Implementing and compiling clustering using Mac Queens alias K-means apriori algorithm

[1]Dr.O. Nagaraju, [2]B.Kotaiah , [3]Dr. R.A. Khan, [4]M.RamiReddy, [5]N.S.Kalyan Chakravarthy

[1]Assistant Professor, Department of CSE,
Acharya nagarjuna University, Guntur, 522510,India
onrajunrt@gmail.com
[2]Researach Scholar
Department of IT
BBA university,Lucknow
Kotaiah_bonthuklce@yahoo.com
[3]Associate Professor, Department of IT,
BBA university,Lucknow, Idia
khanraees@yahoo.com
[4]Assistant Professor,Department of Physics
Acharya nagarjuna University, Guntur, 522510,India
[5]Chairman,SNS Group,Vengamukkapalem,
Ongole-523272, Andhrapradesh, India.
suryatg@yahoo.com

Abstract

This paper aims at implementing a Symmetric Multi-Threading. The paper provides a true concurrency, also known as Symmetric Multi-Threading (IACKMA), which occur when multiple threads execute instructions during the same clock cycle. It gives high-performance to Java developers, a tremendous opportunity for speeding up programs. The proposed algorithm divides the dataset into several identical unit blocks. Then, it calculates the centroids and related statistics of patterns in each unit block to represent an approximation of the information in the unit blocks. We use this reduced data to reduce the overall time for distance calculations. We find that the clustering efficiency is closely related to determining how many blocks should be partitioned. In fact, since the algorithm uses discrete assignment rather than a set of continuous parameters, the "minimum" it reaches cannot even be properly called a local minimum. Despite these limitations, the algorithm is used frequently as a result of its ease of implementation.

Keywords-

Cluster Analysis, K-Mean Algorithm, Symmetric Multi-Threading, Mean Clustering, Objects-Centroids distances.

DOI: 10.5121/ijdms.2012.4205

1. INTRODUCTION

1.1 DATA MINING

Data Mining means mining the knowledge from the historical data. i.e. retrieving the data from the already existing data. Data Mining is the "automated extraction of hidden predictive information from databases".

Data mining tools predict future trends and behaviors, allowing us to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were very time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations. It is worth mentioning that data mining can as well be used in other fields involving research[1].

Data mining software allows users to analyze large databases to solve business decision problems. Data mining is, in some ways, an extension of statistics, with a few artificial intelligence and machine learning twists thrown in. Like statistics, data mining is not a business solution, it is just a technology.

1.2 Cluster Analysis

One of the Data Mining application is Cluster Analysis. A cluster is a group of objects and are related in a way such that the objects within the cluster are similar and the objects are dissimilar with the objects present in the another cluster.Clustering is the task of segmenting a heterogeneous population into a number of more homogeneous subgroups or *clusters.*

- Clustering does not rely on predefined classes.
- The records are grouped together on the basis of self-similarity.

Both the clustering and classification are used for grouping the objects, but the difference is, in classification whenever a new object is entered, first the characteristic of that object is checked and then it can be compared with the characteristics of the groups, if they both are having the similar characteristics then they will be grouped. Where as in clustering, whenever a new object is entered first it checks the characteristics of that object, it does not knows about the characteristics of the clusters, just by performing some mathematical calculations the object can be placed in to the corresponding cluster[2].

1.3 Clustering Applications:

$$E = \sum_{j=1}^{n_c} \sum_{i=1}^{N} \left\| v_i - c_j \right\|^2$$

- Marketing
- Biology
- Library
- Insurance

To form such type of clusters we are going to use K-Means algorithm, because this algorithm is very efficient and easy to implement. In this algorithm the mathematical calculation Centroid is used for the formation of the cluster. Generally there are three measurements for the calculation of the Centroid, those are mean, median, mode. Among these three measurements in our project mean is used.

Clustering is a process in which a group of unlabeled patterns are partitioned into a number of sets so that similar patterns are assigned to the same cluster, and dissimilar patterns are assigned to different clusters. There are two goals for clustering algorithms: determining good clusters and doing so efficiently. Clustering has become a widely studied problem in a variety of application domains, such as in data mining and knowledge discovery, statistical data analysis, data classification and compression, medical image processing and bioinformatics.

