PSO Driven Collaborative Clustering: a Clustering Algorithm for Ubiquitous Environments

Benoît Depaire[1], Rafael Falcón[2], Koen Vanhoof[1], and Geert Wets[1]

[1] Data Analysis and Modeling,
Hasselt University,
3590 Diepenbeek, Belgium
{benoit.depaire,koen.vanhoof,geert.wets}@uhasselt.be
[2] Computer Science Department,
Central Univ. of Las Villas (UCLV),
Santa Clara, Cuba 54830
rfalcon@uclv.edu.cu

Abstract. The goal of this article is to introduce two existing clustering approaches into the domain of ubiquitous knowledge discovery. First we demonstrate how horizontal collaborative clustering can be performed in a ubiquitous environment and discuss the ability of these two clustering techniques to cope with privacy constraints. Next, we illustrate how a particle swarm optimization driven version of this clustering algorithm can be used in KDUbiq research and we introduce a fitness functions whose objective is to find similar cluster composition across data locations. Finally, we run an experiment which shows the potential of PSO driven collaborative clustering in a ubiquitous environment with privacy issues

Key words: Ubiquitous Knowledge Discovery, Resource-Awareness, Privacy Restrictions, Horizontal Collaborative Clustering, Particle Swarm Optimization

1 Introduction

Nowadays, computing environments and technologies are more and more evolving towards a mobile, finely distributed, interacting, dynamic environment containing massive amounts of heterogeneous, spatially and temporally distributed data sources. Some typical examples of such ubiquitous computing environments are peer-to-peer systems, grid systems and wireless sensor networks. Knowledge discovery which faces the challenges imposed by these new computing environments is also called ubiquitous knowledge discovery or KDubiq.

KDubiq research has some characteristic features which set it aside from traditional knowledge discovery and distributed data mining. Firstly, KDubiq algorithms have to operate in an environment where both computing power

and data sources can be heavily distributed at a much greater order than in distributed data mining.

Secondly, due to the distributed nature of the environment, communication between the different data sources and computing locations is necessary. A KDubiq environment typically consists of several computing devices which perform some local data mining in situ using limited information at hand while communicating with others.

Thirdly, the computing devices often possess limited computing resources (e.g. sensor network) thus calling for resource aware data mining algorithms.

Fourthly, since data is distributed across several sources, data mining algorithms must be able to cope with privacy and security issues which prevent data from being gathered at a centralized repository.

Last but not least, KDubiq algorithms often have to deal with data streams and must be able to process the data in real-time in contrast to traditional data mining techniques which perform a batch analysis on centralized data. All these characteristics have been studied in depth by the Coordination Action for Ubiquitous Knowledge Discovery and have been reported in the KDubiq Blueprint [1] which will be available as Springer book by Fall 2008. Readers who are interested in a more elaborated discussion of these and other features of KDubiq research may refer to this work. Let us now depict the crucial motivation behind this work and how it fits into the KDubiq realm.

Motivating Example. A company holds information on a set of potential customers and wishes to segment them into different groups which will make it easier to identify potential opportunities and act appropriately. Other companies might also hold information on the same set of potential customers and have a similar need to identify different groups. Obviously, due to privacy, security or business reasons, these companies are unwilling or even prohibited to exchange their information. Yet an overall discovery of common patterns through some collaboration mechanisms enforced over the companies could be highly profitable in contrast to a confined discovery of local knowledge structures (clusters). In some sense, these companies could be regarded as members of a ubiquitous environment where data and computing power is distributed across the different partners which are able to communicate with each other. Therefore, they might benefit from a KDubiq clustering algorithm which allows them to perform the segmentation locally by using local data and actively consider the findings coming from other companies without violating privacy, security or business constraints.

Contributions and outline. In this article, we will discuss how a particle swarm optimized horizontal collaborative fuzzy clustering algorithm, which is a slightly modified version of the algorithm introduced by Falcón et al. [2], matches the problem sketched in the above example and tackles some of the challenges imposed by KDubiq, such as privacy issues and distributed knowledge discovery. The main goal of this article is to study the potential of this clustering approach for applications which are KDubiq in the sense of privacy constraints. Our motivating example does not impose further constraints related to time,

cpu, memory or communication availability. These type of constraints fall beyond the scope of this article. However, we do mention some basic ideas how the presented techniques could be made resource aware in section 2. This article should be considered as a critical test for the viability of the presented clustering algorithms in a KDubiq setting with a main focus on privacy constraints.

