제30회 한국정보처리학회 추계학술발표대회 논문집 제15권 제2호 (2008. 11)

센서 네트워크 기반의 홀리스틱 분산 클러스터링 알고리즘

진 평*, 임기욱**, 남지은**, 이경오**
*중경우전대학 자동화학과
**선문대학교 전자계산학과
e-mail : cp02210321@hotmail.com, rim@sunmoon.ac.kr, njy1204@sunmoon.ac.kr, leeko@sunmoon.ac.kr

A holistic distributed clustering algorithm based on sensor network

Chen Ping*, Kee-Wook Rim**, Nam Ji-Yeun** Lee KyungOh**
*Dept. of Automation, Chongqing University of Posts and Telecommunication
**Dept. of Computer Science, Sun Moon University

Abstract

Nowadays the existing data processing systems can only support some simple query for sensor network. It is increasingly important to process the vast data streams in sensor network, and achieve effective acknowledges for users. In this paper, we propose a holistic distributed k-means algorithm for sensor network. In order to verify the effectiveness of this method, we compare it with central k-means algorithm to process the data streams in sensor network. From the evaluation experiments, we can verify that the proposed algorithm is highly capable of processing vast data stream with less computation time. This algorithm prefers to cluster the data streams at the distributed nodes, and therefore it largely reduces redundant data communications compared to the central processing algorithm.

Keyword : clustering algorithm, data streams, k-means

1. Introduction

Sensor networks can be deployed for many applications from environment monitoring to military surveillance, and sensor networks are consisted of many nodes with sensing, computing, and wireless communication capabilities. With the recent advance in hardware development, deployment of wireless sensor networks could be largely stimulated. Sensor nodes promise to collect a necessary amount of data that we are interested in around our environment, and sensor nodes also process the data and send them to terminal users. However, sensor nodes have limited resources [1]. First, sensor nodes usually have limited communication bandwidth resulting in limited quality of service and frequent drops of data packets. Second, sensor nodes use battery for power supply, and power efficiency is required. Third, sensor nodes have limited computing capability and memory sizes, which restrict the ability of outputting intermediate results to be processed and stored on the sensor nodes. Therefore, it is critical to use these limited resources effectively and satisfy the users' special requests. Some universities have successfully developed sensor network query processing systems for data processing, and the best known ones are Cougar [6] and TinyDB [7].

Many researchers have done a series of contributions for sensor network, including data computing, data stream aggregating in sensor networks, and querying from users.

Among these contributions, some new methods have been proposed [1],[2],[3],[4],[5]. In order to evaluate self-join efficiently, in-network execution [2] of monitoring queries has been proposed. These efforts enable wireless sensor networks (WSN) to be easily deployed and used. For the distributed sensor network, however, clustering distributed data streams is a better approach.

Clustering technology is a well-known and widely used data analysis method in data mining. In fact, we can consider the sensor network system as a distributed system, and in this distributed environment it is difficult and ineffective to cluster all the data streams at a centralized node. Most of the exiting clustering algorithms need to collect all the data streams at a central node. This is not desirable for the sensor networks and not scalable either. Sensor networks communicate in a peer to peer style, which allows only local communication among neighboring sensor nodes. It needs the data processing algorithm to work in such environment, and there are not too many algorithms are proposed in the literature fit for this distributed environment. Therefore, it is necessary to design distributed algorithm tailored for sensor network.

Section 2 describes some related work for clustering data streams in sensor network. Section 3 provides the clustering theory and the proposed holistic distributed k-means algorithm. Section 4 does some test experiments for evaluation of the algorithm. Finally, section 5 concludes the future work of this article.

2. The related work

In sensor network, the data stream processing algorithms mostly focus on centralized and distributed styles. The distributed methods process the data locally at independent sensor nodes. The centralized methods need to collect all the data streams to a central node, and then continue to analyze the related information for the data sets at the central node.

2.1 Analysis of distributed methods

In this section, in general, we present a short view of the distributed methods for clustering data streams. The distributed clustering algorithm is classified into two categories. The first category needs to perform multiple rounds of information transfer. Therefore, it requires high synchronization with the system. The second category uses local clusters to transmit information to the central node, and the central node combines all the clusters from the distributed nodes. This category requires low synchronization for the whole system, especially suitable for the sensor network. In sensor network, with the topology changing fast, it is difficult to keep synchronization for the system. Theses distributed algorithms only require a simple round of information passage. It also has the advantage of saving power for the system.

A data clustering algorithm on distributed memory multiprocessors is implemented in Ref. [8]. The algorithm uses inherent data parallelism in the k-means algorithm. For the input dataset, the algorithm divides it into n blocks and each node computes and updates current k centroid based on the local data. All the nodes broadcast the centroid. After a node has gotten all the centroid from others, it can form global centroid using an average value. Some other distributed clustering algorithms are in related literature. [9], [10],[11]

2.2 Analysis of centralized methods

Most of the distributed clustering algorithms are working in asynchronous manner through generating the clusters locally and then combine the distributed clusters at the central node. These methods require low level synchronization of the system. But when the local patterns are much smaller than the local data, the information passage will become very complex. Therefore, there is a trade-off between local processing and communication cost.

