PSO driven collaborative clustering: a clustering algorithm for ubiquitous environments

Non Peer-reviewed author version

DEPAIRE, Benoit; FALCÓN MARTINEZ, Rafael; VANHOOF, Koen & WETS, Geert

(2011) PSO driven collaborative clustering: a clustering algorithm for ubiquitous environments. In: Intelligent Data Analysis, 15(1). p. 49-68.

DOI: 10.3233/IDA-2010-0455
Handle: http://hdl.handle.net/1942/11628

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] School of Information Technology and Engineering (SITE),
University of Ottawa,
800 King Edward Ave, Ottawa, ON
Canada K1N 6N5
rfalcon@uottawa.ca

Abstract. The goal of this article is to introduce a collaborative clustering approach to the domain of ubiquitous knowledge discovery. This clustering approach is suitable in peer-to-peer networks where different data sites want to cluster their local data as if they consolidated their data sets, but which is prevented by privacy restrictions. Two variants exist, i.e. one for data sites with the same observations but different features and one for data sites with the same features but different observations. The technique contains two parts, i.e. a collaborative fuzzy clustering technique and a particle swarm optimization to optimize the collaboration between data sites. Empirical analysis show how and when this PSO-CFC approach outperforms local fuzzy clustering.

Key words: Ubiquitous Knowledge Discovery, Privacy Restrictions, Collaborative Clustering, Particle Swarm Optimization

1 Introduction

Computing environments and technologies are increasingly evolving towards mobile, finely distributed, interacting and dynamic environments containing massive amounts of heterogeneous, spatially and temporally distributed data sources. Examples are peer-to-peer systems, grid systems and wireless sensor networks. These ubiquitous computing environments pose new challenges to the field of knowledge discovery and data mining which remain unsolved by traditional data mining techniques. The research discipline fostering the inception of innovative data mining methodologies which are capable to handle these new challenges has been termed Ubiquitous knowledge discovery (KDubiq).

* The authors wish it to be known that, in their opinion, both first two authors should be regarded as joint First Authors.

KDubiq is a very wide research area with features setting it aside from traditional data mining and distributed data mining. KDubiq algorithms typically operate in environments with distributed computing power and distributed data sources. KDubiq algorithms must be capable of communication with different data sources and computing sites and should also be usable in environments too large to set up a master computer which collects and consolidates the different site results. Therefore, a KDubiq environment typically consists of several computing devices performing local data mining on site using limited information at hand while communicating with other sites. Sometimes computing and power resources are limited and require resource-aware KDubiq techniques. Sometimes, privacy and security restrictions hold which prevent raw data to be communicated or to be gathered centrally. Sometimes, KDubiq algorithms have to deal with data streams and must be able to process the data in real-time. Not all of these features have to be present at the same time for an environment to be ubiquitous and neither does a KDubiq algorithm necessarily have to deal with all these challenges. It all depends on the application. A sensor network will e.g. need algorithms which are very much resource-aware and limit communication to the absolute minimum while grid systems have much less resource constraints. On the other hand, KDubiq algorithms working on a peer-to-peer network are more likely to have to deal with privacy issues than algorithms working in a sensor network. For a more elaborated discussion of the different KDubiq characteristics, the interested reader may refer to the KDubiq Blueprint [1] which will be available as a Springer book by Fall 2008. Next, two motivating examples are given to illustrate the type of applications our KDubiq algorithm can deal with and to identify the KDubiq challenges faced in such applications.

Motivating Example 1. A company holds information on a set of potential customers which it wishes to segment. This will allow them to identify new opportunities and act appropriately. At the same time, other companies hold other information on the same set of potential customers and have a similar need to identify different segments. Due to privacy, security or business reasons, these companies are unwilling or prohibited to exchange their data. This prevents them from consolidating their data and performing analysis on the enriched data set. 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 peer-to-peer ubiquitous environment where data and computing power are distributed. They might benefit from a KDubiq clustering algorithm which allows them to segment the customers by using local data and findings coming from other companies without violating privacy, security or business constraints.

Motivating Example 2. Two companies retain the same type of information about their customers. Both companies have a different customer base and are not necessarily active in the same market. They want to segment their customers

to identify customer stereotypes and they expect that the stereotypes of the first company shows some overlap with the stereotypes of the second company. To increase the validation of specific customer stereotypes found by cluster analysis, both companies would prefer to enrich their own data with the data from the other company. However, privacy, security and business restrictions prevent them from exchanging their data. Yet again, 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). The companies can be considered as members of a peer-to-peer ubiquitous environment where data and computing power are distributed and where both could benefit from an adequate KDubiq clustering algorithm.

Both examples illustrate a ubiquitous environment with distributed data sources and distributed computing power where privacy issues prevent the exchange of raw data. This environment requires a clustering technique which can run locally, but which can find similar results as if all data was consolidated at one site, without violating the privacy restrictions. Other KDubiq features, such as resource limitations and real-time data mining are of much less importance in these types of applications and it should be noted that the KDubiq technique presented is not designed to consider these challenges.

1.1 Contributions and outline.

This article, introduces a KDubiq clustering algorithm which is a combination of particle swarm optimization (PSO) and collaborative fuzzy clustering (CFC). This algorithm, which is a modified version of the one introduced by Falcón et al. [2], matches the type of problem sketched in the motivating examples.

The next section gives an overview of relevant literature on related existing data mining techniques. The third section is devoted to discuss the original CFC algorithm, introduced by Pedrycz [3, 4] while the fourth section introduces PSO and shows how both techniques cooperate in this KDubiq environment. The fifth section, contains the results of elaborated experiments which test the added value of our KDubiq clustering technique compared with local fuzzy clustering (i.e. without collaboration between data sites). Finally, the limitations of the current technique and the directions for future research will be discussed before the final conclusions are drawn.

2 Relevant Work

Our approach can be considered as a distributed clustering approach which is not new to the literature. Several methodologies aimed at amalgamating information from multiple clustering analyses can be found, ranging from multi-cluster combiners [5] to consensus clustering [6] or the well-known cluster ensembles [7, 8]. The motivation behind these approaches stems from the need to combine the output of manifold unsupervised learning algorithms, following the footsteps of well-settled meta-classifier techniques, as a way of getting a coherent picture of

the underlying data dynamics, given the wide diversity of optimization criteria driving the overall clustering task in dissimilar scenarios.

Another reason for this fusion of knowledge structures is to assess the robustness of a specific clustering algorithm to the variability of sampled data, which is a pivotal issue in noisy and uncertainty-permeated environments. The representative approach in this category is consensus clustering [6], i.e. a rather simple resampling technique executing the same clustering method multiple times over a set of perturbed instances of the same original data set. Finally, it attains an agreement or consensus among the multiple runs. Among the different clustering aggregation protocols, cluster ensembles [7, 8] have proved to be an efficient and general framework for mapping a set of hard partitions into an optimal, combined partition, even when the number of clusters in each partition may vary. Assuming a unique, global collection of patterns (possibly distributed over several feature spaces), data confidentiality at the local level is enforced by exchanging only the crisp labels corresponding to the cluster assignments of each data object. Given the high computational complexity of the ensuing "median partition" problem as posed in [7] and [8], the need for heuristic methods to derive approximate solutions of an acceptable quality becomes relevant to the topic. This doesn't happen in collaborative fuzzy clustering, where good solutions to the clustering guided by the augmented objective function can be obtained with relatively modest computational effort. Moreover, the representation of cluster assignments as fuzzy memberships in our approach allows capturing the inherent relationships occurring at each individual repository at a deeper level.

