Journal of Software Engineering

ISSN 1819-4311

Academic Journals Inc.

www.academicjournals.com

Journal of Software Engineering 5 (4): 116-126, 2011
ISSN 1819-4311 / DOI: 10.3923/jse.2011.116.126
© 2011 Academic Journals Inc.

Distributed K-means based-on Soft Constraints

[1,2]Y.C. Yu, [1]J.D. Wang, [1]G.S. Zheng and [3]Y. Jiang
[1]College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China
[2]College of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang, 212003, China
[3]College of Information Engineering, Yangzhou University, Yangzhou, 225009, China

Corresponding Author: Y. Yuecheng, College of Computer Science and Engineering, Jiangsu University of Science and Technology, China Tel: (86)13921587271

ABSTRACT

Pairwise constraints can effectively improve the clustering results. However, noise constraints will seriously affect the performance of clustering. To improve the distributed clustering with constraints, distributed k-means based-on soft constraints, which constraint violations can be effectively dealt with, is presented in this paper. Aiming at the limitation of distributed clustering, such as communication cost and data privacy etc., only positive constraints by chunklets are used in the proposed method. To simplify the treatment of constrained data points, the mean value of chunklet is used as the representative point. Then positive constraints among chunklet are approximately transformed into pairwise positive constraints between each data points from the chunklet and the mean value. Thus, the cluster label of each mean value is regarded as the label estimation of data points from the chunklet. Based on the above approximation, a new measure of partition cost used to deal with constraint violations is defined. Therefore, for unconstrained data points, the within-cluster sum of distance squares can be minimized. Meanwhile, for constrained data points, the sum of distance between data points and corresponding centriods and the cost of constraint violations is minimized too. The experimental results showed that the proposed method decreases the computation complexity of constraint violations. Compared with hard constrained distributed clustering, the clustering accuracy of the proposed method is increased.

Key words: Constrained clustering, parallel k-means, positive constraints, constraints transformation, constraint violations

INTRODUCTION

Machine learning algorithms are often designed to deal with centralized data. However, in many application scenes, data is distributed among several sites, where each site generates its own data and manages its own data repository (Park and Kargupta, 2002). From the point of view of data partition, this belongs to the horizontal partitioning data set, i.e. all sites contain the same attributes of data but different data records (Johnson and Kargupta, 1999; Tasoulis and Vrahatis, 2004). Clustering is one of the major tasks of machine learning. In the distributed scene, the data set are generated by many companies or institutes. To obtain the global clustering results, all distributed data sets are centralized on the central site. When huge amounts of data are frequently produced at different sites, the cost of communication, storage and computation will dramatically increase. In addition, this approach will suffer from the problem of data privacy

protection (Kriegel *et al*., 2005). Obviously, it is inappropriate to centralize distributed data into a data warehouse on which to apply the traditional clustering methods (Klusch *et al*., 2003).

To effectively deal with the distributed data set, many traditional clustering had been extended. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is the clustering algorithm based on density (Ester *et al*., 1996). Density Based Distributed Clustering (DBDC) is the distributed version of DBSCAN and performed in each local clustering phase which the local clusters are represented by special objects that have the best representative power, then DBSCAN is used once again in global clustering (Januzaj *et al*., 2004a). However, DBDC suffers from some drawbacks, such as the local noise and the number of representatives. To overcome the above limitations, the scalable density-based distributed clustering was presented by Januzaj *et al*. (2004b). Recursive Agglomeration of Clustering Hierarchies by Encircling Tactic (RACHET) and the Collective Hierarchical Clustering (CHC) are the two distributed version of hierarchical clustering algorithm (Samatova *et al*., 2002). RACHET performes the hierarchical clustering and the local dendrograms are created on each site, then the centroid descriptive statistics are sent for each cluster. CHC works on data that is heterogeneously distributed, with each site having only a subset of all features.

