ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

A General Decentralized Clustering Using K-Harmonic Means

Shabana AS[1], Rajesh Kumar PM[2]

PG Scholar, Computer Science, LBS College of Engineering, Kasaragod, India[1]

Assistant Professor, Computer Science, LBS College of Engineering, Kasaragod, India[2]

Abstract: In peer-to-peer systems, large amounts of data are distributed among multiple sources. Analysis of this data and identifying clusters is a difficult task due to processing, storage, and transmission costs. In this paper, GD Cluster, a general fully decentralized clustering method, which has an ability of clustering dynamic and distributed data sets. Nodes continuously working through decentralized gossip-based communication to maintain summarized views of the data set. Distributed data mining focuses on the adaptation of data-mining algorithms for distributed computing environments. In this paper, we propose a GD Cluster, a general fully decentralized clustering method using K-Harmonic means algorithm, which is having the ability of clustering dynamic and distributed datasets. K-Harmonic Means is essentially insensitive to the initialization of the centers, so that its performance does not depend on the initialization of centers.

Keywords: Distributed systems, clustering, dynamic system, partition-based clustering, density-based clustering.

I. INTRODUCTION

Clustering or unsupervised learning is important for analyzing large data sets. Clustering divides data into groups (clusters) of similar objects, with the objects within one cluster are highly similar and dissimilar with the objects in other clusters. With the growth of large-scale distributed systems, huge amounts of data are increasingly arising from scattered sources. Analysis of this data, using centralized processing, is often not feasible due to communication, storage and computation overheads.

In fully distributed clustering algorithms, the data set as a whole remains dispersed, and the participating distributed processes will gradually discover various clusters. Typical applications of distributed clustering can be categorized under military and surveillance, environmental monitoring, application in Logistics and Transportation, Healthcare applications and in robotic applications, commonly used approach in distributed clustering is to merge local representations in a central node, or aggregate local models in a hierarchical structure.

II. RELATED WORK

Distributed data mining is a dynamically growing area. A discussion and comparison of several distributed centroid based partitional clustering algorithms is provided in [1]. Different from many existing distributed clustering algorithms, the proposed algorithm does not require a central site to coordinate execution rounds, and/or merge local models. Also, it avoids global message flooding. A distributed partition-based clustering algorithm for clustering documents in a peer-to-peer network is proposed by Eisenhardt et al. [2]. The algorithm requires rounds of information collection from all peers in the network.

A K-means monitoring algorithm is proposed in [3]. This algorithm executes K-means by iteratively combining data samples at a central cite, and monitoring the deviation of centroids in a distributed manner. Some distributed clustering proposals impose a special structure in the network. A hierarchical clustering method based on K-means for P2P networks is suggested in [4].Some solutions which consider pure unstructured networks, require state-aware operation of nodes, work in static settings, or are aimed at computing basic functions like average and sum.

Fellus et al. [5] propose a decentralized K-means algorithm which executes in iterations, and in each iteration nodes compute an approximation of the new centroids in a distributed manner. The major drawback of the majority of existing approaches is lack of efficient solutions for adaptability in dynamic settings, which introduces significant challenges for applying the algorithms in large-scale real-world networks. Also, majority of approaches limit nodes to finding the same number of clusters.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

III.SYSTEM MODEL

System model consists of a set P =$\{p_1, p_2, \ldots\ldots p_n\}$ of n networked nodes. Each node p stores and shares a set of data items D_p^{int} as its internal data, which may change over time. D =∪ p ∈ P D_p^{int} is the set of all data items available in the network. Each data item d is presented using an attribute (meta data) vector denoted as d_{attr} . Whenever transmission of data items is mentioned in the text, transmission of the respective attribute vector is intended.

While discovering clusters, p may also store attribute vectors of data items from other nodes. These items are referred to as the external data of p, and denoted as D_p^{ext}. The union of internal and external data items of p is referred to as D_p=D_p^{int} ∪ D_p^{ext}.

