International Journal On Engineering Technology and Sciences – IJETS™
ISSN(P): 2349-3968, ISSN (O): 2349-3976
Volume III, Issue V, May- 2016

GD CLUSTER: A GENERAL DECENTRALIZED CLUSTERING ALGORITHM

Sreejith P S
ME (CSE)
Jay Shriram Group of Institutions Tirupur
sri4jith@gmail.com

Mr. T Sreenivasan
Assistant Professor
Jay Shriram Group of Institutions Tirupur
ktsreenivasan@gmail.com

Dr.S.Rajalakshmi
Associate Professor/ CSE
Jay Shriram Group of Institutions Tirupur
mrajislm@gmail.com

ABSTRACT

Data sets measuring gigabytes and even terabytes are now quite common in data and text mining, where a few million data points are the norm.. When a sequential data mining algorithm cannot be further optimized or when even the fastest available serial machine cannot deliver results in reasonable time, it is natural to look to parallel computing. Clustering algorithms are often useful in applications in a various fields such as visualization, pattern recognition, learning theory, computer graphics, neural networks, AI, and statistics. Practical applications of clustering include unsupervised classification and taxonomy generation, nearest neighbor searching, time series analysis, text analysis and navigation. Data clustering is a frequently used and is a useful tool in the data mining. There are a variety of data clustering algorithms, which are generally classified, into two categories. hierarchical algorithms and partition algorithms. Nevertheless, the processes participating in decentralized clustering algorithm will gradually discover various clusters and can provide this information to third parties, if required. Distributed Data Mining (DDM) explores methods of applying data mining algorithms decentralized data, utilizing distributed resources of computation and communication. Classically, data mining algorithms attempt to the optimize storage and processing costs, whilst additional requirements arise in DDM, such as maintaining the scalability, low communication overhead, and privacy.

Keywords
Clustering, Peer to Peer, Data Mining, Data Sets.

1. INTRODUCTION

Peer-to-Peer systems are distributed systems without centralized control in which each node shares and exchanges data across a network. Each node is connected directly to a number of nodes. In order for a node to communicate with any other node in the network, it has to do so through one of its peers. Data Clustering is one of the major data mining problems. One of the most commonly used clustering algorithms is the K-Means algorithm. The goal of this algorithm is to partition dataset into separate groups (clusters), each group is represented by its centroid. The portioning is based on minimization of the sum of squared Euclidean distances between patterns and their corresponding cluster centers.

An important distributed data mining problem which has been investigated recently is the distributed data clustering problem. The goal of data clustering is extract new potential useful knowledge from a generally large data set by grouping together similar data items and by separating dissimilar ones according to some defined dissimilarity measure among the data items them. In a distributed environment, this goal must be achieved when data cannot be concentrated on a single machine, for instance because privacy concerns or due to network bandwidth limitations, or because of the huge amount of distributed data. In data management applications, deployed peer-to-peer systems have proven to be able to manage very large databases made up by thousands of personal computers. Many proposals in the literature have significantly improved the existing P2P systems in several aspects, such as searching performance, query expressivity; multi-dimensional distributed indexing the ensuing solutions can be

effectively employed in the forthcoming new distributed database systems to be used in large grid computing networks and in clustering database management systems.

In light of the foregoing, it is natural to the foresee evolution of P2P networks towards supporting distributed data mining services, by which many peers spontaneously negotiate and cooperative perform a distributed data mining task. In particular, the data clustering task matches well the features of P2P networks, since clustering models exploit local information, and consequently clustering algorithms can be effective in handling topological changes the data updates. Current distributed data clustering algorithms cannot be directly applied to data stored in P2P networks because they expect data to be organized according to traditional distributed database management systems where distribution of the relational schema is planned a-priori in the design phase.