A common denominator of any clustering aggregation technique, including bagging [9] and multi-cluster combination [5] is that the emphasis lies on getting a corporate representation of the individual findings supplied by multiple sources, often accomplished by unified consensus functions. Since the input labelings don't necessarily come from diverse, participating data sites, no collective pursuit for global knowledge directly affecting the distributed object collection takes place. This is precisely the purpose of the distributed clustering protocols, i.e. to gradually modify the data in each repository in such a way that meaningful results can be achieved and no implicit or explicit constraints over the data are violated. In that sense, privacy preservation plays a vital role as the cornerstone of lately released distributed clustering models in ubiquitous environments.

For example, the scheme presented in [10] is concerned with distributed object collections which are described by the same group of features. The proposed framework is general enough to embrace unsupervised and semi-supervised scenarios and supports a broad range of data types and learning algorithms. However, it suffers from several drawbacks compared to collaborative fuzzy clustering, among them the assumption about the existence of an underlying probabilistic distribution of the information dwelling at each local repository. Such distributions are to be learned (e.g. Gaussian with full-variance, von Mises-Fisher, etc.) and their condensed parameters are sent to a central location for further aggregation into a unified model. The existence of such a controller may be infeasible in ubiquitous scenarios like sensor networks deployed for emergency

monitoring or hazard surveillance. Furthermore, the quantification of the privacy requirements or the weights associated with the local models makes room for subjective criterion which leads to undesirable effects at the global level. None of these requisites become a hindrance for collaborative clustering.

A good analogy to CFC could be the fine-grained distributed version of the very popular K-Means algorithm with full privacy preservation put forward in [11]. It targets the horizontal mode (i.e. feature-scattered clustering) in such a way that an elaborate degree of isolation in the communication phase among the data sites has been envisioned. As part of the secure multiparty computation approach enforced to prevent exchange of valuable data, not even cluster prototypes or distances between patterns to clusters are disclosed, which turns the algorithm more complicated and awkward. In collaborative fuzzy clustering, either partition matrices or cluster prototypes stand as the main vehicles for conducting the whole optimization process. This type of granular information may offer some insight about the local information but not to the extent of compromising the identity of the actual patterns. Another problem with the algorithm in [11] is the way it computes the closest cluster for each data point, which is believed to be the sum of the partial distances found at the local level. While this is true for the most common distance measures (e.g. Euclidean, Manhattan, etc.), this feature severely restricts the shape of the clusters to be found and thus narrows the scheme's applicability to non-traditional scenarios.

3 Collaborative Fuzzy Clustering

Our approach is based on Collaborative Fuzzy Clustering (CFC) which was introduced in 2002 by Pedrycz [3] as a novel clustering algorithm intended to reveal the overall structure of distributed data which at the same time respects any restrictions preventing data sharing. As discussed in the previous section, this approach exhibits both similarities and differences with other existing techniques under the umbrella of distributed clustering (cf [4]).

The collaborative clustering scheme contains two major steps. First, a local clustering analysis is performed at each individual data site separately. Next, the local findings are exchanged and an augmented clustering algorithm is applied at each data site. This augmented clustering algorithm takes the local data as well as the results of other sites and into account. The second step is repeated until some termination criterium is met.

Typically, two types of collaborative clustering can be distinguished, i.e. the horizontal mode and the vertical mode. The horizontal mode assumes that each data site holds information on the same set of objects but described in different feature spaces, as is the case in the first motivating example. The vertical mode assumes that each data site holds information on different objects described in the same feature space. This is the case in the second motivating example. Next, the two versions of CFC will be explained in detail with the motivating examples in mind.

3.1 Horizontal CFC

There are P companies and each company $[ii]$ measured $A[ii]$ variables $x_{a[ii]}$ for the same set of N customers. The companies agreed to segment their customers in C clusters. The first step of the collaborative scheme performs a local fuzzy C-means cluster analysis (FCM) at each company $[ii]$ separately using only the local data. The generic version of the FCM method was proposed by Dunn [12] and Bezdek [13] in the 1980s, but has undergone significant changes over the years. The reader may refer to Hoppner et al. [14] 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 constrained to sum to 1. The FCM analysis tries to minimize the objective function

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

where d_{ik} denotes the distance between case k and cluster center i and where $[ii]$ refers to one of the companies at which the local analysis is performed.

The local analysis provides each company with a $C \times A[ii]$ cluster prototype matrix containing the cluster centers and a $N \times C$ partition matrix containing the membership degrees of each case k to each cluster i. Note that the size of the cluster prototypes differ across data sites because each site $[ii]$ has a different feature set size $A[ii]$.

In the second stage of CFC, the companies have to exchange their local results, without violating the privacy restrictions. In horizontal CFC the communication between data sites is realized by exchanging the partition matrices. The partition matrices are comparable between data sites in Horizontal CFC because they relate to the same customers and clusters. At the same time, no private information about the customers is exchanged and without the prototype matrices, which are not exchanged, it is impossible to retrieve the original data. By realizing the communication at the level of granular information, collaborative clustering succeeds in complying to any privacy, security or business constraints.

Once the companies received the partition matrices, the true collaborative FCM can be applied, which minimizes an augmented objective function (cf. eq. 3). This function integrates the information from the other companies with the local data and uses collaboration links $\alpha[ii, jj]$ to control the extent of collaboration between two companies $[ii]$ and $[jj]$. The set of all collaboration links is called the collaboration matrix.

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

The augmented objective function adds a second term to the goal function which quantifies the differences between the partition matrices of every data site. This forces the CFC algorithm to search for similar clustering results across data sites. During this collaborative stage, the entries of the partition matrix and prototype matrix will be recomputed. Next, the new partition matrices will be exchanged and each company will minimize the augmented objective function again. This is 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 1 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
```

3.2 Vertical CFC

In the vertical version of CFC, there are P companies and each company ii has a different set of $N[ii]$ customers which are all measured by the same set of A features $\{x_1, \ldots, x_a, \ldots, x_A\}$.The companies agreed to segment their customers in C clusters. The first step of the collaborative scheme performs a local fuzzy C-means cluster analysis (FCM) at each company $[ii]$ separately using only the local data. This step is exactly the same as for the horizontal CFC variant. The local analysis provides each company with a $C \times A$ cluster prototype matrix containing the cluster centers and a $N[ii] \times C$ partition matrix containing the membership degrees of each case k to each cluster i. Note that this time the size of the partition matrices differ across data sites because each site $[ii]$ has a different set of $N[ii]$ customers.

In the vertical CFC, the prototype matrices are exchanged instead of the partition matrices. Since the feature sets are equal across data sites, the cluster prototypes can be compared among data sites which allows us to measure the difference between the local cluster solutions. Only prototype matrices are exchanged which preserves the privacy of the data.

Once the companies receive the prototype matrices, the true collaborative FCM can be applied, which minimizes an augmented objective function (cf. eq. 3). As in the horizontal version, a new term is added to the goal function

which quantifies the inequality between the cluster solutions at different data sites. The second term of the augmented function compares the local membership degrees $u_{ik}[ii]$ with the membership degree $u_{ik}[jj]$. This is the membership a case k would have if cluster center i was positioned at the location of cluster center i from data site jj. The vertical version also uses collaboration links $\alpha[ii, jj]$ to control the extent of collaboration between two companies $[ii]$ and $[jj]$.

$$Q^*[ii] = Q[ii] + \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \alpha[ii, jj] \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik}[ii] - u_{ik}[jj])^2 d_{ik}[ii] \tag{3}$$

During the collaborative stage, the entries of the partition matrix and prototype matrix will be recomputed and the new prototype matrices will be exchanged. Next, each company will minimize their augmented objective function again and this is repeated until some termination criterion is reached. Algorithm 2 displays the breakdown of the vertical collaborative clustering scheme.

Algorithm 2 The vertical collaborative clustering scheme