During algorithm execution, each node p gradually builds a summarized view of D, by maintaining representatives, denoted as R_p =$\{r_1^p, r_2^p, \ldots\ldots\ldots r_{kp}^p\}$.Each representative r ∈ R_p is an artificial data item, summarizing a subset D_r of D. The attribute vectors of ,r_{attr} is ideally the average of attribute vectors of data items in D_r.

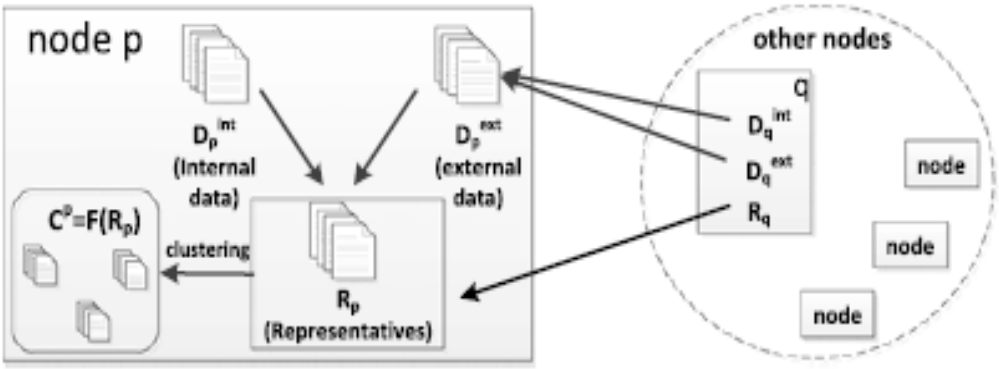

Fig. 1. A graphical view of the system model

A. Building The Summarized View

The entire data set can be summarized in each node p, by means of representatives. Each node p is responsible for deriving accurate representatives for part of the data set located near D_p^{int}. For other parts, it solely collects representatives. Accordingly, it gradually builds a global view of **D**. Each node continuously performs two tasks in parallel: i) Representative derivation, which is named as DERIVE and ii) representative collection, which is named as COLLECT. The two tasks can execute repeatedly and continuously in parallel. An outline of the tasks performed by each node is demonstrated in Fig. 2.

