Asian Journal of Scientific Research

ISSN 1992-1454

science alert
http://www.scialert.net

ANSInet
an open access publisher
http://ansinet.com

Asian Journal of Scientific Research 1 (4): 444-450, 2008
ISSN 1992-1454

A Semaphore Based Multiprocessing k-Mean Algorithm for Massive Biological Data

[1]M. Hemalatha and [2]K. Vivekanandan
[1]Karpagam Arts and Science College, Coimbatore-21, India
[2]BSMED, Bharathiar University, Coimbatore, India

Abstract: In the present scenario, the concept of Distributed Processing System is more beneficial with respect to time saving, cost reduction and more clarity of clustering process. Earlier, while clustering huge amount of data it consumed lot of time, energy and cost. Now, by applying parallel and distributed approach, we can minimize the total time necessary for clustering the data thereby reducing cost. In this research, parallel and distributed version of k-means clustering algorithm is proposed. The proposed algorithm will be implemented using Matlab and will be tested with large synthetic data sets.

Key words: Distributed, semaphore, cluster, parallel, k-means

INTRODUCTION

Clustering is an unsupervised operation. It is used where you wish to find groupings of similar records in your data without any preconditions as to what that similarity may engross. Clustering is used to identify attractive groups in any type of data that may not have been acknowledged before. Distributed processing today is a largely advantageous technology of bridging together a system of multiple computers and processor systems in running applications. Distributed and parallel processing plays a vital role in applications of clustering the huge amount of data. A primary goal of developing new algorithms, processors etc. are to perform large, complex data faster. After human genome project the biological databases are increased enormously with massive amount of data. Anyhow, a large task can either be performed serially, one step following another, or can be decomposed into smaller tasks to be performed simultaneously, i.e., in parallel.

There are many different types of distributed computing systems and many challenges to overcome in successfully designing one. The problem of clustering huge amount of data is a very time consuming operation. An efficient algorithm is required to cluster the huge amount of data. This research proposes a simple semaphore based distributed multi processing architecture for clustering bulk, multi dimensional data sets.

Distributed Data Mining (DDM) model assumes that the data sources are distributed across multiple sites. Algorithms developed within this field address the problem of efficiently getting the mining results from all the data across these distributed sources. Since the primary (if not ouly) focus is on efficiency, most of the algorithms developed to date do not take security consideration into account.

Issues that cause a disparity between local and global results include: Values for a single entity may be split across sources. Data mining at individual sites will be unable to detect cross-site correlations. The same item may be duplicated at different sites and will be over-weighted in the results. Data at a single site is likely to be from a homogeneous population, hiding geographic or demographic distinctions between that population and others. Algorithms have been proposed for distributed data mining. Distributed classification has also been addressed. A meta-learning approach

Corresponding Author: M. Hemalatha, 397, Maruthachalam lane, Puliyakulam Road, Sungam, Coimbatore-45, Tamil Nadu, India Tel: 0422-2611082

has been developed that uses classifiers trained at different sites to develop a global classifier (Goldreich *et al*., 1987). This could protect the individual entities, but it remains to be shown that the individual classifiers do not disclose private information. Recent work has addressed classification using Bayesian Networks in vertically partitioned data (Kantarcioglu *et al*., 2002) and situations where the distribution is itself interesting with respect to what is learned (Data Mining News).

However, none of this work addresses privacy concerns. There has been research considering how much information can be inferred, calculated or revealed from the data made available through data mining algorithms and how to minimize the leakage of information (Agrawal and Srikant, 2000). However, this has been restricted to classification and the problem has been treated with an all or nothing approach. We desire quantification of the security of the process. Corporations may not require absolute zero knowledge protocols (that leak no information at all) as long as they can keep the information shared within bounds.

In Agrawal and Srikant (2000), data perturbation techniques are used to protect individual privacy for classification, by adding random values from a normal/Gaussian distribution of mean 0 to the actual data values). One problem with this approach is a trade of between privacy and the accuracy of the results (Hambaba, 1996). More recently, data perturbation has been applied to boolean association rules. One interesting feature of this work is an exible definition of privacy; e.g., the ability to correctly guess a value of 1 from the perturbed data can be considered a greater threat to privacy than correctly learning a 0. (Chen *et al*., 2001) Use cryptographic protocols to achieve complete zero knowledge leakage for the ID3 classification algorithm for two parties with horizontally partitioned data. There has been work in cooperative computation between entities that mutually distrust one another. Secure two party computations was first investigated by Yao (1986) and later generalized to multiparty computation. The seminal paper by Goldreich proves existence of a secure solution for any functiouality (Prodromidis *et al*., 2000). The idea is as follows: the function F to be computed is first represented as a combinatorial circuit and then the parties run a short protocol to securely.

