

# A Distributed and Parallel Clustering Algorithm for Massive Biological Data

M.Hemalatha\**Corresponding author*, P. Ranjith Jebah Thangiah,, K.Vivekanandan

1. Karpagam Univesity,Coimbatore -21,Tamil Nadu, India

Research Scholar, School of Computer Science and Engineering,

2. Bharathiar University, Coimbatore - 641 046. Director/IC, BSMED,

3. Bharathiar University, Coimbatore, Tamil Nadu, India

*kvivekbsmed@gmail.com*

## Abstract

*Distributed processing today is a largely advantageous technology of bridging together a system of multiple computers and processor systems in running applications. The concept of Distributed processing has allowed time cutting and therefore reduction in costs. Using this, we aim to address clustering techniques in developing new method for further reduction in time and costs. The problem of clustering huge amount of data is a very time consuming operation. So by applying parallel and distributed approach, we can minimize the total time necessary for clustering the data. In this research, a 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.*

## Keywords

*Distributed, semaphore, cluster, parallel, k-means*

## 1. 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

Depending on the clustering technique, clusters can be expressed in different ways:

- Identified clusters may be exclusive, so that any example belongs to only one cluster.
- They may be overlapping; an example may belong to several clusters.
- They may be probabilistic, whereby an example belongs to each cluster with a certain probability.

- Clusters might have hierarchical structure, having crude division of examples at highest level of hierarchy, which is then refined to sub-clusters at lower levels.

Clustering is usually achieved using statistical methods, such as a k-means algorithm, or a special form of neural network called a Kohonen feature map network. Whichever method is used, the basic operation is the same. Each record is compared with a set of existing clusters, which are defined by their 'centre'. A record is assigned to the cluster it is nearest to, and this in turn changes the value that defines that cluster. Multiple passes are made through a data set to re-assign records and adjust the cluster centres until an optimum solution is found.

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 term semaphore is a mechanism for restricting access to critical sections of code to a single user or process at a time. In any preemptively scheduled environment, the use of semaphores is vital to protect data structures that can be accessed by more than one thread of execution The term mutex object is a synchronization object whose state is set to signaled when any thread does not own it, and non-signaled when it is owned. Its name comes from its usefulness in coordinating mutually exclusive access to a shared resource. Only one thread at a time can own a mutex object. For example, to prevent two threads from writing to shared memory at the same

time, each thread waits for ownership of a mutex object before executing the code that accesses the memory. After writing to the shared memory, the thread releases the mutex object. Based on the above techniques the proposed research was carried out. 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.

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.

## 2. Methodologies and Design 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

1. Start iteration
2. Select k Center in the problem space (it can be random).
3. Partition the data into k clusters by grouping points that are closest to those k centers.
4. Use the mean of these k clusters to find new centers.
5. Repeat steps 2 and 3 until centers do not change.
6. Calculate the total distances between cluster centers and all the points of each cluster.
7. Repeat the steps 2 to 6 for N iterations.
8. Among the N results, find the result with minimum distance.
9. Display the results corresponding to that minimum distance.
10. 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.

## 3. 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 only 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.

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

1. 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.
2. From each computer, do the following
3. Select a file from shared network location, check its status.
4. If the file is already processed or locked by some other process, then skip that file and select another file and check the same.
5. If a file is unprocessed and available, then immediately lock it to have the exclusive access to that file.
6. Read the File content, and get the clustering function handler and input parameters Data and number of clusters k and run the function locally.
7. Select k Center in the problem space (it can be random).
8. 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.
9. Repeat steps 2 and 3 until centers do not change.
10. 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.
11. 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.

12. Among the N results, find the result with minimum distance. Display the results corresponding to that minimum distance.
13. Stop iterations.