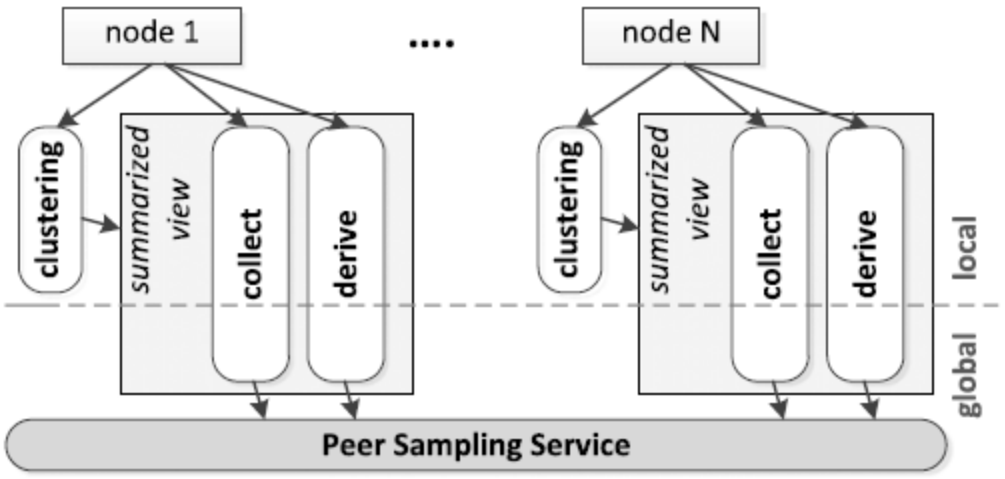

Fig. 2. The overall view of the algorithm tasks

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

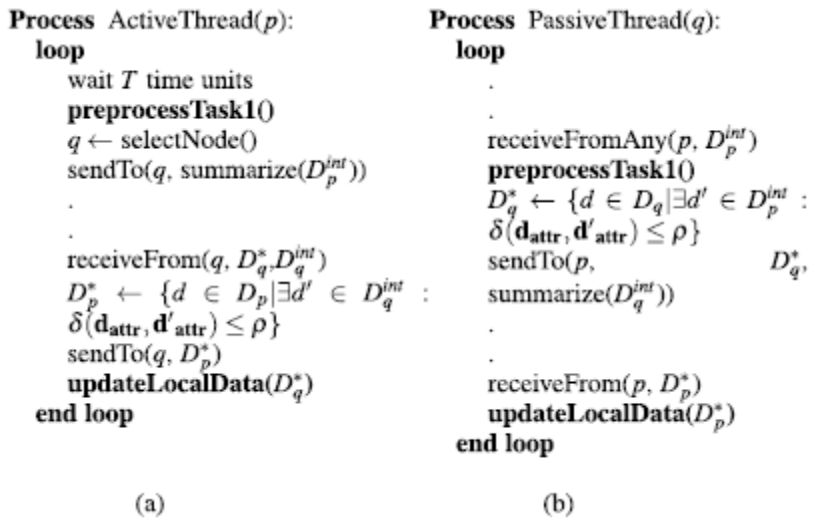

Fig. 3. Task DERIVE: (a) active thread for p and (b) passive thread for selected node q.

B.DERIVE

To derive representatives for part of the data set located near D_p^{int} p should have an accurate and up-to-date view of the data located around each data $d \in D_p^{int}$. In each round of the DERIVE task, each node p selects another node q for a three-way information exchange, as shown in Fig. 3. It should first send D_p^{int} to node q. If size of D_p^{int} is large, it can summarize the internal data by an arbitrary method such as grouping the data using clustering, and sending one data from each group. Node p then receives from q, data items located in radius ρ of each $d \in D_p^{int}$, based on a distance function ∂. ρ is a user-defined threshold, which can be adjusted as p continues to discover data. In the same manner, it will also send to q the data in D_p that lie within the . ρ radius of data in D_q^{int}. The operation updateLocalData() is used to add the received data to D_p^{ext}.

C.COLLECT

To fulfill the COLLECT task, each node p selects a random node every T time units, to exchange their set of representatives with each other (Fig. 5). Both nodes store the full set of representatives. The summarize function used in the algorithm, simply returns all the representatives given to it as input. A special implementation of this function is described in Section 5.1, which reduces the number of representatives.

Initially, each node has only a set of internal data items, D_p^{int}. Thus, the set of representatives at each node is initialized with all of its data items, i.e.,$R_p = D_p^{int}$.

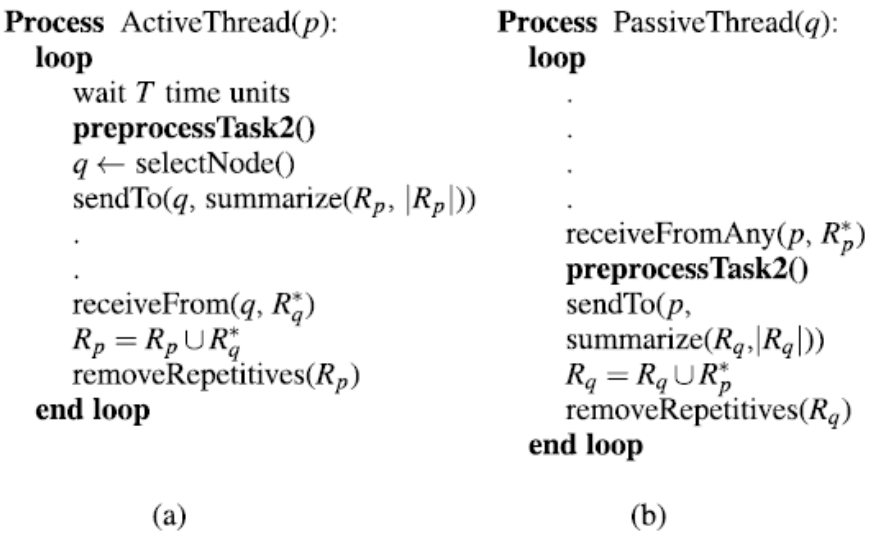

Fig. 5. Task COLLECT: (a) active thread for p and (b) passive thread for selected node q