The next section is devoted to discuss the original collaborative fuzzy clustering algorithm, introduced by Pedrycz [3, 4] and we shall illustrate how the problem of privacy can be solved. Next, in section 3, we will elaborate on the determination of the values governing the collaborative scheme between the different computing nodes. Next, we will set up an experiment to compare the cluster results from a global cluster approach, a local cluster approach with no collaboration and a local cluster approach with collaboration. Finally, we will discuss some limitations and directions for future research.

2 Collaborative Fuzzy Clustering

In 2002, Pedrycz [3] introduced a novel clustering algorithm, called Collaborative Fuzzy Clustering, which intended to reveal the overall structure of distributed data (i.e. data residing at different repositories) but, at the same time, complying with the restrictions preventing data sharing. It can be stated that this approach exhibits significant differences with other existing techniques under the umbrella of distributed clustering [4]. This clustering approach is an interesting starting point for a KDubiq clustering algorithm.

Generally speaking, two types of collaborative clustering are envisioned, i.e. the horizontal mode and the vertical mode. The vertical mode assumes that each computing location collects and holds information on different objects described in the same feature space. For example, a network of weblog crawlers, which all collect the same information from crawled blog sites, could use the vertical collaborative clustering to cluster the different blogs into a predetermined number of groups. The horizontal mode, on the other side, assumes that each location holds information on the same set of objects but described in different feature spaces, as is the case with the example from section 1. In this article we will focus on the horizontal collaborative clustering scheme and we will integrate the example in the discussion. However, it should be clear that the ideas presented here can be readily extended to the vertical fashion and other examples.

Assume that each company holds different information on the same set of N customers and that they have agreed to arrange them into C clusters. The collaborative scheme starts off with a local fuzzy FCM cluster analysis performed by each company $[ii]$ separately on their local data, although any objective-function-based clustering algorithm can be used. The generic version of the FCM method was proposed by Dunn [5] and Bezdek [6] in the 1980s, but has undergone significant changes over the years. The reader may refer to Hoppner et al. [7] for a comprehensive reference on this topic. FCM identifies C cluster centers and assigns each record k (i.e. a customer in our case) with a specific membership degree u_{ik} to cluster i. The membership degrees u_{ik} for $i = 1, \cdots, C$ are con-

strained to sum to 1. The FCM analysis tries to minimize the following objective function $Q[ii]$ where d_{ik} denotes the distance between case k and cluster center i and $[ii]$ refers to the company where the local cluster analysis is performed.

$$Q[ii] = \sum_{k=1}^{N} \sum_{i=1}^{C} u_{ik}^2[ii] d_{ik}^2[ii] \tag{1}$$

The local analysis provides each company with an initial set of cluster centers and a $N \times C$ partition matrix containing the membership degrees of each case k to each cluster i. Next, each company will exchange its partition matrix with the other companies. Because companies only exchange membership degrees, no private information about the customers is exchanged and consequently no privacy, security or business constraints are violated. This is indeed one of the key features of collaborative clustering: the communication between the data sites is realized at the level of granular information, i.e. partition matrices in the horizontal mode and cluster prototypes in the vertical one. Once the companies have received the partition matrices, the true collaborative FCM can be applied, which minimizes an augmented objective function (cf. eq. 2). This function integrates the information from the other companies, but uses collaboration links $\alpha[ii, jj]$ to control the extent of collaboration between each company $[ii]$ and $[jj]$. The set of all collaboration links is called the collaboration matrix.

$$Q^*[ii] = Q[ii] + \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \alpha[ii, jj] \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik}[ii] - u_{ik}[jj])^2 d_{ik}[ii] \tag{2}$$

With this new objective function, each company will get a new set of cluster centers and a new partition matrix. Once again, the new partition matrices will be exchanged and each company will minimize the augmented objective function again. These steps are repeated until some termination criterion is reached, which relies on the changes to the partition matrices obtained in successive iterations of the clustering method. Algorithm 2 displays the breakdown of the horizontal collaborative clustering scheme.