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();
    }
}

```

**Table 1.** Hardware Used to form the Distributed System

SI No	System Description	System Configuration	Software
1	Single Core PCWIN Workstation I (PC-I)	Intel Celeron M Processor, 1.6 G.Hz, 400 M.Hz 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

#### 4. Results

The proposed distributed parallel processing system has been implemented and tested in MatLab under Windows Operating System. The proposed distributed and semaphore based k-means algorithm was run in parallel on three Computers with different hardware capabilities.

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.

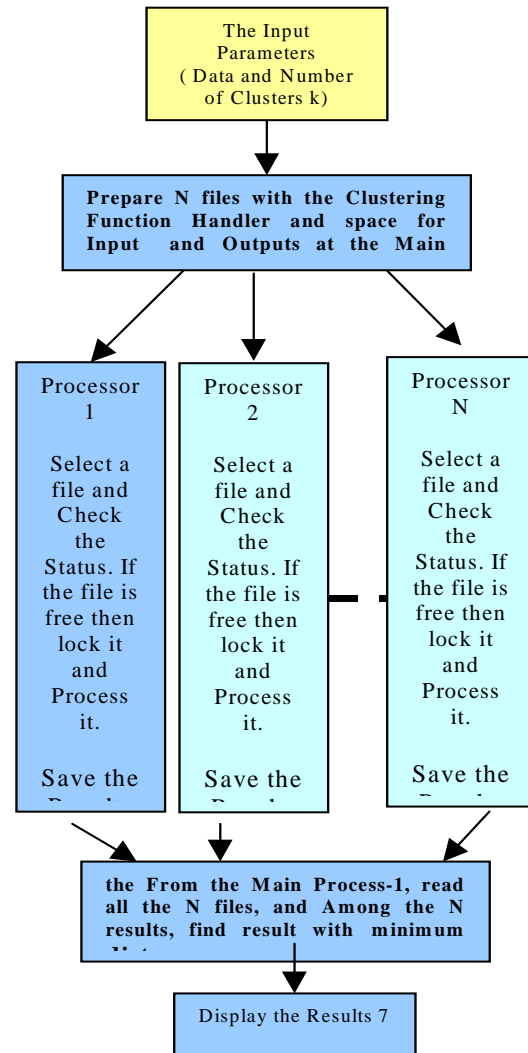
Very large multidimensional synthetic numeric data sets were used to test. This data sets was created randomly by Gaussian distribution, it is used to measure the speed and accuracy of clustering.

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

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



**Figure 1.** Block diagram of Proposed Architecture

The Total Number of Records = 500-2500  
Result With Single Core  
Time needed for 10. Runs : 6.31 seconds.  
The Rand Index is : 1.00

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 environment 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.

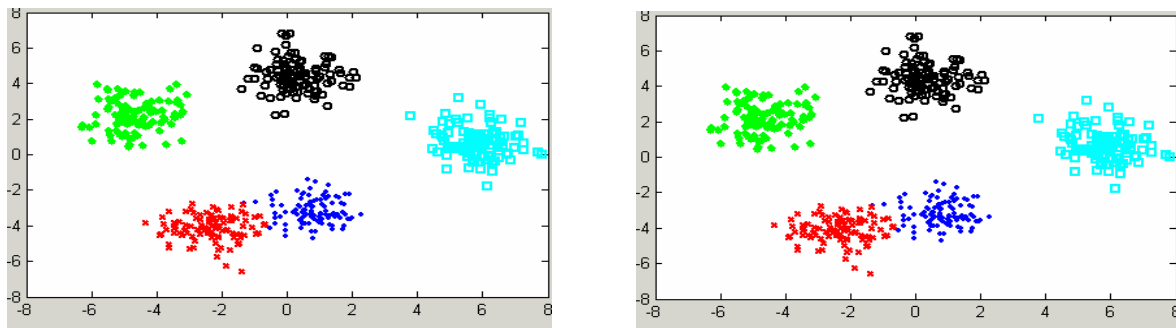
The pictorial representation of the histogram (Figure 4) clearly shows that the performance of proposed distributed model with respect to speed. It

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.

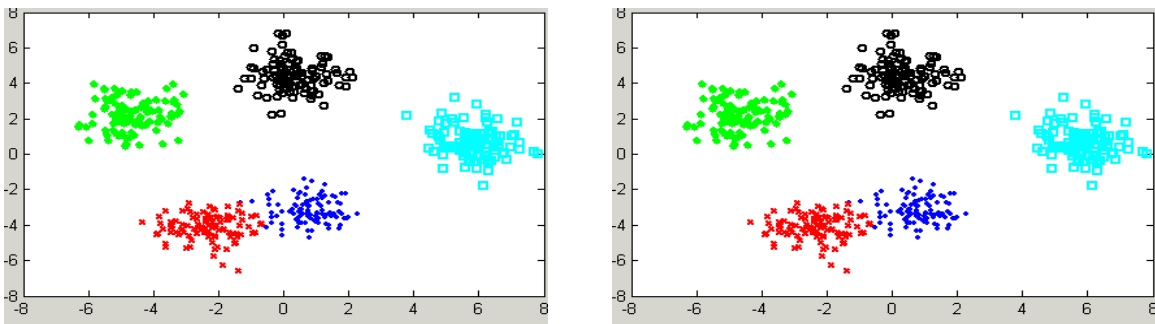
Therefore the accuracy is high than the normal k-means and the overall result is, the performance of semaphore based k-means clustering model is better than the normal k-means

**Table 2.** Clustering datasets in different no of cores to show the Performance of time and accuracy

Number of Records	Time Taken for Clustering (sec)			R-Index