2.1 K-Means Clustering

There are two existing basic versions of k-means clustering, a non-adaptive version introduced by Lloyd and an adaptive version introduced by MacQueen. The most commonly used k-means clustering is the adaptive k-means clustering based on the Euclidean distance Adaptive k-means clustering can be considered as a special case of the gradient descent algorithm where only the winning cluster is adjusted at each learning step[3]. This paper concentrates only on adaptive k-means clustering as the algorithm can be used for on-line training of RBF network. Adaptive k-means clustering tries to minimize the cost function in fallowing equation by searching for the centre cj on-line as the data are presented. As the data sample is presented, the Euclidean distances between the data sample and all the centres are calculated by using fallowing equations.

$$d(x, y) = \sum_{i=1}^{n} |y_i - x_i|$$

2.2 Numerical Example

The basic step of k-means clustering is simple. In the beginning we determine number of cluster K and we assume the Centroid or center of these clusters. We can take any random objects as the initial centroids or the first K objects in sequence can also serve as the initial centroids.

Then the K means algorithm will do the three steps below until convergence

Iterate until stable (= no object move group):

Determine the Centroid coordinate

Determine the distance of each object to the centroids

Group the object based on minimum distance

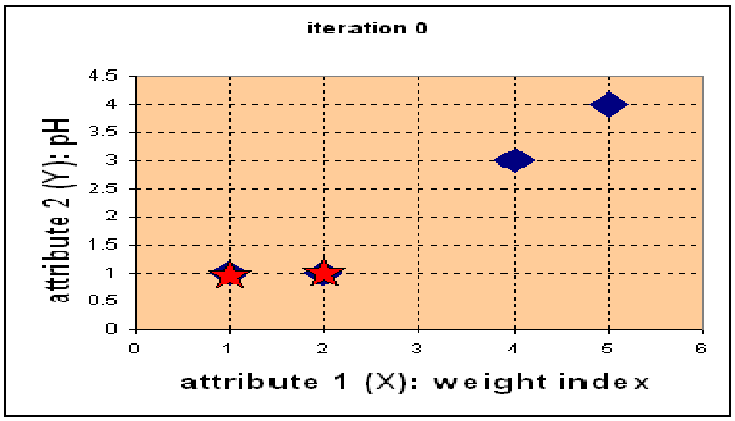
The numerical example below is given to understand this simple iteration. Suppose we have several objects (4 types of medicines) and each object have two attributes or features as shown in table below. Our goal is to group these objects into K=2 group of medicine based on the two features (pH and weight index).

Object	attribute 1 (X): weight index	attribute 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

Table 1

Each medicine represents one point with two attributes (X, Y) that we can represent it as coordinate in an attribute space as shown in the figure below.

2.2.1. Initial value of centroids : Suppose we use medicine A and medicine B as the first centroids. Let c_1 and c_2 denote the coordinate of the centroids, then $c_1 = (1,1)$ and $c_2 = (2,1)$



2.2.2 Objects-Centroids distance : we calculate the distance between cluster centroid to each object. Let us use Euclidean distance, then we have distance matrix at iteration 0 is

$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (2,1) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

Each column in the distance matrix symbolizes the object. The first row of the distance matrix corresponds to the distance of each object to the first centroid and the second row is the distance of each object to the second centroid. For example, distance from medicine C = (4, 3) to the first centroid $c_1 = (1,1)$ is $\sqrt{(4-1)^2+(3-1)^2} = 3.61$, and its distance to the second centroid $c_2 = (2,1)$ is $\sqrt{(4-2)^2+(3-1)^2} = 2.83$, etc.

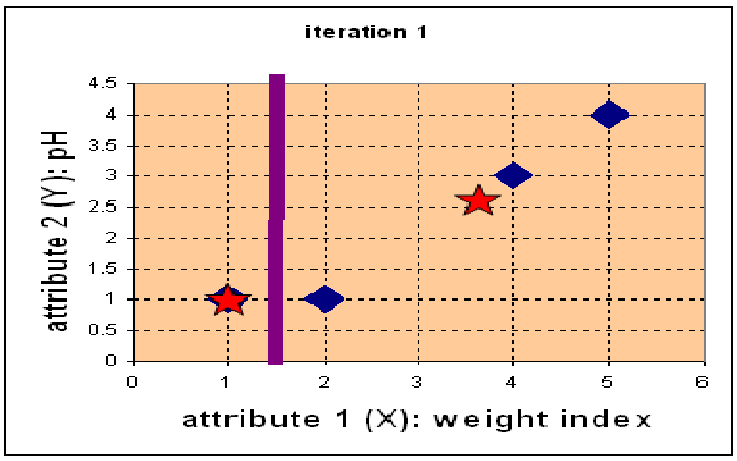
2.2.3. Objects clustering : We assign each object based on the minimum distance. Thus, medicine A is assigned to group 1, medicine B to group 2, medicine C to group 2 and medicine D to group 2. The element of Group matrix below is 1 if and only if the object is assigned to that group.

