Unit Volume Based Distributed Clustering Using Probabilistic Mixture Model*

Keunjoon Lee[1], Jinu Joo[2], Jihoon Yang[2], and Sungyong Park[2]

[1] Kookmin Bank, 27-2,
Yeouido-Dong, Yeoungdeungpo-Ku, Seoul, Korea
leekjsg@hanmail.net
[2] Department of Computer Science and
Interdisciplinary Program of Integrated Biotechnology, Sogang University,
1 Shinsoo-Dong, Mapo-Ku, Seoul 121-742, Korea
jujoo@mllab.sogang.ac.kr,
{yangjh, parksy}@sogang.ac.kr

Abstract. Extracting useful knowledge from numerous distributed data repositories can be a very hard task when such data cannot be directly centralized or unified as a single file or database. This paper suggests practical distributed clustering algorithms without accessing the raw data to overcome the inefficiency of centralized data clustering methods. The aim of this research is to generate unit volume based probabilistic mixture model from local clustering results without moving original data. It has been shown that our method is appropriate for distributed clustering when real data cannot be accessed or centralized.

1 Introduction

Data clustering is a method of grouping or partitioning similar patterns to subsets. Patterns that are grouped in the same cluster can be analyzed to have closer relationship than other patterns in different clusters. Clustering algorithms are applied in various areas such as visualization, pattern recognition, learning theory, computer graphics, and neural networks [1]. Recently, issues on distributed clustering has arisen. Distributed clustering is particularly useful when distributed data are hard to be centralized because of privacy, communication cost, and the limit of storage. Against this background, we suggest practical distributed algorithms.

Our method runs in three steps. First, local clustering results are gathered without moving the original data. Second, based on the received results (mean and covariance that represent the local clusters) each unit volume in the global data space is assigned to clusters with the highest probability. Finally, clusters with similar probability distributions are merged.

* This work is supported by grant No. R01-2004-000-10689-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

Unit volume based distributed clustering has three advantages compared to other existing distributed clustering methods. First, distributed data are clustered in parallel without physically moving the data, and each local site may run different clustering algorithms. Second, the size of the unit volume can be configured freely based on the size and domain of the data, so that global clustering can approximate on local data distributions. Third, when merging normally distributed models in global clustering, a diversity of mixture models can be deduced during the process which will lead to near optimal clustering models that represent overall data.

Two types of distributed data exist: *horizontally* distributed data where data are distributed by instance, and *vertically* distributed data where data are distributed by attributes. In this paper we concern only horizontally distributed data to perform clustering.

2 Related Work

Two major types of distributed algorithms exist depending on the relationship between global clustering and local clustering. The first type is where the local clustering algorithm is related to the global clustering algorithm. The other type is where the local and global clustering is independent and does not interfere with each other. The former includes DBDC [2] and k-windows distributed clustering [3,4], and the latter includes privacy preserving distributed clustering [5].

DBDC is an extension of DBSCAN [6] which is a density based clustering algorithm. It performs DBSCAN in each local clustering phase and DBSCAN is used once again in global clustering. This method requires large memory space for saving every distance value between data as a tree. And global clustering algorithm is deeply related to local clustering algorithms. Therefore, different clustering algorithms can not be introduced separately for local clustering and global clustering.

K-windows distributed clustering uses k-windows algorithm to cluster data at each local site. Global clustering is performed by merging local clusters of high similarity. The drawback is that there are too many parameters that the user must define, and local clustering algorithm is bound to k-windows clustering algorithm, which means that the user doesn't have any freedom to use other clustering algorithms for local clustering.

Privacy preserving distributed clustering algorithm complements the inherent drawbacks of DBDC and k-windows distributed clustering, and let local clustering algorithms be independent from global clustering algorithms. In other words, each local site is allowed to use different clustering algorithms to cluster its local data. The drawback is that the user must define the universal set of potential clustering models and produce a dataset through MCMC sampling before global clustering.

Based on these backgrounds about distributed clustering, our algorithm not only allows each local site to choose its clustering algorithm freely but the user only needs to decide the unit volume size which gives better chances to deduce various mixture models.