	One Core	Two Cores	Three Cores	
500	7.98	5.08	3.16	1.00
1000	12.34	8.11	3.16	1.00
1500	15.66	10.88	5.77	1.00
2000	21.88	12.84	7.28	1.00
2500	27.20	20.30	9.41	1.00



**Figure 2.** Comparisons of original and semaphore based k-means clustering model in data clustering



**Figure 3.** Comparisons of k-means and semaphore based k-means clustering model in data clustering

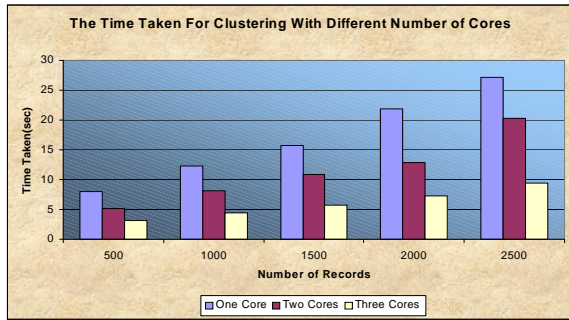


Figure 4. Clustering in terms of speed

## 5. Conclusion

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.

## 6. Acknowledgement

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

## 7. References

[1] Rizvi, S.J. and J.R. Haritsa, Privacy-preserving association rule mining, In proceedings of 28th international conference on very large data bases. VLDB, pp: 20-23, 2002.

[2] Agrawal, D. and C.C. Aggarwal, On the design and quantification of privacy preserving data mining algorithms, In proceedings of the twentieth ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems, santa barbara, california, USA, pp: 21-23, 2001.

[3] Agrawal, R. and R. Srikant., Privacy-preserving data mining, In proceedings of the 2000 ACM SIGMOD Conference on management of data, dallas, TX, pp: 14-19, 2000.

[4] R., K., Sivakumar and H. Kargupta, Distributed web mining using bayesian networks from multiple data streams. IEEE International conference on data mining, 2001.

[5] Goldreich, O., S. Micali and A. Wigderson., 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, 1987.

[6] Hambaba, M.L. Computational Intelligence for Financial Engineering, In proceedings of the IEEE/IAFE conference on 24(26): 111- Digital Object Identifier10.1109/CIFER.501832., 1996.

[7] Kantarcioglu, M *et al.*, 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, 2002.

[8] Yao. A.C., How to Generate and Exchange Secrets, In proceedings of the 27th IEEE symposium on foundations of computer science, 62(167), 1986.