Algorithm 1 The horizontal collaborative clustering scheme

```
for each data location [ii] do
    Perform standard FCM clustering, minimizing objective function Q[ii]
end for
repeat
    Exchange the current partition matrices between the data locations
    for each data location [ii] do
        Run the collaborative FCM clustering, minimizing Q*[ii]
    end for
until some termination criterion is reached
```

Besides offering a distributed clustering algorithm which tackles privacy issues, the horizontal collaborative clustering algorithm can also be made resource-aware, which is often an issue in KDubiq research. This can be realized through the collaboration matrix, which determines the level of collaboration between two companies. If we take a look at the augmented objective function $Q^*[ii]$, we can see that it adds penalties when membership degrees differ across companies. These penalties realize the collaboration effect, but also increase the complexity of the clustering algorithm which demands more computing resources. However, if the collaboration link between two companies $[ii]$ and $[jj]$ is set to zero, no penalties are added for discrepancies in membership degrees between these two companies, which reduces complexity and computing demands. In the most extreme case, an entire row of the collaboration matrix could be set to zero if company $[ii]$ has very low computing resources. This would imply that company $[ii]$ doesn't use the information from the other companies at all and performs a classic local FCM cluster analysis. Of course, this would also eliminate the collaboration effect for this company. A possible approach to make the technique resource-aware could be to set the lowest collaboration links to zero if the algorithm demands too much from the computing location. Another option could be to exchange as less information as possible, i.e. not to pass the whole partition matrix but a more limited subset of information. This would eventually lead to another formulation of the collaborative clustering scheme which is beyond the scope of this article.

3 Optimizing the Collaboration Matrix

Determining the collaboration links can be done based on expert knowledge. Since the collaboration links control the effect of collaboration between two companies (data locations), a company expert could choose which companies they want to cooperate with and to what extent. However, this can still be a difficult task which could also lead to unbalanced results, i.e. there is no guarantee that collaboration will yield a meaningful result no matter how strong the connection between two companies might be.

In their work [2], Falcón et al. provided a way to learn the optimal collaboration matrix during the clustering analysis by applying the evolutionary optimization technique of Particle Swarm Optimization (PSO). We will use their approach while modifying the fitness function of the PSO for learning the optimal collaboration matrix. In their approach, the objective was to achieve maximum collaboration measured in terms of the partition matrices stemming from the collaboration, while our approach will focus on finding similar cluster compositions across companies. We prefer the latter approach because we want to mimic the situation where all companies would gather the entire information provided by the multiple feature spaces under discussion and perform a single global cluster analysis, which is however impossible due to aforementioned reasons.

PSO is an evolutionary optimization technique developed by Kennedy and Eberhart [8], inspired by the swarming behaviour of bird flocks and fish schools.

The optimization algorithm first initializes Z particles x_z, each particle representing a possible solution to the optimization problem. Next, the particles start to fly through the solution space and at each time interval t, the fitness of the solution is evaluated by means of a fitness function. During their flight, each particle remembers its own best position p_z. The direction of a particle in the solution space is influenced by the particle's current location $x_z(t)$, the particle's current velocity $v_z(t)$, the particle's own best position p_z and the global best position among all particles p_g. The particle's new position $x_z(t+1)$ is calculated by eq. 3 and eq. 4

$$v_z(t+1) = wv_z(t) + c_1r_1(p_z - x_z(t)) + c_2r_2(p_g - x_z(t)) \tag{3}$$

$$x_z(t+1) = x_z(t) + v_z(t+1) \tag{4}$$

where w is the inertia weight and c_1, c_2 are the acceleration constants drawing the particle toward the local and global best locations, respectively. The stochastic component of the PSO meta-heuristic is given by r_1 and r_2, which stand for two uniformly distributed random numbers. All particles keep moving in the solution space until some criterion is met. The global best position at the end is the solution to the optimization problem. For a broader insight about this widespread optimization technique, refer to [9].

In our particular case, a single particle will represent an entire collaboration matrix and the flight of the particles represents the search for a collaboration matrix which optimizes the similarity of the cluster compositions across data locations. To achieve such optimization, we formulate an appropriate fitness function which represents the dissimilarity in cluster composition across data locations. The goal of the PSO algorithm will be to minimize this function.

We redefine a cluster $C_i[ii]$ as a set of membership degrees $\{u_{1i}[ii], \cdots, u_{Ni}[ii]\}$. Now we can express the dissimilarity between cluster i from data site $[ii]$ and cluster j from data site $[jj]$ as follows:

$$d(C_i[ii], C_j[jj]) = \frac{1}{N}\sum_{k=1}^{N} |u_{ik}[ii] - u_{jk}[jj]|. \tag{5}$$

This dissimilarity measure will become zero, which is the lower bound, when all patterns belong to both clusters with the same degree. On the other hand, it will become 1, which is the upper bound, when both clusters are crisp and don't have any pattern in common. Furthermore, this measure is also symmetric. Next, to measure the dissimilarity between the entire cluster solution of data site $[ii]$ and data site $[jj]$, we compare each cluster of data site $[ii]$ with each cluster of data site $[jj]$ and only consider the smallest dissimilarity for each cluster (cf. eq. 6). Note that this measure equals to 0 when both cluster solutions are identical.