```
for each data location [ii] do
    Perform standard FCM clustering, minimizing objective function Q[ii]
end for
repeat
    Exchange the current prototype 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
```

4 Optimizing the Collaboration Matrix

One of the parameters of CFC is the collaboration matrix which can be set by using expert knowledge. Company experts should set high collaboration links with companies they want to cooperate strongly with and low collaboration links with companies whose data is not believed to be very compatible. Choosing the right collaboration links can be a difficult task which could lead to unbalanced results if chosen incorrectly. There is no guarantee that collaboration will yield a meaningful result no matter how strong the connection between two companies might be.

Falcón et al. [2] developed a technique to optimize the collaboration matrix during the clustering analysis by applying the evolutionary optimization technique of Particle Swarm Optimization (PSO). Their objective was to a maximize the level of collaboration which is different from our objective. Our goal is to mimic the situation where all companies would consolidate their data sets

and perform a single global cluster analysis, which is impossible due to privacy constraints. Therefore, our approach will need a PSO objective function which focusses on finding similar cluster solutions across companies. This section presents a modification of the PSO objective function to suit the needs of our KDubiq environment.

It should be noted that the KDubiq approach only makes sense if companies consider their own data as incomplete and want to improve the quality of their local analysis by collaborating with other companies which they believe have compatible data. If the data from other companies are not compatible or relevant or companies don't want to dilute the knowledge structures found by local analysis in favor of the knowledge structures revealed through a global clustering approach, companies should stick with their local analysis.

Particle Swarm Optimization (PSO) is an evolutionary optimization technique developed by Kennedy and Eberhart [15], inspired by the swarming behavior of bird flocks and fish schools. The optimization algorithm initializes Z particles x_z, each representing a possible solution to the optimization problem. Next, the particles start to fly throughout 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. 4 and eq. 5:

$$v_z(t+1) = wv_z(t) + c_1r_1(p_z - x_z(t)) + c_2r_2(p_g - x_z(t)), \tag{4}$$

$$x_z(t+1) = x_z(t) + v_z(t+1), \tag{5}$$

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 components of the PSO meta-heuristic are given by r_1 and r_2, which represent 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 [16].

In the PSO-CFC approach, 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 solutions across data locations. To achieve such optimization, an appropriate fitness function is developed which represents the average dissimilarity between cluster solutions across data sites. The goal of the PSO algorithm is to minimize this function.

The objective function can be defined in three steps. The first step, which is different for the horizontal and vertical variant, measures the dissimilarity between cluster i from data site $[ii]$ and cluster j from data site $[jj]$. In the horizontal variant, cluster dissimilarity must be based on the partition matrices which is the only information available about the clusters from other data sites.

In horizontal PSO-CFC, a cluster $C_i[ii]$ is redefined as a set of membership degrees $\{u_{1i}[ii], \cdots, u_{Ni}[ii]\}$ and the dissimilarity between cluster i from data site $[ii]$ and cluster j from data site $[jj]$ are measured as follows:

$$d(C_i[ii], C_j[jj]) = \frac{1}{N} \sum_{k=1}^{N} |u_{ik}[ii] - u_{jk}[jj]|. \tag{6}$$

This dissimilarity measure becomes zero, which is the lower bound, when all patterns belong to both clusters with equal membership 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.

In the vertical variant, dissimilarity between two clusters must be based on the prototype matrices which is the only information known from the cluster solutions at other data sites. In vertical PSO-CFC, the dissimilarity between cluster i from data site $[ii]$ and cluster j from data site $[jj]$ is measured by calculating the Euclidean distance between the two cluster centers $v_i[ii]$ and $v_j[jj]$ (cf Equation 7 where $v_{ia}[ii]$ represents the a^{th} feature of cluster center $v_i[ii]$).

$$d(C_i[ii], C_j[jj]) = \sqrt{\sum_{a=1}^{A} (v_{ia}[ii] - v_{ja}[jj])^2} \tag{7}$$

This distance measure only has a lower bound equal to 0 which is reached when both cluster centers are at the exact same location in the feature space. No upper bound exists.

The second step is to measure the average dissimilarity between the clusters of data site $[ii]$ and data site $[jj]$. This requires a proper mapping between the clusters of data site $[ii]$ and the clusters of data site $[jj]$. Instead of developing an algorithm to perform this non-trivial mapping, our approach uses a simple heuristic which appears to work very well. This heuristic maps a cluster i from data site ii to the least dissimilar cluster j from data site jj. Dissimilarity is measured with Eq. 6 for horizontal PSO-CFC and Eq. 7 for vertical PSO-CFC. The average dissimilarity between two data site $[ii]$ and $[jj]$ is calculated by means of Eq. 8. Note that this measure equals 0 when both cluster solutions are identical.

$$D[ii, jj] = \frac{1}{c} \sum_{i=1}^{c} \underset{j=1}{\overset{c}{\mathbf{Min}}} \left[d(C_i[ii], C_j[jj]) \right] \tag{8}$$

The third and final step of constructing the objective function represents the level of dissimilarity present between all data sites. The PSO objective function, which will be termed ρ, measures the average dissimilarity of all possible combinations of two data sites $[ii]$ and $[jj]$. With P data sites, there are $\frac{P(P-1)}{2}$ data site combinations, which results in the following objective function:

$$\rho = \frac{2}{P(P-1)} \sum_{ii=1}^{P} \sum_{jj>i}^{P} D[ii, jj] \tag{9}$$

The PSO-CFC algorithm uses this objective function to determine the optimal set of collaboration links. In the KDubiq clustering setting, this implies that aside from data locations, which are called data nodes, a computing location is need 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 particular data node, but doesn't have to be. Algorithm 3 shows how the collaborative clustering scheme and the particle swarm optimization are integrated to automate the determination of the collaboration links.

Algorithm 3 The horizontal collaborative clustering scheme

Initialize Z particles x_z *(coordination node)*
repeat
  for each particle x_z **do**
    Perform Alg. 1 or 2 with collaboration matrix x_z *(data nodes)*
    Send the partition matrices to the coordination node
    Calculate the fitness function ρ *(coordination node)*
    Update p_z *(coordination node)*
  end for
  Update p_g *(coordination node)*
  for each particle x_z **do**
    Calculate the new position $x_z(t+1)$ *(coordination node)*