As k-means is widely used to clustering the centralized data set, many distributed versions of k-means have been proposed. Sanghamitra Bandyopadhyay proposed the extension version of k-means to be used to the clustering of distributed data stream (Bandyopadhyay *et al*., 2006). The other important distributed version of k-means is parallel k-means (Dhillon and Modha, 1999). Parallel k-means make use of the inherent parallelization capabilities of k-means and can run on the parallel supercomputers. This make the algorithm adapt to the clustering of distributed data set. Semi-supervised clustering can significantly improve the clustering performance by introcucing some limited supervision information to the process of clustering. Inspired by the idea of semi-supervised learning, (Yuecheng *et al*., 2011) proposed the extend version of parallel k-means based on constrained information. During the process of distributed clustering, the constraints coming from each site can lead the algorithm to bias the search for an appropriate clustering of the data.

Constraints are generated from background knowledge about the data set or from a subset of the data with known labels. To constrained clustering, it is easier to specify pairwise constraints, namely whether pairs of points belong to the same cluster or different clusters, than to provide the class labels of the points. Pairwise Constrained Clustering (PCC) (Basu *et al*., 2004) is one of the major constrained clustering algorithms and used to clustering the centralized data set. As described in PCC, the inappropriate constraints selection and the noise constraints will decrease the performance of constrained clustering (Davidson *et al*., 2006). The similar case will also appear for distributed constrained clustering. To decrease the difficulty of constraints selection and hold the performance of distributed parallel k-means based on constraints, the Distributed k-means Based on Soft Constraints (DBSC) is proposed in this study. In the framework of DBSC, each site user can select the pairwise constraints according to their background knowledge. Sometimes noise constraints may exist. By penalizing the constraint violations, DBSC can effectively deal with noise constraints and the accuracy of clustering can be guaranteed.

RELATED WORK

As one of the distributed version of k-means, parallel k-means has been widely used to the clustering of distributed data set, such as distributed data streams or P2P networks. Depending on

the method of semi-supervised clustering, constrained parallel k-means is proposed by Yuecheng *et al.* (2011) and the clustering accuracy of distributed clustering is improved. The following will simply describe the above methods, which are the foundation of DBSC.

Parallel k-means: Assuming data set X = $\{x_1, x_2,..., x_N\}$ is distributed on L sites, where x_i (i = 1,..., N) is the ith point and N is the total number of data points. If S_l (l = 1,..., L) denotes the data subset locating on the l^{th} site and L is the number of sites, then the data set X can be denoted as $X = \cup_{l=1}^{L} S_l$ Accordingly, $x_{l,i} \in S_l$ denotes the ith point locating on the site and $|S_l|$ denotes the number of data points locating on the l^{th} site. To partition the data set X into K clusters, parallel k-means needs iterated search, which is similar to the standard k-means. The objective function of parallel k-means can be formulated as formular (1) (Dhillon and Modha, 1999).

$$J_{Pkm} = \min \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{x_{l,i} \in C_k} \| x_{l,i} - m_k \|^2 \quad (1)$$

where, $\{C_k\}_{k=1}^{K}$ denotes the K clusters, $|C_k|$ is the number of data points that their cluster labels are K and $\{m_k\}_{k=1}^{K}$ is the corresponding cluster centroids.

Constrained parallel k-means: There are two forms of pairwise constraints in semi-supervised clustering, namely must link and cannot link. If pariwise data points (x_i, x_j) satisfies must link constraint, then pairwise data points x_i and x_j should have the same cluster labels. Accordingly, if pariwise data points (x_i, x_j) satisfies cannot link constraint, then they should have the different cluster labels. Moreover, when more data points are known having the same cluster labels, these data points can be grouped as a small set, called chunklet (Bar-Hillel *et al.*, 2005). In distributed environment, there are only partial data points at each site, so site users can specify whether pairwise data points have same cluster labels but cannot confirm whether pairwise data points have different cluster labels. In fact, even though the two data points stored on the same site are quite different, they are possibly partitioned into the same cluster in the process of global clustering. However, it is relatively easy to specify whether the pairwise data points must have the same cluster labels. This means that only positive constraints (must link) can be used. In the framework of Constrained Parallel k-means (CPKM), chunklets are introduced into the objective function as positive constraints, where each chunklet is the subsets of data set X and all data points belonging to the same chunklet must be generated from the same site. The objective function of CPKM is formulated as formula (2) (Yuecheng *et al.*, 2011).