$$D[ii, jj] = \frac{1}{c}\sum_{i=1}^{c} \mathop{\mathbf{Min}}_{j=1}^{c} [d(C_i[ii], C_j[jj])] \tag{6}$$

The final fitness measure, which we will term as ρ, can be envisioned as the mean dissimilarity of the cluster solutions across all data sites.

$$\rho = \frac{2}{P(P-1)} \sum_{ii=1}^{P} \sum_{jj>i}^{P} D[ii, jj] \tag{7}$$

Given this fitness measure, we can use PSO to determine the optimal set of collaboration links. In our setting of KDubiq clustering, this implies that aside from the data locations, which we will call data nodes, we will need a computing location which performs the PSO algorithm. This location will act as the coordination node. It should be noted that the coordination node can be the same physical location as a specific data node, but this isn't necessary. Algorithm 2 shows how the collaborative clustering scheme and the particle swarm optimization can be integrated to automate the determination of the collaboration links.

Algorithm 2 The horizontal collaborative clustering scheme

```
Initialize Z particles x_z
repeat
    for each particle x_z do
        Perform alg. 1 with the collaboration matrix represented by x_z (data nodes)
        Send the partition matrices to the coordination node
        Calculate the fitness function ρ (coordination node)
        Calculate the new position x_z(t+1) (coordination node)
        Update p_z (coordination node)
    end for
    Update p_g (coordination node)
until some termination criterion is reached (coordination node)
Send the optimal collaboration links to the data nodes
Perform alg. 1 with the optimal collaboration matrix (data nodes)
```

4 Empirical Results

4.1 Data Description

The data used for our experiment comes from a customer satisfaction survey performed in the family entertainment sector. Customers were asked to rate the performance of several attributes of 4 different products from the same company on scales from 1 [Low] to 10 [High]. The customers also had to indicate how satisfied they were with each product as a whole on a scale from 1 [Low] to 10 [High]. In total, 666 respondents who bought all 4 products completed the survey entirely and were retained for our experiment. Table 1 shows the number of attributes for each product. Although all products were sold by the same

company, the data could also reflect 4 companies selling a single product to the same customer population. In the remainder of this article, we will assume the latter situation.

Table 1. Attribute Dimensions.

Attribute dimension	Number of attributes
Product A	7
Product B	4
Product C	6
Product D	3

4.2 Experiment and Discussion

If no privacy or security issues exist and all four companies are willing to exchange private customer information, they could collect all their customer data and use this to segment the customer population into different groups. We shall call this approach the *global clustering approach* (GC). However, companies often don't want to share private customer information or privacy constraints forbid to do so. Therefore, the common situation is that companies only use their own limited data to perform a customer segmentation. We shall call this situation the *local clustering approach with no collaboration* (LC). In general, we assume that the global clustering approach provides better results since the clustering algorithm has access to more information about the customers. The KDubiq approach discussed in this article tries to overcome the limitations of the LC approach, without actually exchanging private customer information. We shall call this approach the *local clustering approach, with collaboration* (LCC).

The purpose of the experiment is to analyze the differences between the clusters found by all three approaches. Given the context of customer satisfaction and the fact that all attributes measure performance or satisfaction from a "low-to-high" scale, we considered a 2-cluster model. For the GC approach we collected all data from the four companies and performed a standard FCM cluster analysis. For the LC approach, we performed four different standard FCM cluster analyses, one per company, using only the data available to that company. For the LCC approach, we performed a PSO-driven horizontal collaborative clustering approach with the following parameters: 50 particles, 200 iterations, $c_1 = c_2 = 2.0$, inertia weight dynamically varied from 1.4 to 0.4. Once the PSO-driven horizontal collaborative clustering has found the optimal collaboration matrix, it performs a separate clustering analysis per company and collaboration between the companies is realized by exchanging only membership degrees in the form of partition matrices.

Figures 1, 2, 3 and 4 show the profiles for all the cluster centers found by the three approaches. A profile shows the value of a cluster center for each attribute and gives an idea about the distance between the cluster centers. All four profiles

show that each cluster solution contains two clusters which can be identified as a high satisfaction/performance group and a medium satisfaction/performance group of customers. If we focus on company A, we see that the cluster centers are more separated in the LC approach than in the GC approach. We can also see that the LCC approach provides cluster centers which approximate the GC approach solution much better. This pattern can be found for all four companies. This implies that if companies would share their private information, they would find more subtle customer cluster due to the additional customer information. These results also show that the LCC approach can approximate the GC solution well without exchanging private customer information.