    Send $x_z(t+1)$ to the data nodes
  end for
until some termination criterion is reached *(coordination node)*
Send the optimal collaboration links to the data nodes
Perform Alg. 1 or 2 with the optimal collaboration matrix *(data nodes)*

5 Experiments

5.1 Methodology

Each analysis compares three different clustering approaches, i.e. the global clustering (GC) approach, the local clustering (LC) approach and the collaborative clustering (CC) approach, and requires a separate data set per data site for the CC and LC approach and a consolidated data set for the GC approach. The LC approach performs a standard FCM clustering for each data set separately. This represents the situation where companies only have access to their own data and are not willing to collaborate with other companies. The GC

approach represents the other extreme where no privacy constraints hold and where companies consolidate their data sets to achieve a single data set of higher quality, i.e. larger feature space or more observations. This approach is tested by performing a FCM clustering analysis on the consolidated data set. The CC approach represents the KDubiq environment described in the motivating examples where privacy constraints prevent companies from sharing their data. However, in contrast with the LC approach, where companies work as isolated sites, the CC approach uses a collaboration mechanism to approximate the results of the GC approach. The CC approach is tested by performing a PSO-CFC clustering analysis across all data sites with the following parameters: 20 particles, 100 iterations, $c_1 = c_2 = 2.0$ and the inertia weight dynamically varied from 1.4 to 0.4. The purpose of each analysis is to evaluate the quality of the clustering results from all three approaches. In this article, cluster quality is defined as the number of correctly assigned cases. Analysis were performed on both artificial data sets and real-life data sets.

Artificial data sets have the benefit that we have full control over the structure of the data, which allows us to isolate the effect of a specific data structure property on cluster quality. It also has the benefit that the true cluster memberships are known which allows an exact evaluation of the cluster quality. Once the artificial data structure is designed, a sample can be drawn for each data site and the consolidated data set is created by joining all local data sets. Next, the three approaches are applied to the data sets and the number of incorrectly assigned clusters are counted for each approach. Cluster assignment is based on the cluster with the highest cluster membership.

Assume that we want to compare the cluster quality of the GC approach versus the CC approach. This comparison is done by subtracting ϵ_{cc}, i.e. the number of errors of the CC approach, from ϵ_{gc}, i.e. the number of errors of the GC approach, which results in the new variable $d = \epsilon_{cc} - \epsilon_{gc}$. Note that one experiment leads to a single observation of d which can not lead to reliable conclusions. Therefore, for each analysis, we performed 30 experiments. Each experiment z draws new data from the artificial data distributions and d_z is calculated. This results in a sample of cluster quality comparisons $\{d_1, \ldots, d_z, \ldots, d_{30}\}$. Ultimately, we want to draw inferences about the population mean μ_d which tells us if CC performs better than GC on average or not in general. To draw conclusions about μ_d, we need an estimator of this population parameter, which is typically the sample average $\overline{d} = \frac{\sum_{z=1}^{30} d_z}{30}$. However, the true population mean μ_d might differ to some extent from its estimator $\overline{d}$. Therefore, confidence intervals need to be constructed to know how reliable $\overline{d}$ is as an estimator of μ_d. To build such confidence intervals, the sampling distribution of $\overline{d}$ around μ_d has to be known.

If the underlying variable d is normally distributed, the t-test could be used to evaluate the sample mean and to construct confidence intervals [17, 18]. However, the underlying distribution of d is unknown and the sample size is not very large in our analysis ($z = 30$) which makes the t-test more sensitive to violations of the normality assumption. Instead of relying on the robustness of the t-test, we opted for the nonparametric bootstrap technique to construct confidence intervals.

Nonparametric bootstrap [19] is a recently fashionable way for statistical inference for quantities for which theoretical and/or even asymptotic results are hard to derive. The basic idea behind bootstrapping is that new samples S_b^* are created by resampling with replacement from the original sample $S = \{d_1, \ldots, d_z, \ldots, d_{30}\}$ until the new sample size is equal to the original sample size. This process is repeated a number of times which results in a set of B resamples, denoted as $\{S_1^*, \ldots, S_b^*, \ldots, S_B^*\}$. The key idea is that all these resamples can be considered as samples from the unknown population (or at least they look like the unknown population).

If the sample average $\overline{d}$ based on sample S_b^* is denoted as $\overline{d_b^*}$, then the distribution of $\overline{d_b^*}$ around $\overline{d}$ is analogous to the sampling distribution of $\overline{d}$ around the population mean μ_d [20]. Since we can make B, i.e. the number of resamples, very large, we can get a very detailed empirical distribution of $\overline{d_b^*}$ which provides a detailed estimate of the sampling distribution of $\overline{d}$. This estimate of the sampling distribution can be used to construct confidence intervals around $\overline{d}$. Various approaches to construct bootstrap confidence intervals exist, such as the *normal-theory interval*, the *bootstrap percentile interval* and the *bias-corrected, accelerated percentile intervals* (BC_a). According to Fox [20], the latter are preferable and for a 95% BC_a confidence interval, the number of bootstrap samples should be on the order of 1000 or more. For more technical details about the construction of bootstrap confidence intervals and the use of bootstrapping to evaluate the results of machine learning algorithms, the interested reader should refer to [20, 18].

In summary, the following methodology is used for each analysis:

a) An artificial data distribution is designed.
b) For each of 30 experiments, samples from the artificial data distribution are drawn for each data site and the consolidated data set is created.
c) For each experiment, the measurements of interest are constructed and the average over the thirty experiments is calculated.
d) Bootstrapping is used to generate BCa confidence intervals, based on 10000 bootstrap samples.

Artificial data sets give researchers full control over the structure of the data which allows them to assign changes in cluster performances to specific causes, similar to a controlled laboratory experiment. On the other hand, artificial data sets might not always reflect reality which makes the results and conclusions less usable. Therefore, we also conducted some analysis on a real-life marketing data set. A first problem is that the true cluster memberships of the customers are not known. In our experiments we used the cluster assignments of the GC approach as the estimates for the true cluster memberships. The quality of the CC and LC approach is measured by counting the number of customers they assign to different clusters than the cluster assigned by the GC approach.