$$J_{CPkm} = \min \sum_{k=1}^{K} \sum_{l=1}^{L} \left(\sum_{x_j^l \in C_{l,k} \setminus H_{l,k}} \| x_{l,i} - m_k \|^2 + \sum_{a=1}^{|H_{l,k}|} \| m_{l,a} - m_k \|^2 \right) \quad (2)$$

where, H is the set of chunklets generated from all sites and H_l is the subset of chunklets generated from the lth site, then H can be denoted as $H = \{H_l\}_{l=1}^{L}$, Assuming that $H_{l,k} \subset H_l$ is the chunklet set which all of them are on site l and will be assigned to cluster k, $|H_{l,k}|$ is the number of $H_{l,k}$. For each chunklet $H_{l,a} \in H_l$, the mean value of $H_{l,a}$ is denoted as $m_{l,a}$. To make all data points from the same chunklet have the same cluster labels, the cluster label of mean of each chunklet is used as the representative of the chunklet. When the distance between the mean of

chunklet and the cluster centroid is closest, the average distance between all data points from the chunklet and the centroid will also be closest. Essentially, the second term of the objective function of CPKM is used to deal with the chunklet set H. By this means, all data points from the same chunklet can be assigned to the same cluster and the minimum average distance between the data points from the chunklet and the centroids is guaranteed.

DISTRIBUTED K-MEANS BASED-ON SOFT CONSTRAINTS

To the clustering framework of CPKM, constraints are introduced to the process of distributed clustering by chunklets and the constraints between the pairwise data points can effectively improve the clustering accuracy. The framework of CPKM implies that there is no noise constraint in all chunklets. Essentially, CPKM is the distributed framework of hard constrained clustering. In fact, as the prior information, chunklets come from the prior knowledge of site users. In distributed scene, site users can only know the local information about the data points and cannot know the data points locating on the other sites. Meanwhile, there are some cognitive limitations for site users to completely understand the data points on the own site. Then, the constraints provided by site users may be error. In addition, different site users may have different background knowledge. This will lead to some conflicting constraints. If there are some error constraints or conflicting constraints, the clustering accuracy will reduce rapidly (Davidson *et al*., 2006). To decrease the influence from noise constraints, the objective function of CPKM is modified and the constraints described by chunklets are transformed into must link between pariwise data points in this paper. By penalizing the constraint violations, the cluster labels of each data point belonging to chunklets will be determined by the cost of constraint violations and the distance between data point and the centroids.

Constraints selection and representation: The basic idea of constrained clustering is to use the background knowledge to lead the clustering bias search (Basu *et al*., 2004). Constraints may come from either labeled or unlabeled data points. For labeled data points, their cluster labels have been specified, however, constraints from unlabeled data points only specify whether the pairwise data points have the same cluster labels and the concrete cluster labels of the pairwise data points have not been described. The latter is called pairwise constraints. Compared with labeled data points, pairwise constraints can be obtained easily using background knowledge and the acquisition cost is more small. In this study, only the pairwise constraints are used.

In distributed clustering secne, the constraints have their special forms and requirements, which are different from the centralized clustering. When the distributed data points need to be partitioned, some elements, such as communication cost and data privacy etc., must be considered. Then the pairwise constraints can only be provided by site users according to their background knowledge and each pairwise data points must be located on the same site. Meanwhile, the constraints can only be seen by themselves and cannot be transformed between different sites. This means that constraints are distributed on each site and the constraints information, described as pairwise data points, can only be used to the process of local clustering, which is performed on each site according to the global clustering standard. In addition, for the reason of distributed data set, site users cannot determine whether the pairwise data points must have the different cluster labels. Similar to the clustering framework of CPKM, chunklets are used to introduce positive constraints to distributed clustering in this study.

Cost of constraint violations: In the clustering framework of PCC, weight cost of constraint violations is defined and used to punish the constraint violations during the process of clustering. Then the algorithm of constrained k-means is performed and partitions, denoted as $\{C_k\}_{k=1}^{K}$, can be obtained, where the sum of total within-cluster distance and total cost of constraint violations is minimized. Since the cost of constraint violations is relevant to the cluster assignment order of pairwise data points, in order to obtain the local optimization, greedy strategy is adopted to compute the total partition cost of pairwise data points. As the centralized constrained clustering, all weight costs of constraint violations are equivalent. Obviously, this is inappropriate because of the difference of constraints confidence degree (Bilenko *et al.*, 2004). In this study, the penalty of constraint violations is also used during the process of clustering distributed data points. However, the different measure pattern of weight cost of constraint violations and different treatment strategy of constraint violations are adopted.