$$G^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

2.2.4. Iteration-1, determine centroids : Knowing the members of each group, now we compute the new centroid of each group based on these new memberships. Group 1 only has one member thus the centroid remains in $c_1 = (1,1)$. Group 2 now has three members, thus the centroid is the average coordinate among the three members:

$$c_2 = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = \left(\frac{11}{3}, \frac{8}{3}\right)$$



2.2.5. Iteration-1, Objects-Centroids distances :

The next step is to compute the distance of all objects to the new centroids. Similar to step 2, we have distance matrix at iteration 1 is

$$D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (\frac{11}{3}, \frac{8}{3}) & group\ 2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \\ 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{matrix} \begin{matrix} | \\ | \\ | \end{matrix} \begin{matrix} X \\ Y \end{matrix}$$

2.2.6. Iteration-1, Objects clustering: Similar to step 3, we assign each object based on the minimum distance. Based on the new distance matrix, we move the medicine B to Group 1 while all the other objects remain. The Group matrix is shown below

$$G^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} group-1 \\ group-2 \end{matrix}$$

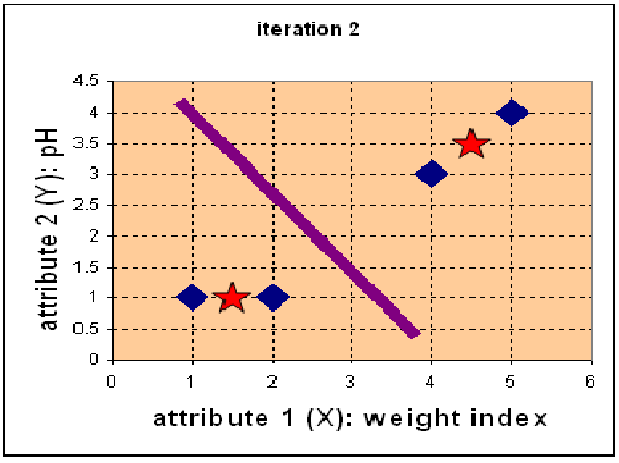
$$\begin{matrix} A & B & C & D \end{matrix}$$

2.2.7. Iteration 2, determine centroids: Now we repeat step 4 to calculate the new centroids coordinate based on the clustering of previous iteration. Group1 and group 2 both has two members, thus the new Centroids are

$$c_1 = (\frac{1+2}{2}, \frac{1+1}{2}) = (1\tfrac{1}{2}, 1)$$ and

$$c_2 = (\frac{4+5}{2}, \frac{3+4}{2}) = (4\tfrac{1}{2}, 3\tfrac{1}{2})$$

Object	Feature 1 (X): weight index	Feature 2 (Y): pH	Group (result)
Medicine A	1	1	1
Medicine B	2	1	1
Medicine C	4	3	2
Medicine D	5	4	2



2.2.8. Iteration-2, Objects-Centroids distances : Repeat step 2 again, we have new distance matrix at iteration 2 as

$$D^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{matrix} c_1 = (1\frac{1}{2}, 1) & group-1 \\ c_2 = (4\frac{1}{2}, 3\frac{1}{2}) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

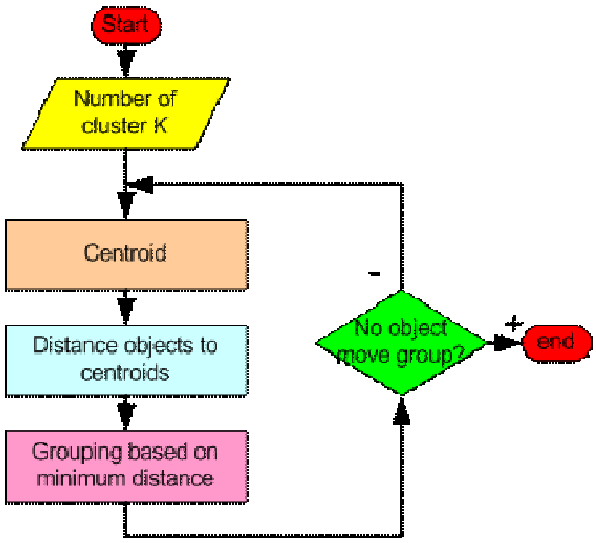
2.2.9. Iteration-2, Objects clustering: Again, we assign each object based on the minimum distance.

