clustering may provide

more refined results, as it allows an object to be assigned to multiple clusters.

Meta-profiles for a mobile phone network represent different types of mobile phone users with respect to nature of their phone calls and the group of users they communicate with. This profiling could be valuable in deciding appropriate phone package for certain user segments.

The meta-profile created for the financial time series data is a comprehensive historical representation of similar daily patterns, using the daily patterns of the same stock from previous days, in a granular environment. The derived profiles show both the nature of the current volatility, as well as the previous 2 weeks' volatility of a stock, which can help investors make trading decisions.

A study of the computational effort required to implement recursive meta-clustering and parallel processing of the proposed technique are two main streams for further research.

References

- Bargiela A, Pedrycz W (2002) Granular computing: an introduction. Kluwer Academic Publishers, Boston
- Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithms. Kluwer Academic Publishers, Norwell
- Caruana R, Elhawary M, Nguyen N, Smith C (2006) Meta clustering. In: Sixth International Conference on Data Mining, 2006. ICDM'06. IEEE, Hong Kong, China, pp 107–118
- Castellano G, Fanelli AM, Mencar C (2002) Generation of interpretable fuzzy granules by a double-clustering technique. Arch Control Sci 12(4):397–410
- Chen YH, Yao YY (2008) A multiview approach for intelligent data analysis based on data operators. Inf Sci 178(1):1–20
- Coppi R, D'Urso P (2002) Fuzzy k -means clustering models for triangular fuzzy time trajectories. Stat Methods Appl 11(1):21–40
- Coppi R, D'Urso P (2003) Three-way fuzzy clustering models for lr fuzzy time trajectories. Comput Stat Data Anal 43(2):149–177
- Coppi R, D'Urso P, Giordani P (2012) Fuzzy and possibilistic clustering for fuzzy data. Comput Stat Data Anal 56(4):915–927
- Davies D, Bouldin D (1979) A cluster separation measure. IEEE Trans Pattern Anal Mach Intell 2:224–227
- Dunn JC (1973) A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Cybernetics 3(3):32–57. doi:10.1080/01969727308546046
- D'Urso P, De Giovanni L (2014) Robust clustering of imprecise data. Chemom Intell Lab Syst 136:58–80
- D'Urso P, De Giovanni L, Massari R (2014) Self-organizing maps for imprecise data. Fuzzy Sets Syst 237:63–89
- Eagle N (2010) The reality mining data. <http://eprom.mit.edu/data/RealityMining-ReadMe.pdf>
- El-Yaniv R, Souroujon O (2001) Iterative double clustering for unsupervised and semi-supervised learning. Machine Learning: ECML 2001. Springer, Freiburg, Germany, pp 121–132
- Gacek A, Pedrycz W (2015) Clustering granular data and their characterization with information granules of higher type. IEEE Trans Fuzzy Syst 23:850–860. doi:10.1109/TFUZZ.2014.2329707
- Gnatyshak DV, Ignatov DI, Kuznetsov SO (2013) From triadic fca to triclustering: Experimental comparison of some triclustering algorithms. CLA 2013:249