A second problem is that we only have one data set and the underlying distribution is unknown which prevents us from drawing new samples. Therefore, the bootstrap principle was used and new data sets were generated by sampling

from the original data set with replacement until the size of the new data sets are equal to the original data set size. According to the bootstrap principle, these resamples can be considered as samples from the unknown population (or at least they look like the unknown population).

The subsequent steps of calculating the average quality difference between CC and LC and the construction of confidence intervals is completely analogous to the methodology followed with the artificial data sets.

5.2 Horizontal Clustering: Empirical Results

Artificial Data Sets. Firstly, three different analysis were performed on artificial data to evaluate the relative quality of the LC and CC approach in different situations. Each analysis uses different data structures, which represent increasing clustering task complexity. We first will discuss the data structure of each analysis, before the results are discussed.

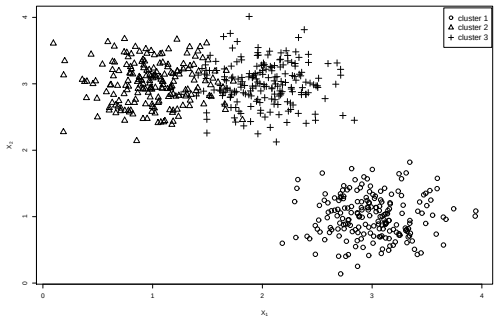
All three analysis concern two data sites. Each data site has a data set with 600 observations from three clusters, i.e. C_1, C_2 and C_3, 200 observations from each cluster. The difference between the data sites are the features used to describe the observations. The consolidated data set is the combination of the local data sets and has always four features, i.e. X_1, X_2, X_3 and X_4.

The first analysis represents the easiest clustering task. It uses cases drawn from three different multivariate normal distribution, i.e. one per cluster, with mean vector μ_1, μ_2 and μ_3 for respectively clusters C_1, C_2 and C_3 and with the same covariance matrix Σ for all three clusters.

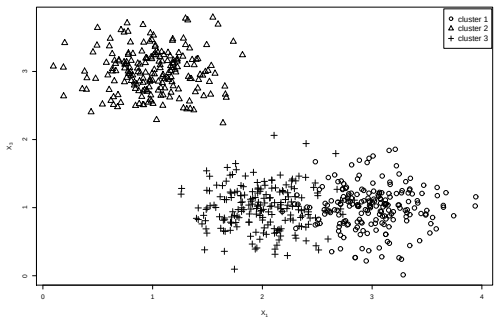
$$\mu_1 = \begin{bmatrix} 3 & 1 & 1 & 2 \end{bmatrix} \qquad \mu_2 = \begin{bmatrix} 1 & 3 & 3 & 2 \end{bmatrix} \qquad \mu_3 = \begin{bmatrix} 2 & 3 & 1 & 2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$

Figure 1 shows the 2 dimensional scatterplots for a random data set generated from the artificial data structure discussed above. These plots show that features X_2 and X_3 are sufficient to perfectly separate the three clusters while feature X_4 does not help the identification of the three clusters at all. In the first analysis, data site A has access to features X_1, X_2 and X_3 and data site B has features X_3 and X_4. Figure 1d reveals that data site A should have no problem assigning the cases to the correct cluster, while Figure 1f shows that it is impossible for site B to separate clusters C_3 and C_1. The first analysis simulated a KDubiq environment where one company has all necessary features to solve a perfectly separable clustering problem, while the second company lacks important features.

The situation simulated in the second analysis differs from the first analysis because in the second analysis no single company has all necessary features to solve the clustering problem. However, the cluster problem can still be perfectly separated with all the features of the consolidated data set. The data sets generated for the second analysis still use the same distribution as above, but this



(a) X_1 vs X_2



(b) X_1 vs X_3

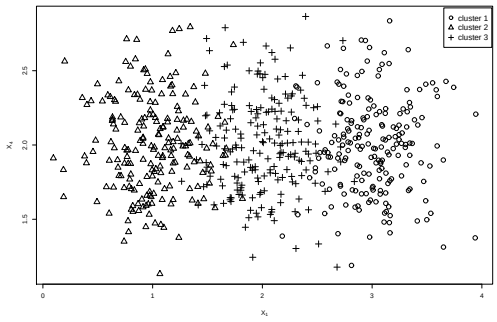
Fig. 1: Scatterplot of full artificial data set [*To be continued ...*]

time data site A has only access to features X_1 and X_2, while data site B has access to features X_3 and X_4. Figure 1a shows that data site A can no longer separate the three clusters as there is some overlap between clusters C_2 and C_3, while Figure 1f shows that data site B still has problems separating clusters C_3 and C_1.

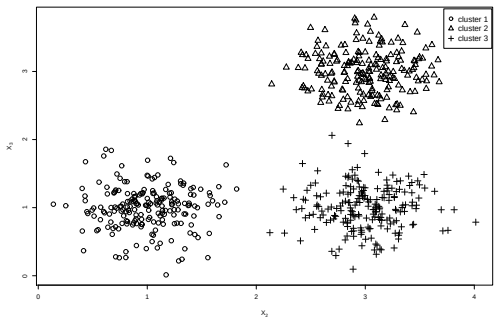
The third analysis represents the same situation as in analysis 2, but now for a problem which is not perfectly separable, even if all features are provided. This is a more realistic situation. The data generated for the third analysis still uses three multivariate normal distributions with the same covariance matrix as above, but with the following mean vectors for respectively clusters C_1, C_2 and C_3:

$$\mu_1 = \begin{bmatrix} 3 & 1.5 & 1 & 2 \end{bmatrix} \qquad \mu_2 = \begin{bmatrix} 1 & 2 & 1.5 & 2 \end{bmatrix} \qquad \mu_3 = \begin{bmatrix} 2 & 2 & 1 & 2 \end{bmatrix}$$

Analogue to analysis 2, data site A received features X_1 and X_2, while data site B has access to features X_3 and X_4.



(c) X_1 vs X_4

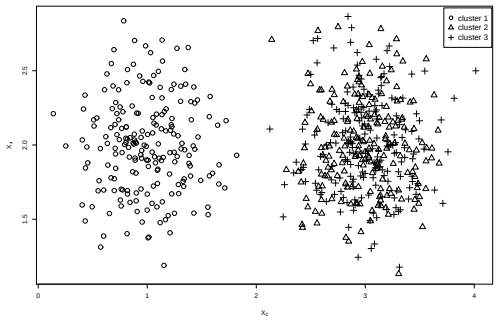


(d) X_2 vs X_3

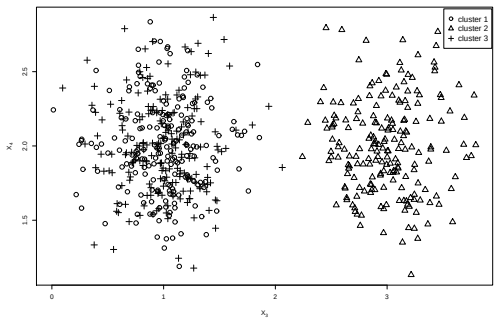
Fig. 1: Scatterplot of full artificial data set [*To be continued ...*]

For all three analysis, the methodology described in subsection 5.1 was used and the results are presented in Table 1. The ϵ statistics refer to the number of incorrectly assigned cases in a certain data site by a specific approach. For example, ϵ_{lc}^{AB} are the number of incorrectly assigned cases for data site A and B by the LC approach. The ϵ statistics for the GC approach are always calculated for the consolidated data set and the superscript denoting the data sites is left out of the notation.

Analyzing the results of the first analysis, the LC approach succeeded in assigning the cases of the first data site to the correct clusters, while it had great troubles assigning the cases of the second data site. These results were expected given the data structure of both sites. The CC approach did succeed to overcome the clustering problem of the second data site. The information exchanged during the collaboration phase of the CC approach successfully lowered the number of incorrectly assigned cases at data site B to 14 cases out of 600, compared with 202 cases out of 600 for the LC approach. On the other hand,



(e) X_2 vs X_4



(f) X_3 vs X_4

Fig. 1: Scatterplot of full artificial data set

it is remarkable that the CC approach did make assignment errors for data site A, which should be perfectly separable. Further analysis of the results revealed that the PSO-CFC algorithm got stuck in a local optimum during three of the thirty experiments. Initially, the PSO-CFC algorithm had even more problems getting stuck in local optima. By changing the default number of particles to 40, we managed to lower the number of local optima problems, but couldn't eliminated them completely. These adjusted settings were used for all horizontal PSO-CFC experiments. Comparing the CC and LC approach, the results are very promising. When one data site contains all information to segment the customers correctly, other data sites clearly benefit by applying the CC approach.