3 Unit Volume Based Distributed Clustering

3.1 Local Clustering

If we let random variable $\mathbf{X} = (X_1, X_2, ..., X_n)$ represent features of the instance, a probability density function indicates a local cluster from the local clustering results. That is, instances in the same local cluster are represented by the mean and the covariance values. Then the clustering results (i.e. mean and covariance value of each local cluster) are sent to the global site to be used in global clustering. Therefore, different clustering algorithms are able to run in each local site. The overall conceptual diagram is shown in Fig 1.

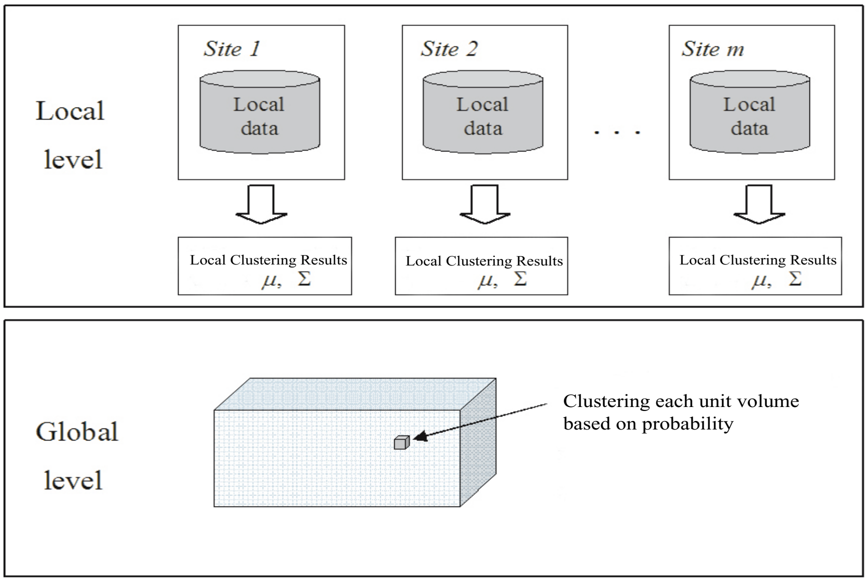

Local level
Site 1
Local data
Site 2
Local data
. . .
Site m
Local data
Local Clustering Results μ, Σ
Local Clustering Results μ, Σ
Local Clustering Results μ, Σ
Global level
Clustering each unit volume based on probability

Fig. 1. Conceptual diagram of unit volume based distributed clustering

3.2 Global Clustering

In global clustering, distribution of instances are presumed by the results of local clustering. First, data space is divided into unit volume $V = h^n$ equally where $n \in \Re$ is the number of features and h is the given unit length. Next, $center_V$, which is the central point of each unit volume, is used to decide which cluster the unit volume is most likely to be assigned to, based on the probability density function. The probability density function is calculated by the mean and the variance calculated by each local clustering. Therefore each unit volume is assigned to clusters depending on the results of,

$$\begin{aligned} Cluster(center_V) &= argmax_{c_i \in C} \int_V f(X = center_V) \\ &= argmax_{c_i \in C}\ volume_V \\ &\times \frac{1}{\sigma\sqrt{2\pi}} \exp[-\frac{1}{2}(center_V - \mu_{c_i})^t \Sigma_{c_i}^{-1}(center_V - \mu_{c_i})] \end{aligned} \tag{1}$$

where c_i is the i_{th} cluster, C is the set of clusters, $volume_V$ is the volume of one unit, μ_{c_i}, Σ_{c_i}, σ are mean, variance, standard deviation values of c_i, respectively.

3.3 Merging

The merging process is to reflect the similarity between local clusters. Two methods, mean based merging and unit volume based merging, have been introduced. We present the following notations to facilitate the description on merging clusters:

- $C^{Global} = \{c_1, c_2, ..., c_i\}$ is a set of global clusters.
- $V_{total} = \{V_1, V_2, ..., V_N\}$ is a set of unit volumes in the data space.
- μ_{c_i} is the mean point of cluster c_i.
- $|D_i|$ is number of instances in cluster c_i.
- $p_{c_i}(x)$ is the approximated integration of probability density function f_{c_i} with unit volume where central point is x.
- p_{merge} is the probability of sampling the sample point from mixture model of two merged clusters.
- p_{split} is the probability of sampling the sample point from two separate clusters.
- $mixtureModel(f_{c_i}, f_{c_j})$ is a mixture of two normally distributed models, f_{c_i} and f_{c_j}, with different weights [7].