For the distributed clustering framework of DBSC, the cluster assignment of the constrained pairwise data points should be determined according to the sum of the distance between the data points and cluster centroids and the cost of constraint violations. This is similar to constrained clustering framework of PCC. Assuming that $CH_{l,a} \subseteq Ch_l$ denotes the a^{th} chunklet of CH_l. Let $Z(x_{l,i})$ and $Z(x_{l,j})$ represent the cluster labels of $x_{l,i}$ and $x_{l,j}$, respectively. For all $x_{l,i}, x_{l,j} \in CH_{l,a}$ if $Z(x_{l,i})$ and $Z(x_{l,j})$ are not equal each other, this means that constraint violation appears.

Each chunklet is composed of data points with same cluster labels. The cluster assignment of each data point will affect the cost of constraint violation of other data points. Obviously, it is difficult to directly deal with these constraint violations. Let $\mu_{l,a}$ is the mean value of $CH_{l,a}$ and $Z(\mu_{l,a})$ is the cluster label of $\mu_{l,a}$. As described in CPKM, when $\mu_{l,a}$ is used as the representative point of $CH_{l,a}$, $Z(\mu_{l,a})$ can be regarded as the estimation of cluster labels of all data points in $CH_{l,a}$, then the average distance between each data point in $CH_{l,a}$ and the cluster centroid is minimum in average meaning. That is to say that the average probability, which cluster labels of all data points from $CH_{l,a}$ is $Z(\mu_{l,a})$, will be maximum. So, the positive constraints between all data points in a chunklet can be approximately transformed into the positive constraints between each data points and the mean of chunklet. Then, for each $x_{l,i} \in CH_{l,a}$, if $Z(x_{l,i})$ and $Z(\mu_{l,a})$ are not equal, it can be regarded as constraint violation. By means of this approach, the effect from the order of pairwise data points can be avoided during the process of estimating the cost of constraint violations.

Let $m_{z(\mu_{l,a})}$ and $m_{z(x_{l,i})}$ are the cluster centriods, which are corresponding to the cluster labels, $Z(\mu_{l,a})$ and $Z(x_{l,i})$, respectively. Intuitively, For each data point $x_{l,i} \in CH_{l,a}$, all of them may have the same cluster labels with $\mu_{l,a}$. Once constraint violation appears, the distance between $x_{l,i}$ and $m_{z(\mu_{l,a})}$ will not be the minimum value. If $m_{z(\mu_{l,a})}$ can shift toward $x_{l,i}$, the value of $Z(\mu_{l,a})$ and $Z(x_{l,i})$ may become equal. Meanwhile, if $x_{l,i}$ can deviate from $m_{z(x_{l,i})}$, the possibility will be increased to make the value of $Z(\mu_{l,a})$ and $Z(x_{l,i})$ become equal. According to the assumption, for each data point $x_{l,i}$, if it satisfies the positive constraint, the cluster assignment will not depend on the single distance between the data point and the cluster centriods; however, the combined measure, namely the weight sum of the distance between $x_{l,i}$ and $m_{z(\mu_{l,a})}$ and the distance between $x_{l,i}$ and each centriod, will be used. If let $d(\cdot)$ denote the square of Euclidean distance, then $d(x_{l,i}, m_{z(\mu_{l,a})})$ denotes the square of Euclidean distance between $x_{l,i}$ and $m_{z(\mu_{l,a})}$, $d(x_{l,I}, m_k)$ denotes the square of Euclidean distance between $x_{l,i}$ and the cluster centriod which $x_{l,i}$ will be assigned to. Thus, for each $x_{l,i} \in CH_{l,a}$, the total cost of cluster assignment, denoted as $G(x_{l,i}, m_k)$ can be computed according to formula (3).

$$G(x_{l,i}, m_k) = (1-\beta_{l,a}) \cdot d(x_{l,i}, m_{z(\mu l,a)}) + \beta_{l,a} \cdot d(x_{l,i}, m_k) \quad (3)$$

where, parameter $\beta_{l,a}$ is the penalty weight of constraint violations and will be used to balance the effect from $m_{z(\mu_{l,a})}$ and m_k during the process of computing the total cost of cluster assignment. When $m_{z(x_{l,i})}$ and $Z(\mu_{l,a})$ are equal, the value of $G(x_{l,i}, m_k)$ and $d(x_{l,i}, m_{z(\mu_{l,a})})$ will also be equal. When $Z(x_{l,i})$ and $Z(\mu_{l,a})$ are not equal, the value of $G(x_{l,i}, m_k)$ will be the weight sum of $d(x_{l,i}, m_k)$ and $d(x_{l,i}, m_{z(\mu_{l,a})})$.