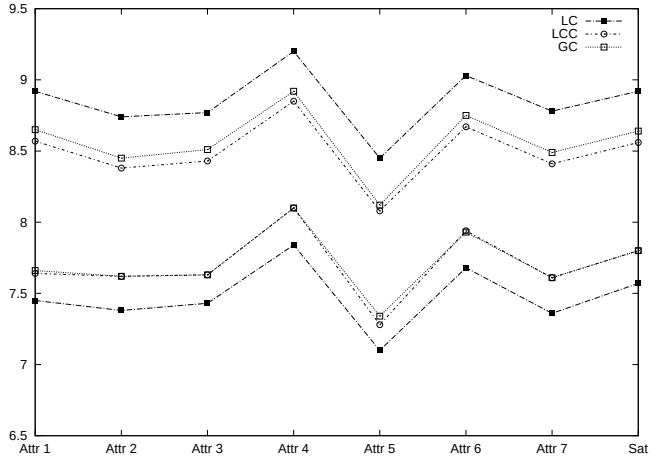

Fig. 1. Company A: Cluster Profiles (Average score per attribute for each cluster for each clustering approach)

Collecting all data as in the GC solution will not only affect the cluster centers, but also the assignment of customers to these clusters. In all three approaches, a fuzzy clustering technique was used and customers were assigned to the cluster for which they had a membership degree greater than 0.5. If a customer had a membership degree of exactly 0.5 for both clusters, the customer was not assigned and marked as undecided. Once again, we start from the idea that the GC approach must offer better results than the local approaches because it has access to more data and information about the customers. Therefore, we will study to which extent the cluster assignment of the LC and LCC approach approximates the cluster assignment of the GC approach. In total, the GC approach assigned 656 out of 666 customers (10 were left undecided). If company A uses the LC approach, 21.5% of the 656 customers were assigned to a different cluster compared with the GC approach. With the LCC approach, only 13.3% of the customers were assigned differently. The same pattern was confirmed for

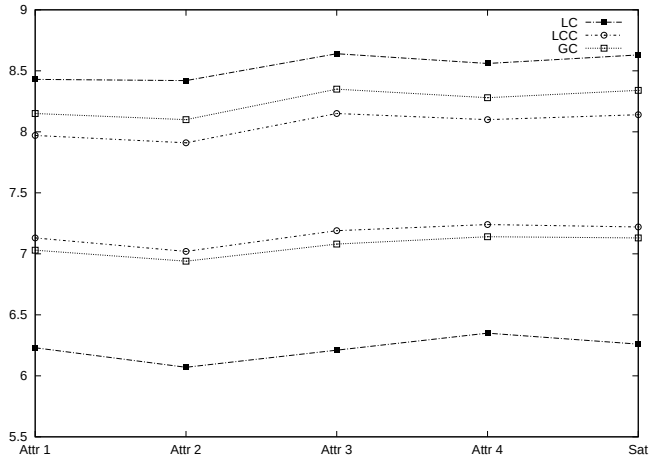

Fig. 2. Company B: Cluster Profiles (Average score per attribute for each cluster for each clustering approach)

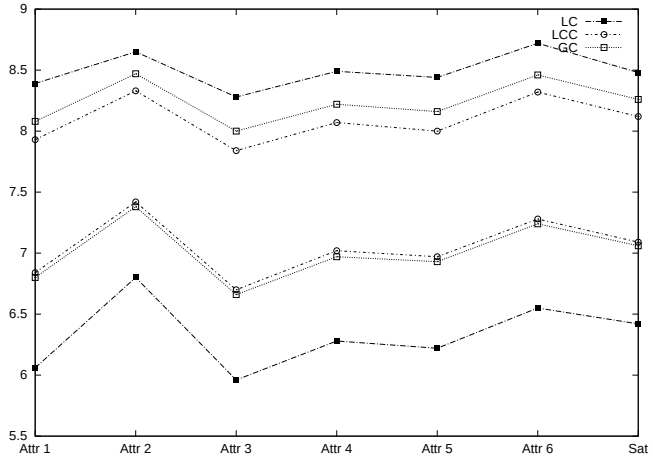

Fig. 3. Company C: Cluster Profiles (Average score per attribute for each cluster for each clustering approach)