The conventional central methods collect all of the data streams together to a central node or a sink node. After the sink node gets all of the data streams from the sensor nodes, the algorithm processes the data streams. This kind of algorithms have much higher communication cost than the distributed algorithms, and consumes more power too. Therefore, it is not a good choice for the resource limited sensor network.

3. The conventional k-means algorithm and the holistic distributed k-means algorithms

In this section we will give a brief description of k-means algorithm and the proposed holistic distributed k-means algorithm. In sensor network, the nodes communicate with other nodes using ad-hoc style, and every node only communicates with the nodes in its communication range. The power consumption for the ad-hoc style communication is much higher than local computation. Therefore, when we develop a data processing algorithm for sensor network, we need to reduce data transmission to the minimum.

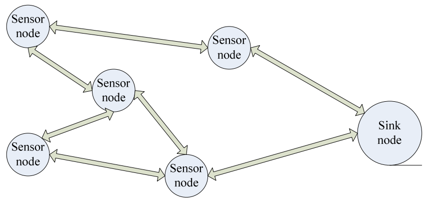

Figure 1 conventional sensor network topology

In sensor network, we assume the network topology as the conventional network topology as Figure 1. The network architecture has one sink node, which is used for gathering data from sensor network; and n sensor nodes, which are used for data stream detecting. The sink node has higher capability and can communicate with users through wire or wireless network. We note that data stream from the sensor node as $X = \{X_1, X_2, ...X_n\}$, X stands for the entire data streams sample from the environment. Here, $X_i = \{x_1, x_2, ...x_m\}$, (i=1,2…n) is the sub-data stream from the sensor node i, and at this sensor node it gets m data points from the environment. Our target processing algorithm is clustering the sub-data stream X_i into k clusters, and then combines the k clusters from the distributed n nodes into one whole k cluster at the sink node.

3.1 Conventional k-means clustering algorithms

The conventional k-means algorithm is described as follows. For the dataset $X = \{x_1, x_2, ..., x_n\}$, it divides the dataset into k clusters. It places all similar data in one cluster, and data among different clusters have very low similarity. In this way, it divides the whole dataset into k different clusters. The similarity defines as the average value of the whole data in a cluster.

Conventional k-means algorithm:
Input: the clusters value k and dataset X;
Output: k clusters, minimum square-error value
Steps:
(1) Random select k data points as initial cluster centers
(2) Allocate data points to the closest centroid, and get the k different clusters
(3) Update the average value of every cluster
(4) Compute the formula function F
(5) Until F does not change significantly

Here $$F = \sum_{i=1}^{k} \sum_{x \in c_i} \left| x - \overline{x_i} \right|^2$$

In this equation, C_i means the i cluster, x means the data point in sensor network, $\overline{x_i}$ means the average value of the cluster.

The conventional k-means algorithm tries to find the k clusters, which makes the formula function F the minimum value. If the dataset is dense and the difference between different clusters is large, the algorithm can get nice results. The algorithm is high-efficiency for the computation complexity is $O(nkt)$. Usually, k<<n, t<<n, the algorithm stops when it gets the local optimization. The conventional k-means algorithm has been widely used in different areas and is effective.

3.2 Central k-means algorithm

The central k-means algorithm processes the data streams after the sink node collects all of the data streams from distributed sensor nodes. When the sink node gets all the data streams, it uses the k-means algorithm to process the data streams.

Central k-means algorithm steps are as follows:

(1) The sink node sends messages to all the distributed sensor nodes for data collecting

(2) The distributed sensor nodes send the sampling data streams to the sink node, and the sink node acknowledges the data receiving.

(3) The sink node processes the data streams using k-means algorithm and outputs the final result.

After the sink node collects all of the data streams from the other nodes, the algorithm processes them, and outputs the clusters for future querying.

3.3 Holistic distributed k-means algorithm

For this algorithm, the sink node selects k data points from the network randomly as the initial centroid, and sends the initial centroid to every sensor nodes. At the local sensor nodes, every node divides the data points into k local clusters. The local sensor nodes then send their respective k clusters back to the parent node. When the sink node gets all of the sub-nodes information and combines all of them together to form k new clusters, the sink node re-computes the average value of every cluster. The sink node estimates whether it satisfies stopping criteria. If it does not satisfy the stopping criteria, the new computed k centroid will send to the sensor network, and start the next iteration.

Description of the algorithm steps:

(1) The sink node selects k data points from the sensor network randomly as the initial centroid, and sends to every sensor node in the network to initiate the square-error sum F

(2) Every senor node uses the centroid received from the sink node to cluster the local data streams into k-clusters, and then send the k-clusters back to parent nodes and combine the clusters with the parent nodes

(3) The sink node combines all of the sub-clusters into holistic k clusters, and judges whether it satisfies the stopping criteria (whether the square-error sum is larger than initial value F). If the square-error sum is larger than F, replaces it and re-computes the average value of the data points in cluster as the new centroid. Send the new centroid to the sensor nodes and repeats (2) and (3)

(4) If the square-error sum is smaller than initial value F, the algorithm satisfies the stopping criteria. It will then stop the algorithm and get the final clusters and centroid.