$$G^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

Its stability and no more iteration is needed. We get the final grouping as the results We obtain result that $G^2 = G^1$. Comparing the grouping of last iteration and this iteration reveals that the objects does not move group anymore. Thus, the computation of the k-mean clustering has reached.

2.3 Working of K-Mean Clustering algorithm:



Here is step by step k means clustering algorithm:

Step 1. Begin with a decision on the value of k = number of clusters.

Step 2. Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly, or systematically as the following:

Take the first k training sample as single-element clusters

Assign each of the remaining (N-k) training sample to the cluster with the nearest centroid. After each assignment, recomputed the centroid of the gaining cluster[2.4,5].

Step 3 . Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

Step 4 . Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

If the number of data is less than the number of cluster then we assign each data as the centroid of the cluster. Each centroid will have a cluster number. If the number of data is bigger than the number of cluster, for each data, we calculate the distance to all centroid and get the minimum distance. This data is said belong to the cluster that has minimum distance from this data[6].

Since we are not sure about the location of the centroid, we need to adjust the centroid location based on the current updated data. Then we assign all the data to this new centroid. This process

is repeated until no data is moving to another cluster anymore. Mathematically this loop can be proved to be convergent. The convergence will always occur if the following condition satisfied:

Each switch in step 2 the sum of distances from each training sample to that training sample's group centroid is decreased.

In our project we can implement three types of algorithms

1. Basic k-means algorithm
2. Benchmarked k-means algorithm
3. Concurent k-means algorithm

2.4 Basic K-Means Algorithm

- The letter k refers to the fact that the algorithm looks for a fixed number of clusters.
- The k-means algorithm is an algorithm to cluster objects based on attributes into k-partitions.
- It is one of the most commonly used clustering algorithms.
- It is also known as Mac Queen's algorithm because it is developed by the scientist "MAC QUEEN".

2.4.1 The Three steps of K-Means algorithm:

Step 1: The Algorithm randomly selects k data points to be the seeds .
Step 2: This step assigns each record to a closest seed.
Step 3: This step is to calculate the centroids of clusters.
The step2 is repeated and each point is once again assigned to the cluster with the closest Centroid[7].

2.4.2.Problem with Basic K-Means algorithm:

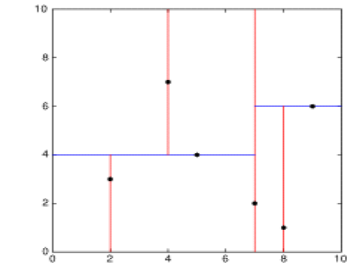
- However, this k-means method requires an execution time proportional to the product of the number of clusters and the number of patterns per iteration.
- The total execution time is computationally very expensive, especially for the large datasets.

2.5 Hierarchal K-Means

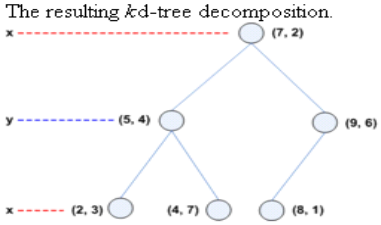
- The k-mean clustering algorithm cannot satisfy the need for fast response time for some applications.
- Hence to reduce the computational time required to cluster a large dataset becomes an important operational objective.
- To solve this and other related performance problems, proposed an algorithm based on the data structure k-d tree known as hierarchal k-means.
- And also it uses a pruning function on the centroid of a cluster.

2.5.1 K-D Tree

- A ***k*d-tree** (short for *k-dimensional tree)* is a space-partitioning data structure for organizing points in a *k*-dimensional space.
- *Kd-trees* are a special case of BSP trees. A *k*d-tree uses only splitting planes that are perpendicular to one of the co-ordinate axes.



The resulting *k*d-tree decomposition.



The resulting *k*d-tree.

Let us consider some data points in a given plane. As sho wn in the above figure there are six data points in the plane. We can form a K-D tree for those data points by considering their x co-ordinate and y co-ordinate. The resulting K-D tree is as shown in the above[8].