As described above, different penalty weight of constraint violations should be adopted for different pairwise constraints. In average meaning, the constraint strength between $x_{l,i}$ and $\mu_{l,a}$ will become weak with the distance increase. Then the proportion of $d(x_{l,i}, m_{z(\mu_{l,a})})$ in the value of $G(x_{l,i}, m_k)$ should be decreased and the value of $\beta_{l,a}$ should be smaller. On the contrary, the constraint strength between $x_{l,i}$ and $\mu_{l,a}$ will become strong with the distance decrease. Accordingly, the proportion of $d(x_{l,i}, m_{z(\mu_{l,a})})$ in the value of $G(x_{l,i}, m_k)$ should be increased and the value of $\beta_{l,a}$ should be larger. In addition, the distance between $x_{l,i}$ and $m_{z(\mu_{l,a})}$ is the other factor which will affect the value of $\beta_{l,a}$. So, the penalty weight of constraint violations can be defined as formula (4).

$$\beta_{l,a} = \frac{\| x_{l,i} - \mu_{l,a} \|^2}{\| x_{l,i} - m_{Z(\mu_{l,a})} \|^2 + \| x_{l,i} - \mu_{l,a} \|^2} \quad (4)$$

For all data points from chunklets, the cluster labels will be determined by the weight sum of distance between the data point and the centriods and the cost of constraint violations. For each $x_{l,i} \in CH_{l,a}$, if $Z(x_{l,i}) \neq Z(\mu_{l,a})$ appears, the cluster assignment should be punished. If there is no appearance of constraint violation, the total cost of cluster assignment is only the distance between data point and the corresponding centroid. Therefore, after the positive constraints among data points in a chunklet are transformed into the pairwise positive constraints specified by each data point and the mean value of chunklet, meanwhile, the cluster label of the chunklet's mean is regarded as the estimation of cluster labels, the computational complexity of the estimation of cluster labels is effectively simplified.

Objective function: In the clustering framework of parallel k-means, all data points are partitioned into K disjoint clusters, where the within-cluster sum of distance squares is minimized. Essentially, the distance between each data points and K centriods must be computed and each data point is partitioned into the closest cluster. So, the distance between data points and centriods can be regarded as the partition cost of data points. DBSC is similar to the parallel k-means in nature. For the data points without any constraints, the same measure of partition cost can be adopted. That is to say that their cluster labels can be determined by the single distance between data points and corresponding centroids . However, to determine the cluster labels of all data points from chunklets, the value of cost function defined in formula (3) need to be computed. In this case, the partition cost is composed of the weight sum of distance between data points and centroids, which the data points will be assigned to and the distance between data points and the centroid estimated by mean value of chunklet. To realize the above objective, the combined objective function is defined as formula (5) in DBSC framework and will be minimized by iterative search.

$$J_{DBSC} = \min \sum_{k=1}^{K}\sum_{l=1}^{L}\Big(\sum_{\substack{x_{l,i}\notin CH_l \\ x_{l,i}\in C_{l,k}}} \| x_{l,i} - m_k \|^2 + \sum_{a=1}^{|CH_l|} \sum_{\substack{x_{l,i}\in CH_l \\ x_{l,i}\in C_{l,k}}} G(x_{l,i}, m_k) \tag{5}$$

Let formula (3) is substituted into formula (5), then the objective function of DBSC can be rewritten as formula (6).