2. RELATED WORKS

Clustering distributed data has been addressed in many publications over the past decade. In the authors presented the collective Principle Component Analysis as a new method for clustering distributed high dimensional data. In [9], a hierarchical clustering algorithm (HP2PC) was proposed for distributed data over a multilayer overlay network of Peer neighbors. In the authors suggested approximating the distributed high dimensional data as precisely as possible with a specified number of bytes before sending them to a centralized server to run the clustering algorithm. Jin et al. presented a new algorithm, called Fast and Exact K-means Clustering (FEKM), which typically requires only one or a small number of passes on the entire dataset, and provably produces the same cluster centers as reported by the original k-means algorithm. The algorithm uses sampling to create initial cluster centers, and then takes one or more passes over the entire dataset to adjust these cluster centers. Januzaj et at. Proposed clustering the data locally and extracting suitable representatives out of these clusters. These representatives are sent to a global server where the complete clustering based on the local representatives is restored. This approach is characterized by carrying out local clustering quickly and independently from each other. Furthermore, algorithm requires low transmission cost, as the number of transmitted representatives is much smaller than the cardinality of the complete data set.

One of the most recent works on the distributed data clustering is the work presented. The authors presented an elegant synchronization technique for clustering distributed data via the K-Means algorithm. The basic idea behind this algorithm is that each node runs a single K-Means iteration over its local data then, the resulting centroids are used to synchronize the clustering results with the neighbor nodes. Each node sends its centroids to its neighbors. Receiving the centroids of a certain cluster obtained at all the neighboring nodes, each node modifies its centroid for that cluster to be the weighted average of the received centroids and its current local centroid. Afterwards, each node starts the next iteration using the obtained average centroids till satisfying the stopping criteria. Also, the authors described how the algorithm should behave in dynamic networks when the network structure or the data may change. The evaluation results showed that this algorithm achieves high accuracy compared to the classical centralized K-Means. Also the communication efficiency of the algorithm is demonstrated.

Distributed Clustering is carried out on two different levels, i.e. the local level and the global level (Figure 1). On the local level, all sites analyze the data independently from each other resulting in a local model which should reflect an optimum trade-off between complexity and accuracy. Our proposed local models consist of a set of representatives. Each representative is a concrete object from the objects located at the local site. Furthermore, we augment each representative r with a suitable covering radius indicating the area represented by r. Thus, r is a good approximation for all objects residing on the corresponding local sites and is contained in

the covering area of r. Next, local model is transferred to a central site, where the local models are merged in order

to form the global model. The global model is created by analyzing the local representatives. This analysis is similar to new clustering of the representatives with suitable global clustering parameters. To each local representative a global cluster identifier is assigned. The resulting of global clustering is sent to all local sites.

If a local object is located in the covering area of global representative, the cluster-identifier from this representative is assigned to the local object. we can achieve that each site has the same information as if their data were clustered on a global site, together with the data of all other sites.

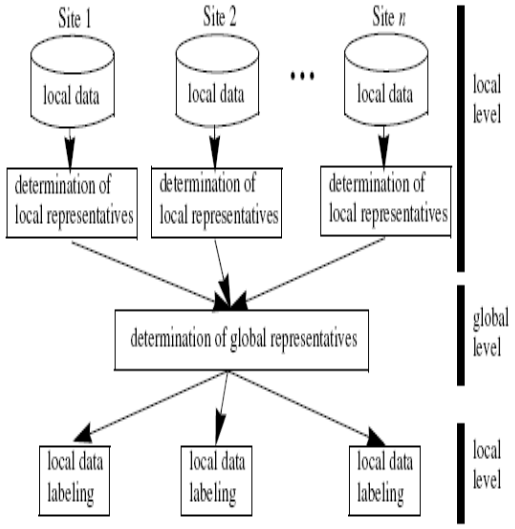

Fig. 1. Distributed clustering.

3. CHALLENGES AND CONTRIBUTIONS

Over the last decade, we have seen revolution in connectivity between computers, and a resulting paradigm shift from centralized computation to highly distributed system. For example, large-scale peer-to-peer (P2P) networks with millions of servers being used and designed for distributed information storage and retrieval, and advances in hardware are leading to the augmentation of our physical environment with sensor networks consisting of hundreds of thousands of small sensor nodes. Applications for such as large-scale distributed systems have three salient properties that distinguish them from traditional centralized or small-scale distributed systems. the dynamics of large-scale distributed systems are often significantly different. For example, P2P networks, individual machines are often under the control of a large number of heterogeneous users may join or leave the network at any time. Sensor networks are often involve the deployment in inhospitable or inaccessible areas are naturally under high stress. Individual sensors may fail at any time, and the wireless network that connects them is highly unreliable.