In this algorithm it also uses the same basic three steps of K-Means algorithm, but it internally uses the data structure called K-D tree. And this algorithm also displays the time required for each and every subtask.

2.5.2 Problem with Hierarchical K-Means Algorithm:

- While this method can reduce the number of distance calculations and the execution time.
- The time required to build a k-d tree is proportional to the size of the dataset.
- The total processing time is still long when a large dataset is involved.

2.6 Concurrent K-Means

In k-means clustering, we are given a set of n data points in d-dimensional space Rd and an integer k and the problem is to determine a set of k points in Rd, called centers, so as to minimize the mean squared distance from each data point to its nearest center[7,9,10].

A popular heuristic for k-means clustering is Lloyd's algorithm. In this paper we present a simple and efficient implementation of Lloyd's k-means clustering algorithm, which we call the filtering algorithm.

2.6.1 Functional Description

1. The K means algorithm will do the three steps below until convergence Iterate until *stable* (= no object move group):

    1. Determine the centroid coordinate
    2. Determine the distance of each object to the centroids
    3. Group the object based on minimum distance (find the closest centroid)

The algorithm consists of a simple re-estimation procedure as follows. Initially, the data points are assigned at random to the K sets. For step 1, the centroid is computed for each set. In step 2, every point is assigned to the cluster whose centroid is closest to that point. The initial centroid is calculated using a equation which divides the sample space for each dimension into equal parts depending upon the value of k.

These two steps are alternated until a stopping criterion is met, i.e., when there is no further change in the assignment of the data points. The basic approach is rather simple:

1. **Develop a single-threaded, sequential, robust, and clearly organized version of your algorithm.** If you're reading this article to speed up something that exists, you may already have this step covered.
2. **Identify the subtasks.** Examine the algorithm to identify the discrete stages. This shouldn't be difficult if you performed Step 1 adequately. Each subtask probably has its own method, or at least a clearly-delineated block of code. If you have difficulty identifying the subtasks, you probably need to understand the algorithm better or reorganize your code. In that case, return to Step 1.
3. **Benchmark the algorithm.** In other words, time the identified subtasks to determine the fraction of time consumed by each. This should be a trivial matter of adding some timing and output statements.
4. **Delegate the most time-consuming subtasks to a thread pool.** Now we come to the most difficult part: reorganizing your code so the slowest subtasks are performed concurrently by multiple subtask threads. Luckily, two utility classes in java.util.concurrent greatly simplify the process.

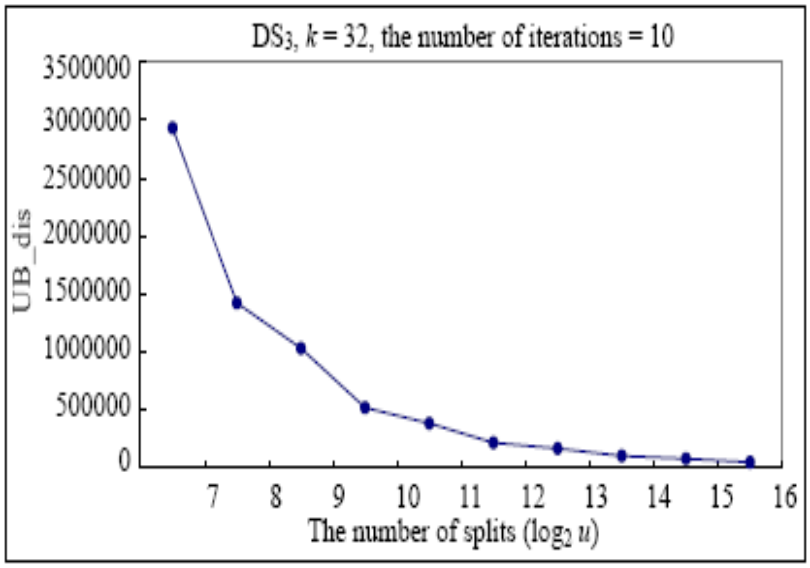
2.6.2 ANALYSIS OF UNIT BLOCK PARTITION

First, the proposed algorithm divides the dataset into several identical unit blocks. Then, it calculates the centroids and related statistics of patterns in each unit block to represent an approximation of the information in the unit blocks. We use this reduced data to reduce the overall time for distance calculations. We find that the clustering efficiency is closely related to determining how many blocks should be partitioned. This problem was investigated in the following.

