www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issues 8 Aug 2016, Page No. 17493-17498

Healthcare Data Clustering with Summary Analysis under Multiple Databases

[1] Dr. C. Kalaiselvi, Ph.D., [2] Ms.Ramya.D

Associate professor and Head,
Dept of Computer Applications,
Tiruppur Kumaran College for Women, Tirupur, Tamilnadu
Research scholar (M.Phil),
Dept of Computer Science,
Tiruppur Kumaran College for Women, Tirupur, Tamilnadu

ABSTRACT

Clustering methods are applied to group the relevant records. Partition based and hierarchy based clustering methods are adapted in the clustering process. Tree based data values and transaction data values are grouped using the clustering process. Transaction similarity is estimated using the distance measures. Data and their geometrical structures are used in the grouping process.

Peer-to-Peer network environment supports multiple database access under the distributed manner. Computational load and communication complexity parameters are considered in the distributed database building process. Distributed data partitioning operations are carried out using the General Decentralized Clustering (GDCluster) mechanism. Data values are formed as summarized views and applied in the clustering tasks. Partition and density based clustering operations are carried out on the summarized views. The GD clustering technique handles the dynamic data values. Weighted K Means clustering algorithm is adapted to perform the distributed data clustering process on healthcare data values.

The General Decentralized (GDCluster) clustering technique is enhanced to support partition and hierarchical data values. Summary analysis model is optimized to handle the hierarchical and grid based data items. The similarity estimation tasks are performed with the priority features. Data update period is also considered in the clustering process. The performance analysis shows that Enhanced GD Clustering scheme reduces the communication delay with high accuracy levels.

Keywords: Distributed Clustering, Hierarchical Data Portioning, Similarity Measures, Summarized Views, Weighted K-Means Clustering

1. Introduction

Identical objects are grouped in the clustering process. Clustering is a main task of explorative data mining and a familiar procedure for statistical data analysis in many areas, including machine learning, image analysis, bioinformatics and information retrieval.

The concept of a cluster varies between algorithms and is one of the many opinions to take when choosing the suitable algorithm for a particular problem [3]. At first the terminology of a cluster seems distinct: a group of data objects. The clusters found by distinct algorithms vary significantly in their properties and grasping these cluster models is key to understand the differences between the various algorithms. General cluster models include they are Connectivity models, Centroid models, Density models, Distribution models, Subspace models and Group models.

A clustering is basically a set of clusters, usually containing all objects in the data set. It specifies the correlation of the clusters to each other, for eg, a hierarchy of clusters fixed in each other. Clustering can be roughly distinguished in they are soft clustering, strict partitioning, hard clustering, subspace clustering, hierarchical clustering and overlapping clustering. Cluster analysis itself is not one specific algorithm; it is the common task to be solved. It can be attained by various algorithms that differ significantly in their notion of what composing a cluster and how to systematically find them. Popular

approach of clusters include groups with low distances among the cluster members, compact areas of the data space, intervals or distinct statistical distributions. The suitable parameter settings and clustering algorithm determined on the individual data set and intended use of the results. Cluster analysis as such is not an instinctive task, but an monotonous process of knowledge discovery that involves try and failure. It will often be mandatory to modify preprocessing and parameters until the result accomplish the desired properties.

Beyond the term clustering, there are a number of terms with identical meanings, including automated classification, numerical taxonomy, typological and botryology analysis. The subtle changes are often in the usage of the results: while in data mining, the developing groups are the matter of activity, in automatic classification primarily their selective power is of interest [5]. This often leads to misconception of researchers coming from the fields of machine learning and data mining, since they use the similar terms and often the similar algorithms, but have different goals.

2. Related Work

Massive research efforts have been devoted to consensus clustering. These studies can be roughly divided into two divisions: CC with implicit objectives (CCIO) and CC with explicit objectives (CCEO). The methods in CCIO do not set any global objective functions for CC. They employ heuristics to find approximate solutions. Although an objective function was described on the normalized mutual information measure, the proposed algorithms actually do not address this optimization problem precisely [9]. Following this idea [1] built different types of graphs to improve the clustering quality. Another class of results in CCIO is based on the similarity matrix. For instance, summarized the information of basic partitioning into a co-association matrix, based on agglomerative hierarchical clustering it was used to find the final clustering. Some work along this line has been suggested subsequently, with the target either on developing hierarchical clustering or on building more informative co-association matrix [2]. Other CCIO methods include Relabeling and Voting [10], Locally Adaptive Cluster based methods [7], fuzzy clustering based methods [8], genetic algorithm based methods and still many more.