MATERIALS AND METHODS

Normal k-Means Algorithms

k-means algorithm is very popular one for data clustering. Generally, k-means algorithm is used several iterations to cluster the data since the result is very much depend on the initial guess of the cluster centers. The Algorithm goes like this

- Start iteration
- Select k center in the problem space (it can be random).
- Partition the data into k clusters by grouping points that are closest to those k centers.
- Use the mean of these k clusters to find new centers.
- Repeat steps 2 and 3 until centers do not change.
- Calculate the total distances between cluster centers and all the points of each cluster.
- Repeat the steps 2 to 6 for N iterations.
- Among the N results, find the result with minimum distance.
- Display the results corresponding to that minimum distance.
- Stop iterations.

The algorithm is simple and has nice convergence but there are number of problems with this

- Selection of value of k is itself an issue and sometimes its hard to predict before hand the number of clusters that would be there in data.
- Experiments have shown that outliers can be a problem and can force algorithm to identify false clusters.

- So we have the repeat the algorithm several times and select the result with minimum distance.
- Experiments have shown that performance of algorithms degrade in higher dimensions and can be off by factor of 5 from optimum.

The Proposed Distributed Multi Processing Based k-Means Algorithm

The job-termination and resumption-model used in this algorithm. The model is inspired by the concept of a semaphore, which is a built-in system data type, with an associated lock with locking and unlocking operations. Semaphores are used for synchronization between multiple processes, when in critical sections. In the resumption model, a single thread may be in critical sections when it is time to terminate. The semaphore concept is used to lock out the section, so that termination can occur ouly after the thread has exited the section and released the lock. Hence the lock is checked continually to ascertain whether or not critical sections are exited. Table 1 shows the hardware details used to test the proposed system.

The following is a semaphore based multiprocessing k-mean algorithm.

- From the main process which is running in a main computer, Prepare N files with a clustering function handler and Matrices for input and output parameters. Keep the files in a globally accessible shared network space.
- From each computer, do the following
- Select a file from shared network location, check its status.
- If the file is already processed or locked by some other process, then skip that file and select another file and check the same.
- If a file is unprocessed and available, then immediately lock it to have the exclusive access to that file.
- Read the File content and get the clustering function handler and input parameters Data and number of clusters k and run the function locally.
- Select k Center in the problem space (it can be random).
- Partition the data into k clusters by grouping points that are closest to those k centers. Use the mean of these k clusters to find new centers.
- Repeat steps 2 and 3 until centers do not change.
- Calculate the total distances between cluster centers and all the points of each cluster. Save the calculated cluster labels and cluster centers in appropriate matrices in the file for future calculations and unlock the file.
- Parallely Repeat the steps 4 to 13 until there is no file left to be processed. From the main process of the main computer, read all the N files.
- Among the N results, find the result with minimum distance. Display the results corresponding to that minimum distance.
- Stop iterations.

Table 1 : Hardware used to form the distributed system

S. No.	System description	System configuration	Software
1	Single Core PCWIN Workstation I (PC-I)	Intel Celeron M Processor, 1.6 GHz, 400 MHZ FSB, IMB L2 Cache, 512 MB DDR RAM, 100, MBPS Ethernet card, 80 GB HDD	Matlab 6.5 on Windows XP
2	Single Core PCWIN Workstation II (PC-II)	Intel Celeron, 2.8 G.Hz, 266 M.Hz FSB, 256 KB L2 Cache, 512 MB DDR RAM, 100, MBPS Ethernet card, 80GB HDD	Matlab 6.5 on Windows XP
3	Single Core PCWIN Workstation III (PC-III)	Intel Celeron, 2.8 G.Hz, 266 M.Hz FSB, 256 KB L2 Cache, 512 MB DDR RAM, 100, MBPS Ethernet card, 80GB HDD	Matlab 6.5 on Windows XP