There are *n* patterns in the dataset and *k* clusters were specified, and we use our algorithm to implement the *k*-means method. The performance from our experiments reveals some variation when the numbers of unit blocks increase or decrease, resulting in the following phenomenon:

1. When number of unit blocks increases, we find the number of unit blocks on cluster boundaries also increases, while the average number of patterns in unit blocks on the cluster boundary decreases. Therefore, the number of distance calculations that determines which patterns in the boundary blocks belong to which clusters will decrease with an increase in the number of unit blocks.

2. Conversely, when the number of unit blocks decreases, the number of unit blocks on the cluster boundary decreases as well, but the average number of patterns in unit blocks on the cluster boundary increases. Since the number of patterns belonging to these clusters increases, the number of distance calculations also increases, as shown in Fig.



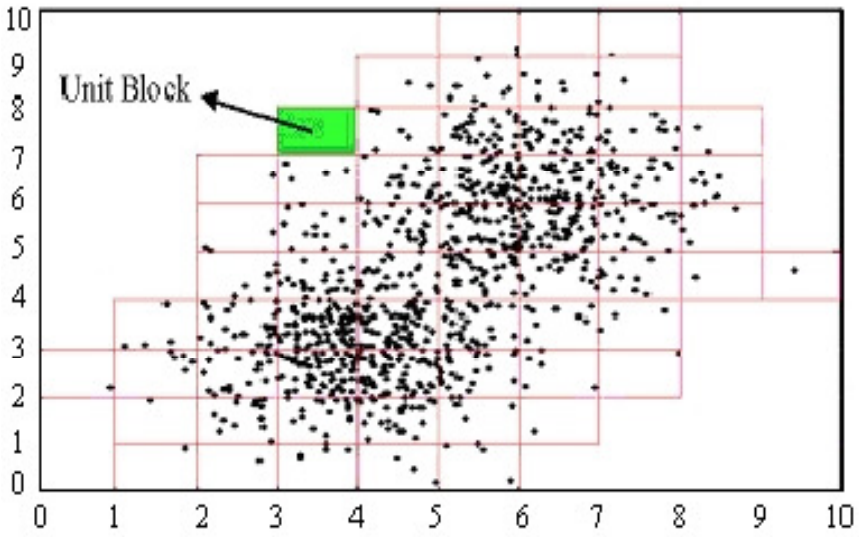
2.6.3 Partitioning the Dataset into Unit Blocks

At the beginning of the algorithm, the dataset is partitioned into several blocks of equal size. Instead of using the *k-d* tree approach, we employ a simple method that does not require more than two scans of the dataset.

Partition method for finding splitting points, for two dimensional data we determine the minimum and maximum values of the data along each dimension. This is the first scan of the data set. These values determine a rectangle that bounds all the dataset patterns[11].

Next, we partition this data space into equally sized blocks. Unlike in *k-d* tree partition, the midpoint approach is used to recursively divide the splitting dimensions into equal parts. We simply choose a fixed number of splits to produce a specified total number of blocks. This can save some computation time. In this empirical study, see that 11 splits resulted in near optimal performance for the datasets with a random distribution. After the partitioning, locate all the

blocks that contain at least one pattern and call them Unit Blocks (*UB*s) as shown in Fig. 1. To find out which *UB* a pattern belongs to, a second scan of the dataset is performed.



3. CONCLUSIONS

An appropriately designed symmetric Multi-Threading may lift the imposed restrictions and reduce the complexity of performance monitoring hardware. For example, if it is not technically possible to equip each logical processor within an SMP/IACKMA system with a dedicated performance monitoring unit, adequate results may be retrieved by employing the method of per-thread performance monitoring.

In this project presents an efficient clustering algorithm based on the *k*-means method. We partitioned datasets into several blocks and used reduced versions of the datasets to compute final cluster centroids. The results of this experiment clearly indicate that the algorithm converges quickly. When compared to two other commonly used clustering algorithms, the direct *k*-means algorithm, in terms of total execution time, the number of distance calculations, and the efficiency for clustering, this algorithm was superior.

Acknowledgment

The issue of the proper choice for the number of unit blocks. We found that it is independent of the number of clusters but is related to the distribution of the dataset. Presently, we use binary splitting to partition the dataset into unit blocks, but other partition methods will be the subject of further study especially for high dimensional datasets.