$$\begin{aligned} J_{DBSC} = \min \sum_{k=1}^{K}\sum_{l=1}^{L}\Big[& \sum_{\substack{x_{l,i}\notin CH_l \\ x_{l,i}\in C_{l,k}}} \| x_{l,i} - m_k \|^2 \\ & + \sum_{a=1}^{|CH_l|} \sum_{\substack{x_{l,i}\in H_{l,a} \\ x_{l,i}\in C_{l,k}}} ((1-\beta_{l,a})\cdot \| x_{l,i} - m_{Z(\mu_{l,a})} \|^2 + \beta_{l,a}\cdot \| x_{l,i} - m_k \|^2)\Big] \end{aligned} \tag{6}$$

To minimize the formula (6), data set is partitioned into two subsets, namely constrained data subset and unconstrained data subset. For the unconstrained data points, the same measure adopted by parallel k-means can be used to determine their cluster labels and the partition cost, denoted as d ($x_{l,i}$, m_k) can be computed according to formula (7). The formula (8) is the definition of partition cost of all constrained data points. According to formula (8), this kind of partition cost, denoted as d ($x_{l,i}$, m_k), need simultaneously measure the distance between data points and centeriods and the penalty quantity of constraint violations. Thus, by minimizing the formula (6), the within-cluster sum of distance squares can be minimized for unconstrained data points, meanwhile, the sum of distance between data points and corresponding centriods and the cost of constraint violations is minimized for constrained data points.

$$d(x_{l,i}, m_k) = \| x_{l,i} - m_k \|^2 \tag{7}$$

$$\tilde{d}(x_{l,i}, m_k) = (1-\beta_{l,a})\cdot \| x_{l,i} - m_{Z(\mu_{l,a})} \|^2 + \beta_{l,a}\cdot \| x_{l,i} - m_k \|^2 \tag{8}$$

Algorithm: As the extension of parallel k-means, the minimization of the objective function of DBSC is also the process of iterating search. By performing E step and M step alternatively, cluster assignment and the estimation of cluster centriods can be realized respectively. In E step, according to whether the data points satisfy the constraints, different measure of partition cost is adopted to minimize the objective function. In M step, the local cluster centroid $m_{l,k}^{(t+1)}$ is estimated according to the cluster assignments of all data points locating on the tth site, where $m_{l,k}^{(t+1)}$ means the kth local cluster centroid on the tth site in the t+1 iteration step and can be computed according to formula (9). Then, all local cluster centriods and the corresponding sample numbers of clusters are sent to central site and the global cluster centroid $m_k^{(t+1)}$ can be estimated according to the formula (10). In fact, the estimation of cluster centroids, including local and global cluster centroids, only relates to the cluster assignments of data points and is independent of the constraints.

$$m_{l,k}^{(t+1)} = \frac{1}{|C_{l,k}^{(t+1)}|} \sum_{x_{l,i}\in C_{l,k}^{(t+1)}} x_{l,i} \tag{9}$$

$$m_k^{(t+1)} = \frac{1}{|C_k^{(t+1)}|}\sum_{l=1}^{L} C_{l,k}^{(t+1)} \cdot m_{l,k}^{(t+1)} \quad (10)$$

where, $C_{l,k}^{(t+1)}$ is the local cluster in which each data point locates on the l site and all cluster labels of them are k in the t+1 iteration step and $C_k^{(t+1)}$ is the global cluster in which all cluster labels of them are k in the t+1 iteration step. Correspondingly, $|C_{l,k}^{(t+1)}|$ represents the sample number of local cluster $C_{l,k}^{(t+1)}$ and $|C_k^{(t+1)}|$ represents the sample number of global cluster $C_k^{(t+1)}$. The algorithm of DBSC can be described as following.

Input: Data set $X = \cup_{l=1}^{L} S_l$, constraints set of chunklets $CH = \{CH_l\}_{l=1}^{L}$ and the corresponding mean set $\{\mu_{l,1}, ..., \mu_{l,|CH_l|}\}_{l=1}^{L}$, the cluster numbers K.

Output: K partitions $\{C_k\}_{k=1}^{K}$ to minimize the objective function (6).