- Gnatyshak D, Ignatov DI, Semenov A, Poelmans J (2012) Gaining insight in social networks with biclustering and triclustering perspectives in business informatics research. Springer, Nizhny Novgorod, Russia, pp 162–171
- Halkidi M, Batistakis Y, Vazirgianni M (2002) Clustering validity checking methods: part II. *ACM SIGMOD Rec* 31:19–27
- Hartigan JA, Wong MA (1979) Algorithm AS136: a k-means clustering algorithm. *Appl Stat* 28:100–108
- Hoerber O (2008) Web Information Retrieval Support Systems: The Future of Web Search. In: *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence—International Workshop on Web Information Retrieval Support Systems*, Sydney, Australia, pp 29–32
- Ignatov DI, Kuznetsov SO, Poelmans J (2012) Concept-based biclustering for internet advertisement. In: *2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW)*, IEEE, Brussels, Belgium, pp 123–130
- Ignatov DI, Kuznetsov SO, Poelmans J, Zhukov LE (2013) Can triconcepts become triclusters? *Int J Gen Syst* (ahead-of-print):1–23
- Li H (2009) Granule, granular set and granular system. In: *IEEE International Conference on Granular Computing*, 2009, GRC'09, IEEE, Nanchang, China, pp 340–345
- Lingras P, Elagamy A, Ammar A, Elouedi Z (2014) Iterative meta-clustering through granular hierarchy of supermarket customers and products. *Inf Sci* 257:14–31
- Lingras P, Butz CJ (2011) Conservative and aggressive rough SVR modeling. *Theor Comput Sci J Sect Theory Nat Comput* (to appear)
- Lingras P, Haider F (2014a) Rough ensemble clustering. submitted for publication
- Lingras P, Haider F (2014b) Temporal recursive meta-clustering. *Inf Sci* (to be submitted)
- Lingras P, Rathinavel K (2012) Recursive Meta-clustering in a Granular Network. In: *Plenary talk at the Fourth International Conference of Soft Computing and Pattern Recognition*, Brunei
- Luo JG, Yao YY (2011) Granular Space State Search. In: *Proceedings of the 24th Canadian Conference on Artificial Intelligence*. St. John's, Canada, pp 285–290
- MacQueen J (1967) Some methods for classification and analysis of multivariate observations. *Proc Fifth Berkeley Symp Math Stat Probab* 1:281–297
- Mirkin B (1996) *Mathematical classification and clustering*. Kluwer Academic Publishers, Boston
- 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* 42:582–590. doi:[10.1109/TSMCB.2011.2170067](https://doi.org/10.1109/TSMCB.2011.2170067)
- Pedrycz W, Chen SM (2011) *Granular computing and intelligent systems design with information granules of higher order and higher type*. Springer, Berlin
- Pedrycz W, Skowron A, Kreinovich V (2008) *Handbook of Granular Computing*. Wiley, New York
- Pedrycz W (2005) *Knowledge-based clustering: from data to information granules*. John Wiley & Sons, New York
- Ramirez-Cano D, Colton S, Baumgarten R (2010) Player classification using a meta-clustering approach. In: *Proceedings of the 3rd Annual International Conference Computer Games, Multimedia and Allied Technology*, Singapore, pp 297–304
- Slonim N, Tishby N (2000) Document clustering using word clusters via the information bottleneck method. In: *23rd annual international ACM SIGIR conference on Research and development in information retrieval*, Athens, Greece, pp 208–215
- Xie Y, Katukuri JR, Raghavan VV, Presti T (2008) *Conceptual biology research supporting platform: current design and future directions. Applications of computational intelligence in biology*. Springer, Berlin
- Yan X, Lau RYK, Song D, Li X, Ma J (2011) Toward a Semantic Granularity Model for Domain specific Information Retrieval. *ACM Trans Inf Syst* 29(3):15. doi:[10.1145/1993036.1993039](https://doi.org/10.1145/1993036.1993039)
- Yao YY (2002) *Information Retrieval Support Systems*. In: *Proceedings of The 2002 IEEE World Congress on Computational Intelligence*, Honolulu, USA, pp 1092–1097
- Yao YY (2003a) A Framework for Web-based Research Support Systems. In: *Proceedings of the 27th Annual International Computer Software and Applications Conference*, Dallas, USA, pp 601–606
- Yao YY (2003b) Granular Computing for the Design of Information Retrieval Support Systems. In: Wu W, Xiong H, Shekhar S (eds) *Clustering and Information Retrieval*. Kluwer Academic Publishers, Boston, pp 299–329
- Yao JT (2007a) A Ten-year Review of Granular Computing. In: *Proceedings 2007 IEEE International Conference on Granular Computing (GrC 2007)*, California, USA, pp 734–739. doi:[10.1109/GrC.2007.11](https://doi.org/10.1109/GrC.2007.11)
- Yao YY (2007b) The Art of Granular Computing. In: *Proceedings of the International Conference on Rough Sets and Intelligent Systems Paradigms*, Warsaw, Poland, pp 101–112
- Yao YY (2008a) A unified framework of granular computing. In: Pedrycz W, Skowron A, Kreinovich V (eds) *Handbook of granular computing*. Wiley, Chichester, pp 401–410
- Yao YY (2008b) *Granular Computing: Past, Present, and Future*. In: *Rough Set and Knowledge Technology (RSKT 2008)*. Springer-Verlag, Berlin, Germany, no. 5009 in LNAI, pp 27–28
- Yao YY (2008c) The Rise of Granular Computing. *Chongqing University of Posts and Telecommunications (Natural Science Edition)* 20(3):299–308
- Yao YY (2009) Integrative levels of granularity. In: Bargiela A, Pedrycz W (eds) *Human-centric information processing through granular modelling*. Springer, Berlin, pp 31–47
- Yao JT (2010a) Human-inspired Granular Computing. In: Yao J (ed) *Novel developments in granular computing: applications for advanced human reasoning and soft computation*. Information Science Reference, Hershey, pp 1–15
- Yao YY (2010b) The Triarchic Theory of Granular Computing (in Chinese). In: YP Z (ed) *Space Quotient. Theory and Granular Computing. Theory and Practice of Structured Problem Solving*, Science Press, Beijing, pp 115–143
- Yao YY (2011) Artificial Intelligence Perspectives on Granular Computing. In: W P, S-M C (eds) *Computing Granular, Systems Intelligent. Design with Information Granules of Higher Order and Higher Type*. Springer, Berlin, pp 17–34
- Zadeh LA (1997) Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst* 90:111–127
- Zadeh LA (1979) Fuzzy Sets and Information Granularity. In: Gupta NRR, Yager R (eds) *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, pp 3–18