The methods in CCEO have precise global objective functions for consensus clustering. For instance, to find the Median Partition based on Mirkin distance proposed three simple heuristics. The comparative studies on some of these heuristics can be found. This elegant idea could be traced back to the work by Mirkin. They further extended their work to using the adopting maximization algorithm with a finite assertion of multinomial distributions for consensus clustering. In addition to this, there are some other interesting objective behavior for consensus clustering, such as the ones which can be cleared up by nonnegative matrix factorization, kernel based methods [10], simulated annealing and genetic algorithms respectively.

There are still many other algorithms for consensus clustering. Readers with interests can refer to some survey papers. While comparing with CCIO methods, CCEO methods might offer better interpretability and greater robustness to clustering results, through the guidance of the objective functions. It is very hard for CCEO methods to make a balance between the high execution efficiency and high clustering quality [4]. Each CCEO method typically works for one objective function, which seriously delimit the relevancy to real life applications in unique domains. These indeed inspire our research in this paper, which attempts to build a familiar theoretic framework for K-means-based consensus clustering using many utility functions.

3. K-means Clustering Technique

In data mining and statistics, K-means clustering is a method of cluster analysis which intend to partition n observations into k clusters in which each examination belongs to the cluster with the nearest mean. This results into a dividing of the data space into Voronoi cells. The dispute is computationally difficult (NP-hard), there are efficient heuristic algorithms that are generally employed that combine fast to a local optimum. These are usually identical to the expectation-maximization algorithm for combination of Gaussian distributions via an iterative refinement approach employed by both the algorithms [6]. They both use cluster centers to model the data, K-means clustering influence to find clusters of comparable spatial extent, although the expectation-maximization mechanism allows clusters to have unique shapes. Given a set of observations (x1, x2, …, xn), in which each examination is a d-dimensional real vector, K-means clustering objective is to partition the n observations into k set ($k \leq n$) S = {S1, S2, …, Sk} so as to decrease the within-cluster sum of squares (WCSS):

$$\arg \min_{s} \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2$$

Where μi is the mean of points in Si.

A variety of heuristic algorithms are generally used. The K-means algorithm discussed underneath has polynomial smoothed running time. It is shown that for capricious set of n points in [0,1]d, if each point is freely perturbed by a common distribution with mean 0 and variance σ2, then the expected running time of K-means algorithm is bounded by O(n34k34d8log4(n) / σ6), which is a polynomial in n k,d and 1 / σ. Better bounds are tested for simple cases.

For example, demonstrated that the running time of K-means algorithm is bounded by O (dn4M2) for n points in an integer lattice {1,…..,M}2 . The most common algorithm uses a constant refinement technique. Due to its ubiquity it is usually called the K-means algorithm; it is also referred to as Lloyd's algorithm, especially in the computer science association. Given an basic set of k means m1(1),…,mk(1) the algorithm proceeds by alternating among two steps:

Assignment step: Assign each examination to the cluster with the closest mean.

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\| \leq \left\| x_p - m_j^{(t)} \right\| \forall 1 \leq j \leq k \right\}$$

Where each xp goes into absolutely only one, even if it could go in two of them. Update step: Calculate the new means to be the centroid of the research in the cluster.

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$

The algorithm is expected to have converged when the assignments no longer change. Generally used initialization methods are Random and Forgy Partition. The Random Partition method first randomly allocate a cluster to each observation and then returns to the Update step, thus computing the initial means to be the centroid of the cluster's randomly allotted points. The Forgy method randomly chooses k observations from the data set and uses these as the original means. The Forgy method aims to spread the basic means out, while Random Partition keeps all of them close to the center of the data set. Conforming to Hamerly et al., the Random Partition method is generally preferable.

As it is a heuristic algorithm, there is no assurance that it will converge to the global optimum and the result may rely on the initial clusters. As the algorithm is generally very fast, it is common to run it many times with different starting conditions. In the worst case, K-means can be very slow to converge: in particular it has been shown that there exist defined point sets, even in 2 dimensions, on which K-means takes exponential time, that is, to converge. These point sets do not look to arise in practice: this is authenticated by the fact that the continuous running time of K-means algorithm is polynomial. The "assignment" step is also known as expectation step, the "update step" as maximization step, creating this algorithm an alternative of the generalized expectation-maximization algorithm. The following list presents the variation of K-means algorithm.