If $p_{merge} > p_{split}$ then the two clusters are merged and $mixtureModel(f_{c_i}, f_{c_j})$ is calculated with weights given by ratio of $|D_i|$. The overall algorithm is:

```
Function $mergingGlobalClusters$ $(C^{Global}, V_{total}, mergeType)$
    if ($mergeType$==meanBased) then
      for $i = 1$ to $|C^{Global}|$ do
          $\mu_{c_i} = E[center_V]\ \forall center_V \in c_i$
      endfor

    for all adjacent $c_i, c_j \in C^{global}$ do
      $w_i = \frac{|D_i|}{|D_i|+|D_j|}, w_j = \frac{|D_j|}{|D_i|+|D_j|}$
      calculate $p_{merge}$, $p_{split}$
```

```
        if $p_{merge} > p_{split}$ then
          $f_{c_i} = mixtureModel(f_{c_i}, f_{c_j})$
          $c_i \leftarrow \{c_i \bigcup c_j\}$ /* merging */
        endif
    endfor
**Endfunction**
```

Mean Based Merging. Mean points in each cluster are considered for merging. Therefore,

$$p_{merge} = w_i p_{c_i}(\mu_{c_i}) \times w_j p_{c_j}(\mu_{c_j}) \tag{2}$$

$$p_{split} = p_{c_i}(\mu_{c_i}) \times p_{c_j}(\mu_{c_j}) \tag{3}$$

Unit Volume Based Merging. Every central point of the unit volume in the cluster is considered for merging. Therefore,

$$p_{merge} = \prod_{center_V \in c_i, c_j} [w_i p_{c_i}(center_V) + w_j p_{c_j}(center_V)] \tag{4}$$

$$p_{split} = [\prod_{center_V \in c_i} p_{c_i}(center_V)] \times [\prod_{center_V \in c_j} p_{c_j}(center_V)] \tag{5}$$

4 Experiments

4.1 Description of Datasets

Five different datasets were tested in the experiment. A real-world dataset, Iris, was from the UCI Machine Learning Repository [1]. The other four datasets, 2D3C3S, 2D3C3Sbiased, 3D3C3S, 3D3C3Sbiased (D: number of attributes, C: number of class, S: number of distributed sites, biased: dataset that has specific class biased in a certain site) were artificially generated. For example, 2D3C3S is generated by selecting 3000 instances randomly out of three different type of normally distributed models each corresponding to three different type of classes, while 3D3C3S is generated by selecting 1800 instances randomly from three different type of normally distributed models. Table 1 shows the number of features, classes, clusters and instances of the datasets used in our experiments.

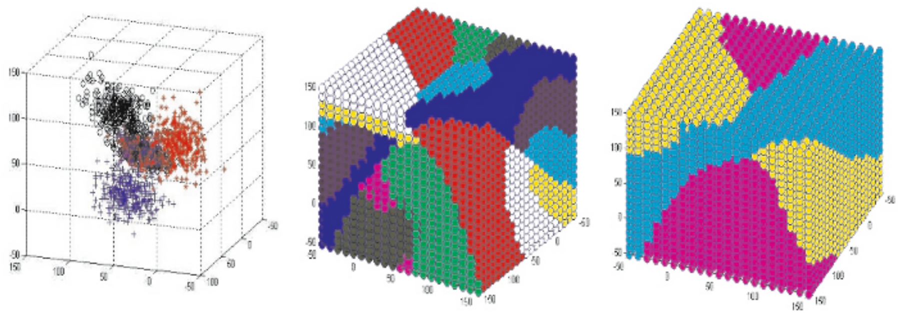
4.2 Experimental Results

In this experiment, k-means clustering has been adopted for local clustering and the classification accuracy was used to measure the quality of clustering. The accuracy was calculated by comparing instances in resulting clusters with actual classes. In other words, accuracy is the hit ratio of the instances that matches with the real class which is specified in formula (6).