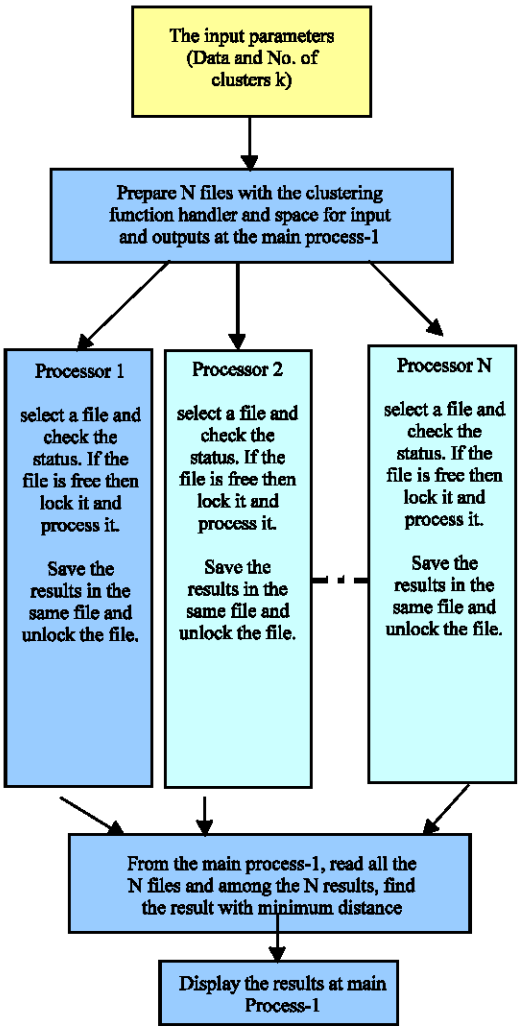

Fig. 1: The block diagram of proposed architecture

The following pseudo-code illustrates the model for the worker thread/process.

```
while (true)
{
 reader_next_job();
  If (!job_to_process)
  {
    acquire_lock();
    Do the job();
    Save the Results();
    release_lock();
    reader_next_job();
}
```

Figure 1 shows the block-diagram of the Semaphore based k-means clustering model

RESULTS

To evaluate the performance of the two algorithms in terms of speed (CPU time) and accuracy (R index), a set of numeric data was used. The results are discussed in the following paragraph.

Very large multidimensional synthetic numeric data sets were used to test. This data sets was created randomly by Gaussian distribution, it is used to measure the speed and accuracy of clustering.

The common attributes of synthetic input data

The number of classes	= 5
The number of dimensions	= 25
The Number of records per class	= 100-500
The Standard deviation	= 0.7500
The total number of records	= 500-2500

Result with single core

Time needed for 10. Runs	: 6.31 sec
The Rand index is	: 1.00

Result with two cores

Time needed for 10. Runs	: 5.08 sec
The Rand index is	: 1.00

Result with three cores

Time needed for 10. Runs	: 3.16 sec
The Rand index is	: 1.00

Plotting

Figure 2 shows the plotting of original and k-means, Fig. 3 shows the plotting of original and proposed distributed and semaphore based k-means clustering model, Fig. 4 shows the plotting of k-means and semaphore based k-means clustering model for the above dataset.

Figure 2 is clear that there is no much difference between the original plotting and k-means plotting. It is also true that the accuracy of clustering is good in k-means clustering. On comparison with the original plotting it is found that the semaphore based k-means clustering model is more or less similar to the original. Thus the clustering accuracy (Rand Index) of proposed model is good.

By comparing k-means and proposed semaphore based k-means clustering model it is obtained that there is difference in accuracy and speed. And it is also noticed that the clustering distributed enviroument gives a significant performance of proposed than the performance of k-means.

The Table 2 shows the overall performance results obtained for the above dataset. The time taken for clustering is noted for 5 sets of data and the dimensions with the interval of 25 dimensions. This table is used to analyze more results about the performance of the proposed model.

It is concluded that there is a rise in difference of time taken and time consumed by the semaphore based k-means clustering model to form the clusters is less than the k-means. The accuracy (Rand index) seems to be high always (Fig. 5). Therefore the accuracy is high than the normal k-means and the overall result is, the performance of semaphore based k-means clustering model is better than the normal k-means.

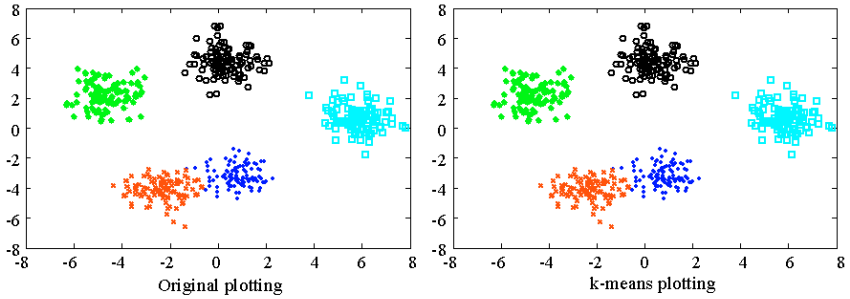

Fig. 2: Comparisons of original and normal k-means in data clustering

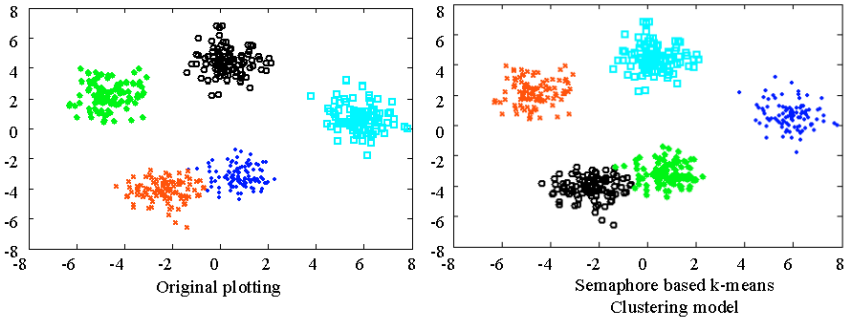

Fig. 3: Comparisons of original and semaphore based k-means clustering model in data clustering