Thus, with massive distribution comes massive instability; consequently, the system as a whole must be fault-tolerant, as node and link failures or temporary communication disruptions are the norm rather than the exception. Second, due to the large number of nodes and the volatility of the system, any reliance on central coordination will limit the system scalability. *Gossip-based* protocols are emerging as an important communication paradigm. In gossip-based protocols, each node contacts one or a more nodes in each round (usually chosen at random), and exchanges information with these nodes. The dynamic information spread bear a resemblance to the spread of an epidemic, lead to high fault tolerance and “self-stabilization”. Gossip-based protocols usually do not require error recovery mechanisms, and thus enjoy a large advantage in simplicity, while often incurring only moderate overhead compared to optimal deterministic protocols, such as the construction of data dissemination trees. The guarantees obtained from gossip are usually probabilistic in nature; they achieve high stability under stress and disruptions, and scale gracefully to a huge number of nodes. In comparison, traditional techniques have absolute guarantees, but are unstable or fail to make progress during periods of even modest disruption.

Third, due to the large scale of the system, the values of aggregate functions over the data in the whole network are often more important than individual data at nodes. For example, in a sensor network with temperature sensors, we are often more interested in the average or median temperature measured by all sensors in an area rather than the single measurement at an individual sensor. In a sensor network with acoustic and vibration sensors, we may want to find out to which extent events of especially large noise or vibration are spatially or temporally correlated. In a P2P system, we may be interested in the total number of files, the average size of files stored, or quantiles about the amount of free space on the machines disks. At the same time, communication bandwidth is often a scarce resource in decentralized settings, so the computation of aggregates should involve only small messages. In particular, any protocol collecting all local data at given node will create communication bottlenecks, or a message implosion at that node.

4. METHODOLOGY OVERVIEW

4.1 Data Clustering Efficacy and Efficiency

The main goal of the experiments described in the next section is to compare the accuracy of the clusters produced by three P2P systems, their efficacy, as a function of the network costs that is their efficiency as clustering algorithms.

To determine the accuracy of clustering, we have compared the clusters generated by each P2P system as a function of the number of hops, with the ideal clustering computed by the system when routing through a large number of hops in order to include the entire network; for our experiments we have choosen 1024. In the latter case, all zones are reachable from every other zone, thus simulating a density-based algorithm operating as if all distributed data were centralized in a single machine, as far as query results are concerned. Limiting the number of hops means the computed estimate is approximation of true estimate computed by routing queries to the entire network, which therefore yields a \reference" clustering.

4.1.1 Density-Based Clustering in P2P Systems

When applying kernel-based cluster to P2P over-lay networks, some observations are in order. It is mandatory to impose a bound on the distance H in hops of the zones containing the objects that contribute to the estimate in a given zone. A full calculation of summation would require answering an unacceptable number of point queries. Note that, depending on the overlay network, the lower bound on the distance from the center of a zone to an object in a zone beyond H hops may be not greater than the radius of the zone itself. There will be a trade of between network messaging costs and clustering accuracy, and clustering results must be experimentally compared with the ideal clustering obtained when H is large enough to reach all objects.

Different peers may prefer different parameters for clustering the network's data, e.g., different values of h, kernel functions, maximum number of hops, whether to use an adaptive estimate. Therefore, a peer interested in clustering the data acts as a clustering initiator, i.e., it must take care of all the preliminary steps need to make its choices available to the network, and to gather information useful to make those choices, e.g., descriptive statistics.

4.2 Our Contributions

The way the algorithm behaves locally at each node is the major concern in the proposed algorithm. The basic idea of this algorithm is: Starting from an initial cluster assignment for all the patterns, the algorithm tries to move each pattern from its current cluster to another cluster as long as this move is going to reduce the overall objective function. The major concern here is the way the algorithm assesses the effect of the move to determine whether the objective function is going to be increased or decreased. Of course,

the assessment needs to be done efficiently without having to recalculate the overall objective function. Also, if the move decision was taken, it is required to update the centroid of both the old and the new clusters of the moved patterns effectively. In order to do that, a formula needs to be developed to assess the move effect and another formula needs to be developed to update the clusters centroids according to a move decision.