The second analysis changed the environment such that no single data site has all information available to solve a separable cluster problem. This is reflected in the results. The LC approach didn't make assignment errors for data site A in the previous analysis, but now it makes 20 assignment errors on average. Note that these errors are caused by the fact that data site A is no longer perfectly

Table 1: Cluster quality comparisons between the GC, LC and CC approach in a horizontal clustering environment on artificial data

		Analysis		
Statistic		1	2	3
ϵ_{LC}^{A}	mean	0.2	21.7	84.6
	95% CI	[0.1,0.4]	[20.4,22.9]	[79.9,90.7]
ϵ_{LC}^{B}	mean	202.1	199.6	307.7
	95% CI	[200.9,203.3]	[197.8,201.3]	[302.7,314.8]
ϵ_{CC}^{A}	mean	14.27	26.4	108.4
	95% CI	[3.6,38.83]	[22.0,33.8]	[94.9,129.0]
ϵ_{CC}^{B}	mean	14.3	26.4	108.4
	95% CI	[3.6,38.9]	[22.0,33.8]	[94.9,129.0]
$\epsilon_{LC}^{AB} - \epsilon_{CC}^{AB}$	mean	201.8	221.0	220.1
	95% CI	[200.5,203.0]	[218.5,223.5]	[213.3,227.9]
$\epsilon_{CC}^{AB} - 2\epsilon_{GC}$	mean	28.0	52.4	44.6
	95% CI	[6.7,77.0]	[43.6,67.1]	[17.87,85.87]
$\epsilon_{LC}^{AB} - 2\epsilon_{GC}$	mean	173.7	168.6	175.5
	95% CI	[124.0,195.0]	[152.1,177.8]	[132.8,203.1]

separable into three clusters. The number of errors made by the LC approach for data site B remains the same as in the first analysis, which was expected since it concerns the same data structure. Also in the second analysis, the CC approach succeeded to cluster the observations of data site B much better than the LC approach, i.e. 26.4 incorrect cluster assignments versus 199.6 incorrect cluster assignments. However, the number of errors made on data site B by the CC approach is higher than in the first analysis, although the data structure of data site B didn't change and the problem is still perfectly separable given all features. Furthermore, the number of errors made by the CC approach on data site A and B together is more or less equal to the difference in number of errors between the CC and the GC approach. This indicates that the GC approach still successfully separates the clusters in the second analysis. Therefore, one can conclude that when not all necessary features are present at a single data site, the CC approach will make more errors, but still significantly decreases the number of errors made compared to the LC approach, i.e. 221 errors less on average over both data sites.

The third analysis reflects the more realistic situation where observations are not perfectly separable given all features. Because of this, the GC approach now also makes assignment errors (86.1 on average, with [83.7,88.6] as 95% CI). Not only does GC makes more errors than in the previous two analysis,

also the LC and CC approach experience more difficulties clustering the data. However, comparing CC and LC, the results show that the CC approach makes on average 220.1 errors less on both data sites combined than the LC approach. Thus, even when data is no longer perfectly separable, the CC approach improves the clustering results compared to the LC approach.

Customer Satisfaction Data Set. Besides the artificial data set, we also experimented with a real data set to see how our approach behaves in a real world setting. This data set comes from a customer satisfaction survey performed in the family entertainment sector. Customers were asked to rate the performance of several attributes for 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 2 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 2: Attribute Dimensions.

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

If no privacy or security issues would exist and all four companies were willing to exchange private customer information, they could consolidate all their customer data and use this to segment the customer population into different groups. This would be the GC approach. 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, which is the LC approach. In general, we assume that the global clustering approach provides better results since the clustering algorithm has access to more information about the customers. In this article, we propose the CC approach, i.e. a third approach trying to approximate the GC results without violating privacy restrictions.

The purpose of this analysis 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 on a "low-to-high" scale, we considered a 2-cluster model. The cluster assignments of the

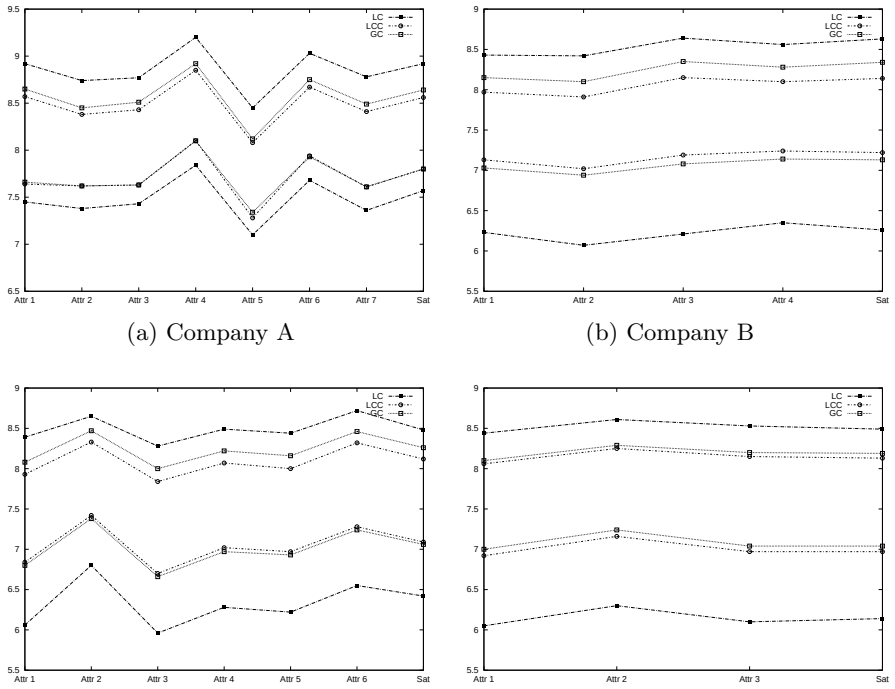
GC approach were used as an estimate of the true cluster assignments. The methodology described in section 5.1 was used to calculate the average number of incorrectly assigned cases for both the CC and the LC approach, together with their bootstrap confidence intervals. The results are shown in Table 3

Table 3: Cluster quality comparisons between the LC and CC approach in a horizontal clustering environment on real data

Statistic		LC	CC
$\epsilon_{\cdot}^{A}$	mean	223.9	185.5
	95% CI	[203.5,242.3]	[170.6, 199.0]
$\epsilon_{\cdot}^{B}$	mean	224.1	185.5
	95% CI	[208.2,238.2]	[170.6, 199.0]
$\epsilon_{\cdot}^{C}$	mean	214.8	185.5
	95% CI	[202.0,226.0]	[170.6, 199.0]
$\epsilon_{\cdot}^{D}$	mean	246.1	185.5
	95% CI	[233.2,257.9]	[170.6, 199.0]

These results look very promising. Under the assumption that the cluster assignments of the GC approach are good approximations of the true cluster assignments, we find that the CC approach outperforms the LC approach on all four data sites. Summed over all four data sites, the CC approach makes 166.7 errors less for 2664 cluster assignments, with a 95% confidence interval of $[152.8, 181.4]$. These results indicate that companies could achieve better cluster compositions by using the CC approach instead of the LC approach. Also note that no local optima problems were experienced for the real data set.

Not only does the collaboration aspect of the CC approach has an impact on the cluster assignments, it also influences the cluster centers. Figure 2 show the profiles for all the cluster centers found by the three approaches for each data site. These figures are based on the original data. A profile shows the values of a cluster center for each feature 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 CC 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 balanced customer clusters due to the additional customer information. These results confirm that the CC approach can approximate the GC solution without revealing private customer information.



(a) Company A (b) Company B

(c) Company C (d) Company D

Fig. 2: Cluster profiles

5.3 Vertical Clustering: Empirical Results

Artificial Data Sets. To assess the performance of the vertical variant of PSO-CFC, four different analysis were conducted on artificial data sets. Each analysis represents a different task, which ranges from easy to difficult. Each analysis assumes two data sites with their own local data sets containing 300 cases for the first three analysis and 450 cases for the last analysis. Both data sites have their data described by two features, X_1 and X_2. The consolidated data set, which is used for the GC approach, always consists of the combination of both local data sets. The data sets in the first analysis contain data from two clusters, while the other analysis use data containing three clusters.