- Fuzzy C-Means Clustering is a soft form of K-means, where each data point has a fuzzy degree of belonging to all cluster.
- Gaussian mixture models qualified with EM algorithm (Expectation-Maximization algorithm) RETAINS probabilistic assignments to clusters, rather than deterministic assignments and multivariate Gaussian distributions rather than means.
- Several methods have been suggested to choose preferred starting clusters. One recent project is K-means++.

The filtering algorithm uses kd-trees to speed up each K-means step. Some methods attempt to speed up each K-means step using support or the triangle inequality.

- Escape local optima by interchanging points among clusters.
- The Spherical K-means clustering algorithm is suitable for directional data.

4. Data Clustering under Distributed Environment

In totally distributed clustering algorithms, the data set as a whole remains dispersed and the participating distributed processes will slowly discover various clusters. Communication complexity and overhead, accuracy (AC) of the derived model and data privacy are amidst the concerns of DDM. Typical applications involving distributed clustering include: clustering distinct media metadata from different machines; clustering nodes' activity history data; clustering books in a distributed network of libraries; clustering scientific achievements from different institutions and publishers. A familiar approach in distributed clustering is to combine and join local representations in aggregate local models or central node in a hierarchical structure. Some new proposals, although being completely decentralized, include synchronization at the end of each round and/or need nodes to maintain history of the clustering.

A typical distributed clustering algorithm (GDCluster) is proposed and started with two popular clustering methods, density-based and partition-based clustering methods. We first introduce a basic method in which nodes constantly build a summarized view of the data set by constantly exchanging information on data items and data representatives applying gossip-based communication. Gossiping is used as a simple, vigorous and efficient dissemination technique, which assumes no predefined in the network. The summarized view is a basis for finishing weighted versions of the clustering algorithms to produce final clustering results.

GDCluster can cluster a data set which is separated among a huge number of nodes in a distributed environment. It can hold two classes of clustering, i.e. partition-based and density-based, while being fully distributed, asynchronous and also flexible to churn. The general design principles operated in the proposed algorithm also allow customization for other classification of clustering, which are skipped out of the current paper. We also discuss improvements to the algorithm especially aimed at improving communication costs.

Huge volume of data values are distributed between multiple systems beneath the Peer to Peer environment. Processing, transmission and storage. cost factors are the key issue in the shared data process. Normal Decentralized (GDCluster) algorithm is adopted to perform clustering on dynamic and shared data sets. Summarized view of the data sets are also used in the clustering process. GDCluster is customized for the completion of the partition-based and density-based clustering models on the summarized views. The clustering model is adjust to adapt the dynamic data values. The GDCluster model helps partition based clustering and density based clustering tasks. Weighted K-means algorithm is adjust to perform partition based clustering under distributed environment. The following issues are identified from the existing system. GD Clustering method is not customized for all cluster types. Summarized view construction is not optimized. Limited cluster accuracy levels are achieved in the system. Communication and computational complexity is high.

5. Summary Analysis based Data Clustering in Multiple Databases

The General Decentralized (GDCluster) scheme is enhanced to support hierarchical and grid based clustering methods. Summarized view construction is tuned for hierarchical and grid data models. Priority factors are adapted for the relationship identification process. Age based data and representative elimination process is integrated with the system.

The General Decentralized (GDCluster) scheme is designed to support hierarchical and grid based clustering process. Weight estimation process is enhanced with priority values. The summarized view is constructed with hierarchical properties. The system is split into four major modules. They are Data Preprocess, Summarized View Construction; Partition based Clustering Process and Hierarchical Clustering Process. The data preprocess module is designed to perform data cleaning process. Summarized view construction process is designed to group up similar data values. Partition based clustering process is designed to perform clustering with weight values. Hierarchical and grid based clusters are constructed with hierarchical relationships.

Data cleaning is performed in the data preprocess. Data values are parsed and updated into the database. Redundant data values are removed from the data set. Missing values are assigned with suitable values. Summarized views are constructed by continuously reciprocate information on data items and data representatives. Data transmission is carried out using gossip-based communication. Gossiping is used as a simple, robust and efficient dissemination technique. Summarized views are used in weighted clustering process.