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

IV. PROPOSED WORK

In this paper, we propose GD clustering using K-Harmonic Means (KHM) . KHM is a class of center-based iterative clustering algorithms, K-Harmonic Means (KHM), which is essentially insensitive to the initialization of the centers, demonstrated through many experiments. The insensitivity to initialization is attributed to a dynamic weighting function, which increases the importance of the data points that are far from any centers in the next iteration. The dependency of the K-Means ' and EM's performance on the initialization of the centers has been a major problem. Many have tried to generate good initializations to solve the sensitivity problem. KHM addresses the intrinsic problem by replacing the minimum distance from a data point to the centers, used in K-Means, by the Harmonic Averages of the distances from the data point to all centers. KHM significantly improves the quality of GD clustering results comparing with both K-Means and EM.

K-Harmonic means clustering
The KHM clustering algorithm **F** is executed on the set of representatives in a node. Node p can execute a weighted version of the clustering algorithm on R_p, any time it desires, to achieve the final clustering result. In a static setting, continuous execution of DERIVE and COLLECT will improve the quality of representatives causing the clustering accuracy to converge. The harmonic average is defined as

$$HA(\{a_1, a_2 \ldots\ldots\ldots\ldots a_k\}) = \frac{K}{\sum_{k=1}^{K} \frac{1}{a_k}}$$

This function has the property that if any one element in $a_1, a_2 \ldots\ldots\ldots a_k$ is small, the Harmonic Average will also be small. If there are no small values the harmonic average will be large. It behaves like a minimum function but also gives some weight to all the other values.

IV. DISCUSSION

When performing general decentralized clustering algorithm using K-means, it have some problems. The K-Means (KM) algorithm is a popular center based algorithm which attempts to find a K-clustering which minimizes MSE. mean-square quantization error (MSE) is a popular performance function for measuring goodness of data clustering. The dependency of the K-Means performance on the initialization of the centers is a major problem. Hence we propose a method of GD clustering using K-harmonic means algorithm. Which is capable of clustering dynamic and distributed data sets. K-Harmonic Means is essentially insensitive to the initialization of the centers, so that its performance does not depend on the initialization of centers.

V. CONCLUSION

In this paper we first identified the necessity of an effective and efficient distributed clustering algorithm. Dynamic nature of data demands a continuously running algorithm which can update the clustering model efficiently, and at a reasonable pace. We introduced GD Cluster using K-harmonic mean algorithm, The proposed method enable nodes to gradually build a summarized view on the global data set and perform clustering efficiently.

REFERENCES

[1] N. Visalakshi and K. Thangavel, "Distributed data clustering: comparative analysis," in Foundations Computational Intelligence, vol. 206, A.Abraham, A.-E. Hassanien, A. de Leon, F. de Carvalho, and V. Snasel, Eds, Berlin, Germany: Springer-Verlag, 2009,pp. 371–397.
[2] M. Eisenhardt, W. Muller, and A. Henrich, "Classifying documentsby distributed P2P clustering," in Proc. Informatik, Sep. 2003, pp. 286–291.
[3] R. Wolff, K. Bhaduri, and H. Kargupta, "A generic local algorithm for mining data streams in large distributed systems," IEEE Trans. Knowl. Data Eng., vol. 21, no. 4, pp. 465–478, Apr. 2009.
[4] K. M. Hammouda and M. S. Kamel, "Hierarchically distributed peer-to-peer document clustering and cluster summarization,"IEEE Trans.Knowl. Data Eng., vol. 21, no. 5, pp. 681–698, May2009.
[5] 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.