The first analysis represents a very easy clustering problem for both data sites. Both sites draw their data from two multivariate normal distributions, i.e. one per cluster, with the following mean vectors μ_1, μ_2 for respectively clusters C_1, C_2 and with the same covariance matrix Σ for both clusters:

$$\mu_1 = \begin{bmatrix} 4 & 4 \end{bmatrix} \qquad \mu_2 = \begin{bmatrix} 0 & 0 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 0.7 & 0 \\ 0 & 0.7 \end{bmatrix}.$$

The data sets in analysis 1 represent a randomly drawn data sample, where both clusters are equally represented and which is perfectly separable.

The second analysis tries to make the clustering problem more challenging by adding a third cluster and by changing the data distribution of the clusters such that the clusters do overlap which makes a perfect cluster assignment no longer possible. This analysis uses the following Σ as a covariance matrix and μ_1, μ_2, μ_3 as mean vectors for respectively C_1, C_2 and C_3:

$$\mu_1 = \begin{bmatrix} 2 & 3 \end{bmatrix} \qquad \mu_2 = \begin{bmatrix} 2.5 & 2 \end{bmatrix} \qquad \mu_3 = \begin{bmatrix} 3 & 3 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}.$$

The second analysis represents a situation where the data is drawn completely at random and where both clusters are equally present in the data, but for a problem which is no longer perfectly separable.

The third analysis makes the situation more realistic and more complex by changing the distribution of the clusters within the data. The data are still sampled at random from the same data distributions as in analysis 2, but this time cluster C_3 is oversampled and C_2 is undersampled in data site A, while C_2 is oversampled and C_3 is undersampled in data site B. The exact sampling distributions of the clusters are shown in Table 4. This analysis represents a situation where data is not perfectly separable and where some clusters might be more present within the data than others. This is a very common situation in real-life data sets. For example, a company which mainly sells to one type of customer will most likely have a majority of this type of customer within its data set. However, it should be noted that conditional on the cluster, the data are still a random sample.

Table 4: Cluster distribution for third analysis

	C_1	C_2	C_3
Site A	100	50	150
Site B	100	150	50

The fourth analysis goes one step further by creating so called biased samples for both data sites. The data for data site A and B are still drawn from the same data distributions as in the previous two analysis, but no longer at random, except for the observations of cluster C_3. Cluster C_1 was split in two parts based on the test expression $X_2 - X_1 > 1$. Observations passing this test, belong to the first part of C_1, those failing the test belong to the second part. Cluster C_2 was split with the test expression $X_2 > 2$. Table 5 shows the exact sampling distribution for the fourth analysis. Note that the clusters are not equally represented locally, neither are they drawn at random.

The results of all four analysis are presented in Table 6. The results of this first analysis show that the starting problem was indeed very easy, since none of the approaches made significant cluster assignment errors. The second analysis,

Table 5: Cluster distribution for fourth analysis

	C_1		C_2		C_3
	$X_2 - X_1 > 1$	$X_2 - X_1 \leq 1$	$X_2 > 2$	$X_2 \leq 2$	
Site A	120	30	60	90	150
Site B	30	120	90	60	150

Table 6: Cluster quality comparisons between the GC, LC and CC approach in a vertical clustering environment on artificial data

		Analysis			
Statistic		1	2	3	4
ϵ_{LC}^{A}	mean	0.1	56.46	69.3	61.9
	95% CI	[0.0,0.2]	[54.4,58.7]	[66.6,72.1]	[59.0,64.6]
ϵ_{LC}^{B}	mean	0.1	57.8	67.9	87.9
	95% CI	[0.0,0.2]	[55.8,59.9]	[65.6,70.9]	[83.5;93.0]
ϵ_{CC}^{A}	mean	0.1	56.7	57.6	68.4
	95% CI	[0.0,0.2]	[54.7,58.7]	[55.9,59.1]	[65.9,70.7]
ϵ_{CC}^{B}	mean	0.1	56.6	56.7	91.4
	95% CI	[0.0,0.2]	[54.1,58.9]	[54.2,58.9]	[88.5,94.8]
$\epsilon_{LC}^{AB} - \epsilon_{CC}^{AB}$	mean	0.1	1.2	22.9	-10.0
	95% CI	[0,0]	[-0.5,2.6]	[20.4,25.5]	[-12.9,-6.8]
$\epsilon_{CC}^{AB} - \epsilon_{GC}$	mean	0.1	2.0	23.5	-4.9
	95% CI	[-0.2,0.1]	[0.4,3.3]	[20.8,25.9]	[-7.8,-1.1]
$\epsilon_{LC}^{AB} - \epsilon_{GC}$	mean	0.1	0.8	0.6	5.1
	95% CI	[-0.2,0.1]	[0.0,1.9]	[-1.03,2.33]	[3.4,6.9]

which represented a problem which was no longer perfectly separable was a bigger challenge. The results show that both the CC and LC approach make some cluster assignment errors, i.e. around 56 cases out of 600. However, no statistically significant difference can be found between the error rate of the CC and LC approach. It should be noted that the LC approach still perform as good as the GC approach, which suggests that CC can only improve on LC in a vertical PSO-CFC setting if LC performs worse than GC. This confirms our idea that the CC approach mimics the GC approach.

The results from the third analysis, show that the error rate of the LC approach increases compared to the second approach. This must be caused by the fact that the clusters are no longer equally present within the local data sites. Apparently, unequal representations of clusters can make it more difficult to find the true clusters within the data when the clusters overlap. However, the CC approach does not seem to suffer from this new cluster challenge. The error

rate of the CC approach on both data sites remain equal to the previous analysis. A direct comparison of the cluster quality of the CC approach versus the GC approach, reveals that the CC approach still performs equally good as the GC approach. Compared with the LC approach, the results show that the CC approach results in significantly less cluster assignment errors.

The results of the fourth analysis, were rather surprising. Apparently, the LC approach outperforms both the GC and CC approach. The biased sampling produced data sets which were easier to cluster individually, than collectively. This suggests that the CC approach (and the GC approach!) should only be used under the assumption that the observations are sampled at random conditional on the cluster.

6 Remarks and Future Research

Some general remarks about the limitations of the current study should be made. Firstly, no real data set was used to confirm the results of the analysis of the vertical PSO-CFC algorithm. This was because the authors had no appropriate data sets at hand with compatible data describing the same set of features coming from different companies. We tried to resample data sets with replacement from the original customer satisfaction data set, but these new data samples have the same distribution as the original one which resembles the first analysis on the artificial data set for the vertical clustering variant. The results of this analysis showed that CC and LC approach performed equally well. These results were confirmed by our analysis on the resampled real-life data sets. Because no other meaningful analysis could be performed with the real-life data sets, they were not considered in the previous section. Future research should try to find appropriate real-life data sets for a vertical clustering approach and use it to test the three approaches.