4. Experiments and evaluation

In this section, in order to evaluate the proposed holistic distributed k-means algorithm we need to simulate a real sensor network. We divide the experiment into two parts. The first one is simulating the dataset generating in sensor network, and the second part is using the simulating dataset to evaluate the proposed algorithm and compare it with the central processing algorithm.

4.1 Experiments test

We use a program to simulate a real sensor network to produce data streams. We call it “sensor nodes dataset generator”, which can produce the data streams for experiments. We assume every sensor node can store 200 data points, and every data point has 4 attributes such as temperature, humidity, etc. These datasets need to divide into 3 clusters. In this simulating condition, we evaluate the proposed algorithm and compare the executing time with central k-means algorithm.

4.2 Evaluation and analysis

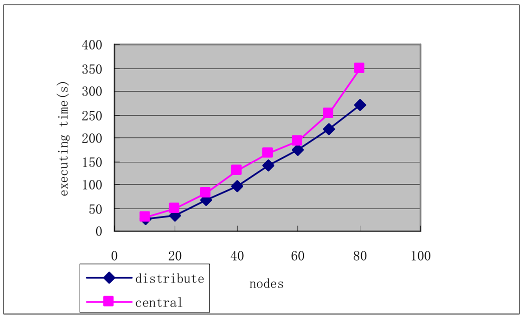

Figure 2 Executing time for the distributed and central algorithms

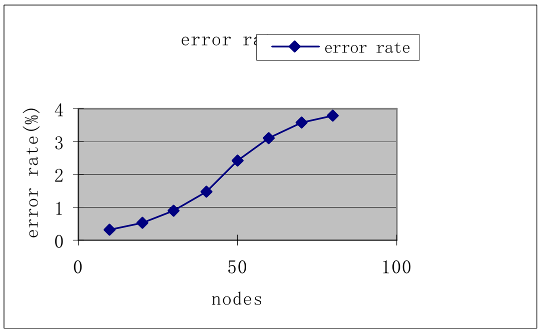

Figure 3 The error rate as a function numbers of nodes

We gave the same initial k centroid to the two processing algorithms at the beginning. We compared the executing time for the two algorithms with the increasing number of sensor nodes. From the results in Figure 2, we can conclude that the proposed algorithm has less executing time with large sensor nodes. From Figure 2, we can also find that the central k-means algorithm has higher communication cost due to its longer executing time. We also tested the error rate of the proposed algorithm as shown in Figure 3. It turns out that the proposed algorithm has low error rate for clustering the dataset from the sensor network.

5. Conclusion and future work

A new distributed k-means algorithm has been proposed for processing large dataset. In order to make the algorithm suitable for the sensor network, the conventional k-means has been improved to holistic distributed k-means algorithm. It has been shown with simulation that the holistic distributed k-means algorithm is indeed superior to the central k-means algorithm for large dataset. The future work will consider other algorithms and make them scalable for different sensor network.

Acknowledgments:
This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) Support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2008-C1090-0801-0020)

References

[1] Yong Yao and Johannes Gehrke . Query Processing for Sensor Network. Proceedings of the 2003 CIDR Conference.
[2] X. Yang, H. B. Lim, M. T. Ozsu, K. L. Tan. In-Network Execution of Monitoring Queries in Sensor Networks. SIGMOD 2007
[3] M. Garofalakis, J. Gehrke, R. Rastogi. Querying and Mining Data streams: You Only Get One Look. A Tutorial. VLDB 2002
[4] M. Wu, J. Xu, X. Tang, W.-C. Lee.Top-k Monitoring in Wireless Sensor Networks. TKDE Jul. 2007
[5] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A.Miu, E. Shih, H. Balakrishnam, S. Madden. CarTel: A Distributed Mobile Sensor Computer System. SenSys2006
[6] Cougar web page. http://www.cs.cornell.edu/database/cougar/.
[7] Samuel Madden, Wei Hong, Joseph M. Hellerstein, and Michael Franklin. TinyDB web page.
[8] I. Dhillon, D. Modha, A data-clustering algorithm on distributed memory multiprocessors, in:
Proceedings of the KDD'99 Workshop on High Performance Knowledge Discovery, 1999, pp.
245–260
[9] G. Forman, B. Zhang, Distributed data clustering can be efficient and exact, SIGKDD
Explorations 2 (2) (2000) 34–38
[10] H. Kargupta, W. Huang, K. Sivakumar, E. Johnson, Distributed clustering using collective
principal component analysis, Knowledge Inform. Syst. J. 3 (2001) 422–448
[11] M. Klusch, S. Lodi, G. Moro, Distributed clustering based on sampling local density estimates, in: Proceedings of the Joint International Conference on AI (IJCAI 2003),2003