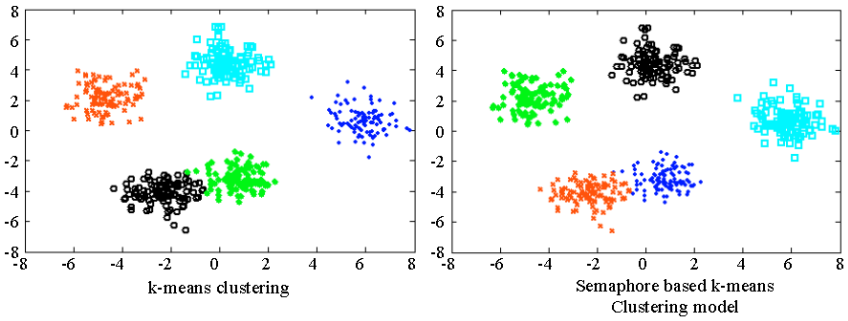

Fig. 4: Comparisons of k-means and semaphore based k-means clustering model in data clustering

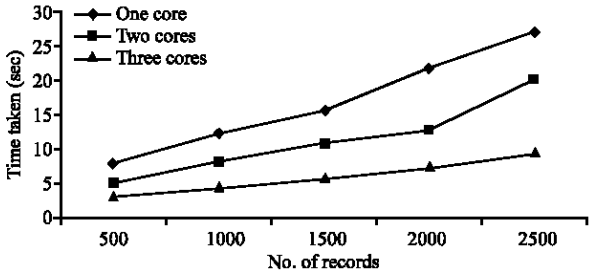

Fig. 5: Clustering synthetics data in terms of speed

Table 2: Clustering datasets in different no of cores to show the performance of time and accuracy

S. No.	No. of records	Time taken for clustering (sec)			R-index
		One core	Two cores	Three cores	
1	500	7.98	5.08	3.16	1.00
2	1000	12.34	8.11	4.41	1.00
3	1500	15.66	10.88	5.77	1.00
4	2000	21.88	12.84	7.28	1.00
5	2500	27.20	20.30	9.41	1.00

CONCLUSIONS AND FURTHER MODELING

To clustering large amount of data, the proposed distributed and semaphore based k-means clustering model has been successfully designed and implemented on MATLAB under Windows operating system using several normal desktop computers. The Performance of the classification algorithm was tested with the very large multidimensional numeric data sets synthetically created randomly by Gaussian distribution. Several tests were made on the system and overall significant results were achieved. As Shown in the graphs, the performance of the algorithm was improved while increasing the cores. The average accuracy of classification defined by the rand index calculated using the calculated labels and true class labels. In the case of distributed k-mean Clustering, it is one since ideal synthetic data is used. Privacy issues in parallel and distributed data mining can be addressed in future work. In future works, the performance of the proposed system and the achieved results may be verified with large medical and bio-informatics data sets. If the system will be implemented by using a suitable programming language such as C or C++, then we can reach better performance necessary for practical applications. These issues also can be addressed in future works.

ACKNOWLEDGMENT

The authors thank the management for their kind support doing the research at Karpagam Arts and Science College, Coimbatore.

REFERENCES

Agrawal, R. and R. Srikant, 2000. Privacy-preserving data mining. In Proceedings of the 2000 ACM SIGMOD Conference on Management of Data, Dallas, TX, May 14-19 ACM.

Chen, R., K. Sivakumar and H. Kargupta, 2001. Distributed web mining using bayesian networks from multiple data streams. In: The 2001 IEEE International Conference on Data Mining. IEEE. Nov. 29- Dec. 2.

Goldreich, O., S. Micali and A. Wigderson, 1987. How to play any mental game - a completeness theorem for protocols with honest majority. In: 19th ACM Symposium on the Theory of Computing, pp: 218-229.

Hambaba, M.L., 1996. Computational intelligence for financial engineering. Proceedings of the IEEE/IAFE 1996 Conference on Digital Object Identifier 10.1109/CIFER.1996.501832, 24 (26): 111.

Kantarcioglu, M. *et al*., 2002. Privacy-preserving distributed mining of association rules on horizontally partitioned data. In: The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02).

Prodromidis, A., P. Chan and S. Stolfo, 2000. Meta-learning in Distributed Data Mining Systems: Issues and Approaches, Chapter 3. AAAI/MIT Press.

Yao, A.C., 1986. How to generate and exchange secrets. In: Proceedings of the 27th IEEE Symposium on Foundations of Computer Science, pp: 162-167.