Secondly, the horizontal PSO-CFC algorithm appeared to get stuck in local optima for some analysis on the artificial data sets. Although this did not occur on the real-life data sets and the overall conclusions on the artificial data sets were favoring the CC approach, future research should investigate this. The analysis should be repeated for other real-life data sets to verify if this problem is perhaps caused by the structure of the artificial data sets. On the other hand, it might be promising to search for an improvement of the current algorithm to prevent this situation.

Thirdly, the current version of the algorithm uses a heuristic (cf Eq. 8) to perform the cluster mapping across data sites, which does not offer a guarantee of a perfect mapping in every situation. However, the results show us no problems with the heuristic. Empirical research done by the authors seems to suggest that the CFC algorithm automatically creates a partially correct mapping because incorrect mappings are penalized by the second term of the augmented objective function of CFC. However, if the algorithm starts with an incorrect mapping, the CFC algorithm needs various runs only to create a correct mapping, before it can do the real collaboration. The authors have the impression from empirical

experiments that this mapping phase takes longer as the number of data sites and clusters increases. Currently, they are working on the integration of a binary integer programming problem to solve the mapping problem such that the CFC algorithm can directly start with the collaboration phase. Future research must show if this approach can increase the speed of the algorithm.

The latter aspect can be important if the current algorithm has to become resource-aware. In this article, the PSO-CFC algorithm has been introduced as KDubiq algorithm, but hasn't considered the computational complexity of the algorithm because the algorithm was developed with a KDubiq situations in mind where computing power is not a scarce resource, such as in P2P networks or grid computing. The current code has not been developed with resource restrictions in mind as can occur in other KDubiq environments such as sensor networks. It could be interesting to perform future research to create a sensor-network variant of the PSO-CFC algorithm. Currently, the complexity of the PSO-CFC algorithm is mainly dependent on the complexity of the CFC algorithm, since the PSO part only repeats the CFC algorithm for each particle a number of iterations long. Empirical experiments suggested that the algorithm is rather sensitive to the number of data sites and the number of clusters. However, exact analysis have not been performed in this study. As for the communication requirements, which is also important in sensor networks, the vertical version is much more interesting than the horizontal version since it only passes cluster prototypes. These are much smaller than partition matrices, which are passed by the horizontal version. There is still a lot of ground to cover on the resource-awareness aspect of the algorithm which creates some interesting paths for future research.

7 Conclusions

In this article, the authors presented a PSO driven collaborative clustering algorithm to the KDubiq community. This technique can address some typical issues in KDubiq research, such as privacy constraints and distributed computing. Previous research of PSO-CFC in non-KDubiq environments and the new empirical results in this study demonstrate the quality of this collaborative clustering approach.

As for the horizontal variant of PSO-CFC, the analysis on artificial data sets showed that the CC approach always outperforms the LC approach. Even if not a single data site has all the necessary features to separate the clusters or when a perfect separation of the clusters is not possible, even with all features from all data site, the CC approach makes less assignment errors and succeeds better in approximating the GC result. The latter was also confirmed by the analysis on the real-life data set.

As for the vertical variant of PSO-CFC, the analysis on artificial data sets showed that CC performs equally well as LC and GC when the data is sampled completely at random and the clusters are equally represented in the different data sets. When the data is drawn at random conditional on the clusters, but

the clusters are no longer equally represented in the data set, the CC approach significantly outperforms the LC approach and performs equally well as the GC approach. Only when the sampling is biased, we can not be certain that CC performs equally well as LC. The analysis showed that, a biased sampling led to better clustering results for LC. However, one has to be careful to generalize this conclusion since our experiment only implemented one specific type of bias.

Overall, the collaborative clustering algorithms are very suitable for applications in certain 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, pp. 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. Ayad, H., Kamel, M.: Finding Natural Clusters Using Multi-Cluster Combiner Based on Shared Nearest Neighbors. In: Proc. 4th Int. Workshop on Multiple Classifier Systems, pp. 166–175 (2003)
6. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus Clustering: a Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. Machine Learning, 52, pp. 91–118 (2003)
7. Strehl, A., Ghosh, J.: Cluster Ensembles: a Knowledge Reuse Framework for Combining Multiple Partitions. Journal of Machine Learning Research, 3, pp. 583–617 (2002)
8. Topchy, A., Jain, K., Punch, W.: Clustering Ensembles: Models of Consensus and Weak Partitions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27, pp. 1866–1881 (2005)
9. Dudoit, S., Fridlyand, J.: Bagging to Improve the Accuracy of a Clustering Procedure. Bioinformatics 19, 1090–1099 (2003)
10. Merugu, S., Ghosh, J.: Privacy-Preserving Distributed Clustering using Generative Models. In: Proc. of the Third IEEE International Conference on Data Mining, pp. 211–218 (2003)
11. Vaidya, J., Clifton, C.: Privacy-Preserving K-Means Clustering over Vertically Partitioned Data. In: Proc. of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 206–215 (2003)
12. 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)
13. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)

14. Hoppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis. John Wiley, Chichester (1999)
15. 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)
16. 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)
17. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures. CRC Press, Boca Raton, FL (1997)
18. Cohen, P.R.: Empirical Methods for Artificial Intelligence. The MIT Press (1995)
19. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall (1993)
20. Fox J.: Bootstrapping Regression Models. `http://socserv.mcmaster.ca/jfox/Books/Companion/appendix-bootstrapping.pdf`. An appendix to: An R and S-Plus Companion to Applied Regression, Sage Publications (2002)