Method
Initialization: Randomly choose K centers to initialize the clusters;

Repeat until convergence:

- At each site, the global cluster centriods $\{m_k^{(t)}\}_{k=1}^{K}$ are received from center site, then the cluster labels of mean value of each chunklet are estimated and the corresponding penalty weight of constraint violations, namely $\beta_{l,a}$, are computed, respectively
- For each data point $x_{l,i}$, if $x_{l,i} \in CH_l$ after computing its partition cost according to formula (7), it should be partitioned into the closest cluster; otherwise, computing its partition cost according to formula (8), it should be partitioned into the cluster with the minimum partition cost
- Computing the value of each $|C_{l,k}^{(t+1)}|$ at all sites, meanwhile, all local cluster centroids $m_{l,k}^{(t+1)}$ should be computed according to the formula (9), then the value of each $|C_{l,k}^{(t+1)}|$ and $m_{l,k}^{(t+1)}$ should be sent to center site
- After computing the global centroids $\{m_k^{(t+1)}\}_{k=1}^{K}$ according to formula (10), the new centriods are sent to each site

If a convergence criterion is not satisfied, go to step 2; until convergence.

EXPERIMENTAL EVALUATION

The test of algorithm performance is executed in the simulating distributed scene and the test data set includes synthetic data set and UCI data sets. The synthetic data set contains 60,000 random data points generated by BNT package according to Gaussian mixture model. The original Gaussian mixture model is consist of three Gaussian components. Then the synthetic data points are randomly partitioned into 3 data subset with equal data size and located on 3 sites, where each site owns 20000 data points.

Two UCI data sets, namely Iris data set and animals data set, are used in the performance test. Iris data set contains 3 classes of 50 data points each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. The data set of animals contains 50,000 data points and its true partitions are composed of 4 classes. To every data point, there is 72 attributes and the value of the 74th attribute is used to record the class label.

Table 1: The clustering accuracy of synthetic data set

Clustering accuracy	Parallel k-means	CPKM[1]	DBSC[2]
Site 1	0.806212	0.876391	0.919635
Site 2	0.825365	0.886714	0.921355
Site 3	0.813516	0.895217	0.926563

Table 2: The average clustering accuracy of the three algorithms

Methods	Parallel k-means	CPKM[1]	DBSC[2]
Clustering accuracy	0.8719	0.9132	0.93106

[1]CPKM is the abbreviation of constrained parallel k-means proposed by Yuecheng *et al*. (2011). [2]DBSC is the abbreviation of Distributed k-means Based on Soft Constraints proposed in this study

Table 3: The clustering accuracy of Iris and animals data set

Data set	Iris	Animals
Site 1	0.870020	0.7913261
Site 2	0.876200	0.7892003
Site 3	0.869531	0.7935410

During the process of test performance using UCI data set, Iris data set are partitioned into 3 data subsets and each subset contains 50 data points. For animals data set, 5,000 data points are randomly selected and partitioned into 3 data subsets located on 3 sites, where each site contains 1650, 1500 and 1850 data points, respectively.

To verify the validity of DBSC, the correlative algorithms, including parallel k-means, CPKM and DBSC, are implemented on the synthetic data set and Iris data set respectively. Table 1 shows the local clustering accuracy about all sites performed on synthetic data set. Accordingly, Table 2 shows the global clustering accuracy performed on Iris data set. As described in Table 1 and 2, no constraints are used in parallel k-means, its local accuracy of all sites and the global accuracy are lower than the other two methods, namely CPKM and DBSC. The results show that constraints improve the distributed clustering, which is similar to the centralized clustering. Compared to CPKM, constraint violations are allowed and can be effectively processed in the framework of DBSC. When the two methods are performed on the data set with noise constraints, DBSC will be more robust than CPKM and the better result can be obtained.

Then DBSC is performed on Iris data set and the subset of animals data set, respectively. Table 3 shows the results of clustering accuracy. The data points of Iris data set have low dimensional, then the cost of constraint violations are sensitive to the Euclidean distance. So, when the constraints are introduced to the process of clustering, the clustering accuracy are improved obviously. For constrained distributed clustering, the cluster assignments are led by the cost of constraint violations. However, the data points of animals data set have higher dimensional. This will decrease the effect from the centroids estimated by means value of chunklets. As described in Table 3, the clustering accuracy of animals data set is improved but it is lower than the accuracy of Iris data set.