Partition based clustering process is carried out under the distributed environment. Clustering process is performed using General Decentralized Clustering (GDCluster) scheme. Decentralized asynchronous communication is carried out under the system. Weighted K means clustering algorithm is used in the system. The GDCluster scheme is improved with hierarchical and grid clustering support. Summarized view construction is improved to handle hierarchical and grid based data values. Data representatives are organized in hierarchical manner. Statistical operations are called on approximated grid cells.

6. Performance Analysis

The distributed data clustering scheme is designed to perform the data partitioning under distributed databases. The data values are collected from various sources. The General Decentralized Clusters (GDC) and General Decentralized Hierarchical Clusters (GDHC) scheme are analyzed in the system. Heart patient data values are used in the partitioning process. The system is tested with two performance measures. They are purity and communication complexity values. The purity measures are used to estimate the cluster quality factors. Data transmission delay is analyzed using the communication complexity measure.

6.1. Communication Complexity

The medical data clustering process is carried out under the distributed environment. The data values are provided from different data providers. The clustering process is carried out after performing the data transfer process. The communication complexity measure is calculated to verify the communication time levels. Figure 6.1. and table 6.1. Shows the communication complexity level analysis between the General Decentralized Hierarchical Clusters (GDHC) and General Decentralized Clusters (GDC) scheme. The analysis result shows that the (GDHC) scheme reduces the communication delay 40% than the General Decentralized Clusters (GDC) scheme.

6.2. Purity

The purity of a cluster represents the fraction of the cluster corresponding to the largest class of documents assigned to that cluster; thus, the purity of the cluster j is defined as

$$Purity\ (j) = \frac{1}{n_j} \max_i (n_{ij}) \qquad (5)$$

The overall purity of the clustering result is a weighted sum of the purity values of the clusters as follows:

$$Purity = \sum_j \frac{n_j}{n} Purity\ (j) \qquad (6)$$

DOI: 10.18535/ijecs/v5i8.11

Table No: 6.1.Communication Complexity Analysis between General Decentralized Clusters (GDC) and General Decentralized Hierarchical Clusters (GDHC)

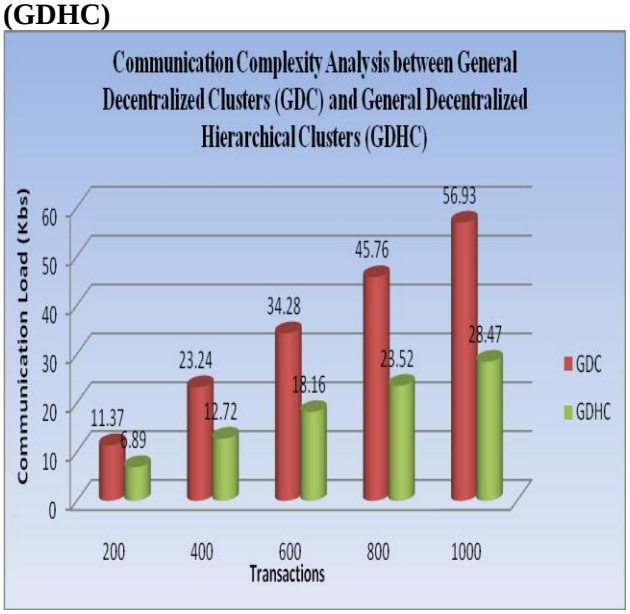
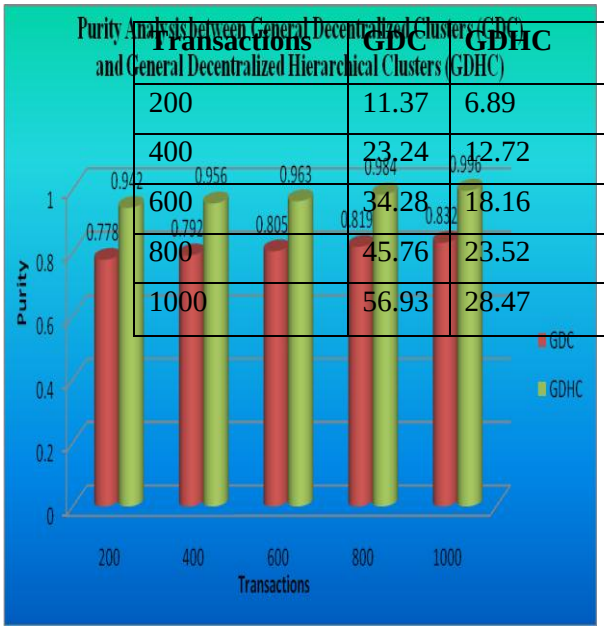