Consider the problem where we have a dataset D distributed over N nodes in a Peer-to-Peer network where each node can directly communicate with its neighbors. The required number of clusters c is initially given to the algorithm.

CONCLUSION AND FEATURE WORKS

We first discussed some application ranges which benefit from an effective and efficient distributed clustering algorithm. Due to economical, technical and security reasons, it is often not possible to transmit all data from different local sites to one central server site where the data can be analyzed by means of clustering. Therefore, we introduced an algorithm which allows the user to find an individual trade-off between clustering-quality and runtime. Our approach first analyses the data on the local sites and orders all objects o according to a quality criterion DynRepQ(o) reflecting whether the actual object is a suitable representative. Note that this quality measure depends on the already determined representatives of a local site. As we produce the local representatives in a give-me-more manner and apply a global clustering algorithm which supports efficient incremental clustering, our approach allows starting with the global clustering algorithm as soon as the first representatives are transmitted from the various local sites. Our experimental evaluation showed that the presented scalable density-based distributed clustering algorithm allows effective clustering based on relatively little information, i.e. without sacrificing efficiency and security. In this paper we have described methods to cluster data in multi-dimensional P2P networks without requiring a specific reorganization of the network and without altering or compromising the basic services of P2P systems, which are the routing mechanism, the data space partition among peers and the search capabilities.

Future Work:

The Present work we introduced GD Cluster, a general fully decentralized clustering model. In our future work, we plan to develop GDCluster can be customized for other clustering types, such as hierarchical or grid-based clustering, which better satisfy specific requirements of distributed systems.

REFERENCES

[1] K. M. Hammouda and M. S. Kamel, “Models of distributed data clustering in peer-to-peer environments,” Knowl. Inf. Syst., vol. 38, no. 2, pp. 303–329, 2014.

[2] E. Januzaj, H.-P. Kriegel, and M. Pfeifle, “Scalable density-based distributed clustering,” in Proc. 8th Eur. Conf. Principles Pract. Knowl. Discovery Databases, 2004, pp. 231–244.

[3] S. Lodi, G. Moro, and C. Sartori, “Distributed data clustering in multi-dimensional peer-to-peer networks,” in Proc. 21st Australasian Conf. Database Technol., 2010, vol. 104, pp. 171–178.

[4] S. Datta, C. R. Giannella, and H. Kargupta, “Approximate distributed k-means clustering over a peer-to-peer network,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 10, pp. 1372–1388, Oct. 2009.

[5] A. Elgohary and M. A. Ismail, “Efficient data clustering over peer to peer networks,” in Proc. 11th Int. Conf. Intell. Syst. Des. Appl.,2011, pp. 208–212.

[6] G. Di Fatta, F. Blasa, S. Cafiero, and G. Fortino, “Epidemic kmeans clustering,” in Proc. Int. Conf. Data Min. Workshops, 2011, pp. 151–158.

[7] J. Fellus, D. Picard, and P.-H. Gosselin, “Decentralized k-means using randomized

gossip protocols for clustering large datasets," in Proc. Data Min. Workshops, 2013, pp. 599–606.

[8] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in Proc. 44th Symp. Found. Comput. Sci., 2003, pp. 482–491.

[9] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, "Gossip-based peer sampling," ACM Trans. Comput. Syst., vol. 25, no. 3, article 8, Aug. 2007.

[10] A. M. Frieze and G. R. Grimmett, "The shortest-path problem for graphs with random arc-lengths," Discr. Appl. Math., vol. 10, no. 1, pp. 57–77, 1985.

[11] D. Mosk-Aoyama and D. Shah, "Computing separable functions via gossip," in Proc. 25th ACM Symp. Principles Distrib. Comput., 2006, pp. 113–122.

[12] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in Proc. 5th Berkeley Symp. Math. Statist. Probability, 1967, vol. 1, no. 14, pp. 281–297.

[13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd Int. Conf. Knowl. Discovery Data Min., 1996, pp. 226–231.