CONCLUSION

DBSC was a modified version of constrained distributed clustering framework of CPKM, which could effectively deal with the soft constraints by punishing the constraint violations. Depending on the mean value of chunklet, the representation of constraints and the cost estimation of

constraint violations were simplified. This led the computation complexity of constraint violations to be decreased and the effect from the order of pairwise data points during the process of estimating the cost of constraint violations to be avoided. Based on the definition of weight partition cost and minimizing the objective function, the clustering result satisfying the user's preference can be obtained. In fact, the within-cluster sum of distance squares for unconstrained data points and the cost of constraint violations for constrained data points were minimized simultaneously. The experiment results showed that the framework of DBSC could effectively use the noise constraints to increase the clustering accuracy of distributed k-means.

ACKNOWLEDGMENTS

This study is supported by the National Natural Science Foundation of China under Grant No. 61170201.

REFERENCES

Bandyopadhyay, S., C. Giannella, U. Maulik, H. Kargupta, K. Liu and S. Datta, 2006. Clustering distributed data streams in peer to peer environments. Inform. Sci., 176: 1952-1985.

Bar-Hillel, A., T. Hertz, N. Shental and D. Weinshall, 2005. Learning a mahalanobis metric from equivalence constraints. J. Mach. Learn. Res., 6: 937-965.

Basu, S., A. Banerjee and R.J. Mooney, 2004. Active semi-supervision for pairwise constrained clustering. Proceedings of the SIAM International Conference on Data Mining, (SDM-2004), April 22-24, 2004, Lake Buena Vista, FL, pp: 333-344.

Bilenko, M., S. Basu and R.J. Mooney, 2004. Integrating constraints and metric learning in semi-supervised clustering. Proceedings of 21st International Conference on Machine Learning, July 4-8, 2004, Banff, Canada, pp: 81-88.

Davidson, I., K.L. Wagstaff and S. Basu, 2006. Measuring constraint-set utility for partitional clustering algorithms. Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, September 18-22, 2006, Berlin, Germany, pp: 115-126.

Dhillon, I.S. and D.S. Modha, 1999. A data-clustering algorithm on distributed memory multiprocessors. Proceedings of the KDD'99 Workshop on High Performance Knowledge Discovery, (KDD'99), San Digeo, USA., pp: 245-260.

Ester, M., H.P. Kriegel, J. Sander and X. Xu, 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, (ICKDDM'96), Portland, pp: 226-231.

Januzaj, E., H.P. Kriegel and M. Pfeifle, 2004a. DBDC: Density-based distributed clustering. Proceedings of the 9th International Conference on Extending Database Technology (EDBT), March 2004, Heraklion, Greece, pp: 88-105.

Januzaj, E., H.P. Kriegel and M. Pfeifle, 2004b. Scalable density-based distributed clustering. Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, September 20-24, 2004, Pisa, Italy, Springer, pp: 231-244.

Johnson, E.L. and H. Kargupta, 1999. Collective, hierarchical clustering from distributed, heterogeneous data. Large Scale Parallel KDD Syst., 1759: 221-244.

Klusch, M., S. Lodi and G. Moro, 2003. Distributed clustering based on sampling local density estimates. Proceedings of the 18th International Joint Conference on Artificial intelligence, August 2003, Acapulco, Mexico, pp: 485-490.

Kriegel, H.P., P. Kroger, A. Pryakhin and M. Schubert, 2005. Effective and efficient distributed model-based clustering. Proceedings of the 5th IEEE International Conference on Data Mining, November 27?30, 2005, IEEE Computer Society, Washington, DC, USA., pp: 258-265.

Park, B.H. and H. Kargupta, 2002. Distributed Data Mining: Algorithms, Systems and Applications. In: The Handbook of Data Mining, Park, B.H. and H. Kargupta (Eds.). Lawrence Erlbaum Associates Publishers, New Jersey, United States, pp: 341-358.

Samatova, N.F., G. Ostrouchov, Al Geist and A.V. Melechko, 2002. Rachet: An efficient cover-based merging of clustering hierarchies from distributed datasets. Distrib. Parallel Databases, 11: 157-180.

Tasoulis, D.K. and M.N. Vrahatis, 2004. Unsupervised distributed clustering. Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks, February 17-19, 2004, IASTED/ACTA Press, Austria, pp: 347-351.

Yuecheng, Y., W. Jiandong, Z. Guansheng and C. Bin, 2011. Parallel k-means algorithm based on constrained information. J. Southeash Univ. Nat. Sci. Ed., 41: 505-508.