Figure.6.1. Communication Complexity Analysis between General Decentralized Clusters (GDC) and General Decentralized Hierarchical Clusters (GDHC)

In general, the larger the purity value is, the better the clustering result is (6). The purity measure is also used to evaluate the cluster accuracy levels. The purity analysis is shown in figure 6.2. and table 6.2. The analysis shows that the General Decentralized Hierarchical Clusters (GDHC) scheme increases the accuracy level 10% than the General Decentralized Clusters (GDC) scheme.

Transactions	HPKMC	SNC
200	0.778	0.942
400	0.792	0.956
600	0.805	0.963
800	0.819	0.984
1000	0.832	0.996

TABLE No: 6.2. Purity analysis between General Decentralized Clusters (GDC) and General Decentralized Hierarchical Clusters (GDHC)



Transactions	GDC	GDHC
200	11.37	6.89
400	23.24	12.72
600	34.28	18.16
800	45.76	23.52
1000	56.93	28.47

Figure No: 6.2. Purity analysis between General Decentralized Clusters (GDC) and General Decentralized Hierarchical Clusters (GDHC)

7. Conclusion and Future Work

Clustering techniques are applied to partition the similar data values. General Decentralized Cluster (GDCluster) scheme is also used for the distributed data clustering process. Partition based clustering and density based clustering operations are helped by the GDCluster scheme. GDCluster scheme is enhanced to support hierarchical and grid based clustering process. Hierarchical and grid based clustering operations are carried out on dynamic and distributed data sets. Summarization views constructed with hierarchical data relationships. Transmission and computational overhead is reduced by the system. High scalability is supported by the system. The distributed data clustering methods can be enhanced with privacy preservation techniques. Data clustering process can be improved to support natural language process. The system can also improved to support clustering on data streams.

References

[1] Abdala, D. Duarte, P. Wattuya and X. Jiang, "Ensemble clustering via random walker consensus strategy," in Proc. 20th Int. Conf. Pattern Recog., 2010, pp. 1433–1436.

[2] X. Wang, C. Yang and J. Zhou, "Clustering aggregation by probability accumulation," Pattern Recog., vol.42, no.5, pp. 668–675, 2009.

[3] Graph Archive Dataset. [Online]. Available: http://staffweb.cms.gre.ac.uk/wc06/partition/, 2014.

[4] Junjie Wu, Hongfu Liu, Hui Xiong, Jie Cao and Jian Chen, "K-Means-Based Consensus Clustering: A Unified View", IEEE Transactions On Knowledge And Data Engineering, Vol. 27, No. 1, January 2015

[5] N. Xu, L. Chen and B. Cui, "Loggp: A Log Based Dynamic Graph Partitioning Method," in Proc. VLDB, 2014.
[6] Ning Xu, Bin Cui, Lei Chen, Zi Huang and Yingxia Shao, "Heterogeneous Environment Aware Streaming Graph Partitioning", IEEE Transactions On Knowledge And Data Engineering, Vol. 27, No. 6, June 2015.
[7] C. Domeniconi and M. Al Razgan, "Weighted cluster ensembles: Methods and analysis," ACM Trans. Knowl. Discovery Data, vol. 2, no. 4, pp. 17:1–17:40, 2009.
[8] K. Punera and J. Ghosh, "Consensus-based ensembles of soft clusterings," Appl. Artif. Intell., vol. 22, no. 7-8, pp. 780–810, 2008.
[9] S. Vega-Pons and J. Ruiz-shulcloper, "A survey of clustering ensemble algorithms," Int. J. Pattern Recogn. Artif. Intell., 2011.
[10] S. Vega-Pons, J. Correa-Morris and J. Ruiz-Shulcloper, "Weighted partition consensus via kernels," Pattern Recog., vol. 43, no. 8, pp. 2712–2724, 2010.