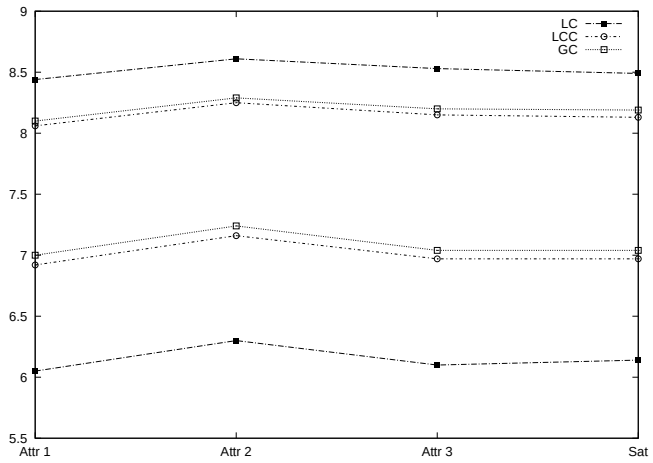

Fig. 4. Company D: Cluster Profiles (Average score per attribute for each cluster for each clustering approach)

the other companies. One can also see that the percentage customers assigned differently for the LCC approach compared with the GC approach, is the same for all four companies. This is caused by the fitness function used during the PSO approach, which tries to come up with the same cluster composition for each data site. These numbers indicate that this goal is well achieved. These results also indicate that companies could achieve very similar cluster composition compared with the results from the GC approach without exchanging private customer information by using the LCC approach.

Table 2. Percentage customers assigned differently compared to GC approach.

	A	**B**	**C**	**D**
LCNC	21.5%	22.3%	19.4%	23.5%
LCWC	13.3%	13.3%	13.3%	13.3%

5 Conclusions and Future Research

In this article, we have presented two existing clustering algorithms to the KDubiq community, i.e. horizontal collaborative clustering and PSO driven collaborative clustering. Both techniques address some typical issues in KDubiq research, such as privacy constraints and distributed computing. Our experiments illustrate that PSO driven collaborative clustering (LCC) benefits from exchanging information coded as partition matrices with other data sites. It resembles global

clustering (GC), which collects all information from all data sites prior to a non-distributed clustering approach, much better than traditional local clustering (LC), which only uses the information available at the specific data site. In conclusion, LCC produces the same meaningful results as GC without violating privacy or security restrictions.

Although the current results provide promising perspectives, future research about this clustering approach in KDubiq environments is needed. Firstly, it is recommended that the analyses are executed on data coming from true different companies, in contrast to our experiment where we simulate this type of environment. However, the current setup of the algorithm assumes that the data from each company relate to the same set of customers. In some situations, this assumption might be too restrictive and future research on how to overcome this limitation is needed. Secondly, now that the results in this paper have shown that this approach works, a benchmark against other distributed clustering approaches which preserve privacy is recommended.

Overall, the collaborative clustering algorithms are very suitable for applications in KDubiq environments, but future research remains necessary. The authors hope that this article can motivate and convince other researchers to explore the use of (PSO driven) collaborative clustering techniques in KDubiq environments.

References

1. Coordination Action for Ubiquitous Knowledge Discovery, `http://www.kdubiq.org/kdubiq/control/index`
2. Falcón, R., Jeon, G., Bello, R., Jeong, J.: Learning Collaboration Links in a Collaborative Fuzzy Clustering Environment In: Gelbukh, A., Kuri Morales, A.F. (eds.) MICAI 2007. LNCS, vol. 4827, pp. 483–495. Springer-Verlag, Berlin Heidelberg (2007)
3. Pedrycz, W.: Collaborative Fuzzy Clustering. Pattern Recognition Letters 23, 1675–1686 (2002)
4. Pedrycz, W., Rai, P.: Collaborative Fuzzy Clustering with the use of Fuzzy C-Means and its Quantification. Fuzzy Sets and Systems, DOI 10.1016/j.fss.2007.12.030 (2008)
5. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well-Separated Clusters. J. Cyber. 3, 32–57 (1973)
6. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
7. Hoppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis. John Wiley, Chichester (1999)
8. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization In: Proceedings of the 1995 IEEE International Conference on Neural Networks. vol. 4, pp. 1942–1948. IEEE Press, Piscataway, NJ (1995)
9. Bratton, D., Kennedy, J: Defining a Standard for Particle Swarm Optimization. In: Proc. of the IEEE Swarm Intelligence Symposium (SIS 2007), pp. 120–127. (2007)