REFERENCES

[1] Billings, S.A., and Fadhil, M.B., 2004,“The practical identification of system with nonnearities”, Proc. 7th IFAC Symp. On Identification and System Parameter Estimation, York, U.K., 155-160.

[2] Cater, J.P., 2004, "Successfully using peak learning rates of 10 (and greater) in back propagation networks with the heuristic learning algorithm", in IEEE First Int. Conf. on Neural Networks (San Diego 2006), Caudill, M., and Butler, C. (eds.), II, 645-651, IEEE, New York.

[3] Chen, S., Billings, S.A. and Grant, P.M., 2004, "Recursive hybrid algorithm for nonlinear system identification using radial basis function networks", Int. J. of Control,55, 1051-1070.

[4] Cichocki, A., and Unbehauen, R., 2003, Neural Networks for Optimisation and Signal Processing, Wiley, Chichester.

[5] Darken, C., and Moody, J., 2007, "Fast adaptive k-means clustering: Some empirical results", Int. Joint Conf. on Neural Networks, 2, 233-238.

[6] Darken, C., and Moody, J., 2005, "Towards fast stochastic gradient search", In: Advance in neural information processing systems

[7] Franzini, M.A., 2007, "Speech recognition with back propagation", In Proc. of the Ninth Annual Conf. of the IEEE Eng. in Medicine and Biology Society, 1702- 1703, IEEE, New York.

[8] Hertz, J., Krogh, A. and Palmer R.G., 2004, Introduction to the theory of neural computation, Addison Wesley, New York.

[9] Ismail, M.A., and Selim, S.Z., 2004, "Fuzzy c-means: optimality of solutions and effective termination of the algorithm", Pattern Recognition, 19, 481-485.

[10] Jacobs, R.A., 2006, "Increased rates of convergence through learning rate adaptation", Neural etworks, 1, 295-307.

[11] Kamel, M.S., and Selim, S.Z., 2005, "New algorithms for solving the fuzzy clustering problem", Pattern Recognition, 27 (3), 421-428. 12. Lloyd, S.P., 1957, "Least squares quantization in PCM", Bell Laboratories Internal Technical Report, IEEE Trans. on Information Theory.

Authors Profiles

Dr.O.NagaRaju received the Masters Degree in Computer Science & Engineering from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. He awarded Ph.D in Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. His research interests include Software Engineering, Network Computing, and Data mining, Image Processing. His published several publications in international and national journals.

N.Surya Kalyan Chakravarthy obtained his Bachelor's degree in Computer Engineering from University of Madras in 1994 and M.Tech from Nagarjuna University in 2010. During the period from 1994 to 2010, he has been involved in various aspects of Information Technology - an engineer, an enterpreneur (IT people Inc.) as well as an educator (Founder of Sri Nidamanuri Educational Society consisting of schools of Engineering, Pharmacy and Sciences). Currently he wishes to conduct research in the area of Data Mining and Information Security .His research interests include software Engineering, Image Processing, Neural networks. He is a member of the IEEE and a member of the ACM.

Dr. R.A. Khan, Presently Working with Babasaheb Bhimrao Ambedkar University (A Central University) Lucknow, UP as an **Associate Professor&Head** in the Department of Information Technology, School for Information Science & Technology since December 2006. He is having More than **Ten Years** of teaching experience at PG and UG Level. **He obtained Master of Computer Application (M.C.A.)** from PTU, Jalandhar securing **73%** marks (**2000**) and **Ph.D.** from Jamia Millia Islamia (A Central University) New Delhi (**2004**).

Mr. Bonthu Kotaiah obtained his Bachelor's degree in Computer Applications from Nagarjuna University in 2001 and M.C.A from Nagarjuna University in 2008. During the period from September, 2001 to 2011, he has been involved in various aspects of Information Technology - an engineer(L-Cube Innovative Solutions), a Corporate Trainer (SyncSoft & Datapro(Vijayawada), COSS(Hyd.)), a Computer Programmer(Acharya Nagarjuna University). Currently he wishes to conduct research in the area of Software Engineering and Data Mining and Artifiicial Neural Networks, Fuzzy Logic & Genetic Alogorithms. His research interests include software Engineering, Neural networks. Presently, he is working as a Full-Time Research Scholar in Babasaheb Bhimrao Ambedkar University (A Central University) Lucknow, UP in the Department of Information Technology.