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html

Table 1. Number of features, classes, clusters and instances of each dataset

Dataset	Feature	Class	Cluster	Instance
Iris	4	3	3	150
2D3C3S	2	3	3	3000
2D3C3S(biased)	2	3	3	300
3D3C3S	3	3	3	1800
3D3C3S(biased)	3	3	3	900



(a) True Model (b) Global Clustered Model (c) Merged Model

Fig. 2. Representation of 3D3C3S dataset

$$accuracy_{Cluster} = \frac{number\ of\ hit\ instance}{number\ of\ total\ instance} \tag{6}$$

In Fig.2 we can see the clustered models with 3D3C3S data. Fig.2(a) is the true model of 3D3C3S data, representing each data point in the three dimensional feature space. Fig.2(b) shows the results of global clustering. Fig.2(c) is the final model after merging. By comparing Fig.2(a) with Fig.2(c) we can see that our clustered model approximates the true model accurately.

Table 2 shows the accuracies for the five datasets compared among the four different algorithms. **Global K-means** indicates experiments performed with

Table 2. Comparison of different clustering algorithms using global and local k-means algorithm and unit volume based distributed clustering with mixture models

Dataset	Global K-means	Local K-means(avg)	MeanBased GC	VolumeBased GC
Iris	86.2%	84.6%	73.5%	79.3%
2D3C3S	84.2%	83.1%	76.8%	78.2%
2D3C3S(biased)	81.3%	52.6%	62.4%	69.3%
3D3C3S	82.1%	79.8%	71.4%	70.6%
3D3C3S(biased)	77.0%	49.5%	59.7%	60.4%

all instances running k-means algorithm 10 times. **Local K-means(avg)** indicates average accuracy calculated by doing k-means on each local site. **Mean-Based GC** is our unit volume algorithm performed by merging clusters with mean points while **Volume Based GC** indicates merging considering all unit volume's center points. From the results, **Global K-means** showed the highest accuracies because clustering was performed on the whole dataset. Note that **Local K-means(avg)** produced better results than both **MeanBased GC** and **VolumeBased GC** for unbiased data, but the latter algorithms outperformed the former algorithm for biased data. This is because **Local K-means** simply computes the averaged results regardless of the distribution of instances, while **MeanBased GC** and **VolumeBased GC** generate appropriate clusters considering all the instances at every site. Additionally as **MeanBased GC** and **VolumeBased GC** scarcely have significant difference on accuracy. **Mean-Based GC** is recommended when dealing with large datasets due to the low computational overhead.

From this experiment we can say that the unit volume distributed clustering performed well, considering real world data being biased in distributed environments, and gives the merit that distributed data can be clustered without being directly accessed or physically moved from one site to another with high accuracy. Among the unit volume distributed clustering algorithms, **MeanBased GC** turned out to be our gold standard algorithm due to low computational cost and high accuracy.

5 Summary and Discussion

Throughout this paper we have proposed a method that clusters global data probabilistically based on the unit volume without physically moving or directly accessing distributed data. Likewise, similar clusters are merged considering the mean points or probability of unit volume's central points in mixture models at global clustering stage. The method introduced in this paper proved to show better performance when distributed data is impossible to reach directly and instance classes are biased at certain sites, in particular.

Some of future research directions include: First, setting the definition of the unit volume to describe clusters more naturally; Second, further experiments with various types of distributed data; Third, using other measures of cluster similarity such as the distance between cluster centers; Finally, improving our global clustering method to overcome its limited capability and to handle data in high dimensional space.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley and Sons Inc. (2000)
2. Januzaj, E., Kriegel, H.P., Pfeifle, M.: Towards effective and efficient distributed clustering. In: International Workshop on Clustering Large Data Sets (ICDM). (2003)

3. Vrahatis, M.N., Boutsinas, B., Alevizos, P., Pavlides, G.: The new k-windows algorithm for improving the k-means clustering algorithm. Journal of Complexity **18** (2002) 375–391
4. Tasoulis, D.K., Vrahatis, M.N.: Unsupervised distributed clustering. In: The IASTED International Conference on Parallel and Distributed Computing and Networks, as part of the Twenty-Second IASTED International Multi-Conference on Applied Informatics, Innsbruck, Austria (2004)
5. Merugu, S., Ghosh, J.: Privacy-preserving distributed clustering using generative models. In: The Third IEEE International Conference on Data Mining (ICDM'03). (2003)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In Simoudis, E., Han, J., Fayyad, U., eds.: Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, AAAI Press (1996) 226–231
7. Trivedi, K.S.: Probability and statistics with reliability, queuing and computer science applications. John Wiley and Sons Inc. (2002)