A NEW ALGORITHM FOR SELECTION OF BETTER K VALUE USING MODIFIED HILL CLIMBING IN K-MEANS ALGORITHM

[1]G KOMARASAMY, [2]AMITABH WAHI

[1]Department of Computer Science and Engineering,

[1]Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India

[2]Department of Information Technology,

[2]Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India

E-mail: [1]gkomarasamy@gmail.com , [2]awahi@rediffmail.com

ABSTRACT

The k-means algorithm is a popular clustering method for text and web collections. It gained its popularity due to its simplicity and intuition. The algorithm is an iteration procedure and requires that the number of clusters, k, be given a priori. This selection of k value itself is an issue and sometimes it is hard to predict before the number of clusters that would be there in data. This problem is resolved by using a metaheuristic method, the k means combined with Bat Algorithm (KMBA) that utilizes the echolocation behavior of bats. KMBA algorithm does not require the user is given in advance the number of cluster. However, this KMBA does not guarantee unique clustering because we get different results with randomly chosen initial clusters. The final cluster cancroids may not be the optimal ones as the algorithm can converge into local optimal solutions. So we blended k-means algorithm uses modified hill-climbing search to attempt to find the global optimal solution of the objective function. These Hill-climbing algorithms are iterative algorithms which make modifications that increase the value of their objective function at each and every step. The experimental result shows that proposed algorithm Modified Hill-climbing aided K-Means Algorithm (MHKMA) is better than the existing algorithm KMBA and ordinary k-means.

Keywords: *Bat Algorithm, Echolocation, Hill-Climbing Algorithm, Objective Function.*

1. INTRODUCTION

Clustering is an important tool for a variety of applications in data mining, data compression, statistical data analysis and vector quantization which is a process of grouping objects with similar properties. Any cluster should exhibit two main properties; low inter-class similarity and high intra-class similarity. Clustering algorithms are mainly divided into two categories: Hierarchical algorithm and Partition algorithm. A hierarchical clustering algorithm divides the given data set into smaller subsets in hierarchical fashion. A partition clustering algorithm partitions the data set into desired number of sets in a single step. Clustering has a long and rich history in a variety of scientific fields and many methods have been proposed to solve clustering problem. *k* -means is a well known prototype-based partitioning and clustering technique that attempts to find a user-specified number of clusters (*k*), which are represented by their centroids.

The following is the *k* -means algorithm steps:

1. Select initial centres of the K clusters. Then, repeat steps 2 through 3 until the cluster membership stabilizes.
2. Generate a new partition by assigning each data to its closest cluster centres.
3. Compute new cluster centres as the centroids of the clusters.

In *k*-means may appear simple and applicable for a wide variety of data types but it is quite sensitive to initial positions of cluster centres. The finally resulting cluster centroids may not be optimal ones as the algorithm can converge to local optimal solutions. The *k*-means algorithm can be run multiple times to reduce this effect. Even an empty cluster can be resulted, if no points are allocated to the cluster during the assignment step. Thus, it is quite important for *k*-means to have good initial cluster centres. The flow diagram for *k* -means is shown in Figure 1 and procedure of *k*-means clustering is shown in Figure 2**.**

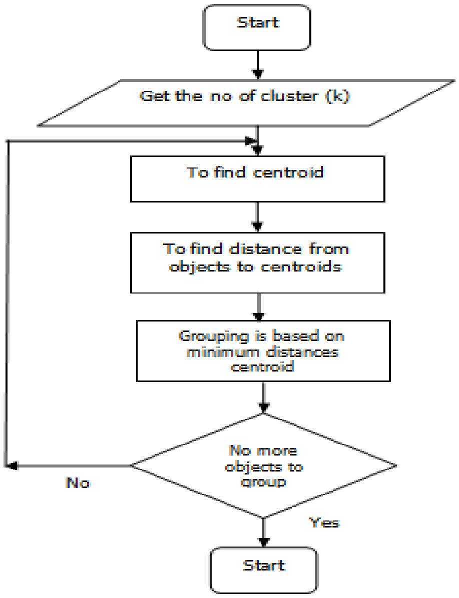
Start
Get the no of cluster (k)
To find centroid
To find distance from objects to centroids
Grouping is based on minimum distances centroid
No more objects to group
No
Yes
Start

Figure 1: K-Means Flowchart

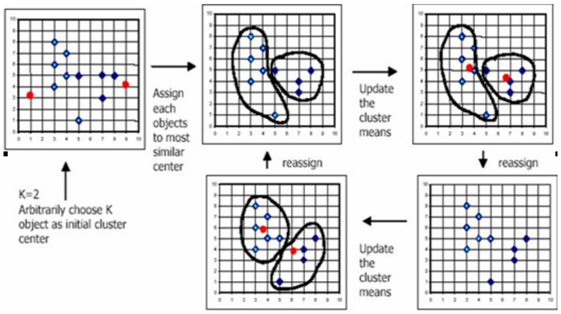
Assign each objects to most similar center
Update the cluster means
reassign
reassign
Update the cluster means
K=2
Arbitrarily choose K object as initial cluster center

Figure 2: Procedure Of K- Means Clustering

In the KMBA, each bat initialize the location clusters has two basic tasks: to search the best number of clusters by using a Bat algorithm and to find the best set of cluster centers by using the *k*-means according to the assigned number of clusters. But it is in need of *K* value which has direct impact on algorithm convergence or performance.

2. RELATED WORK

In [1] author proposed an effective algorithm to compute new cluster centers for each iterative step in the *k*-means clustering algorithm. This algorithm is based on the optimization formulation of the problem and a novel iterative method. The centers of cluster computed using this methodology are found to be very close to the desired cluster centers for iterative clustering algorithms.

Here, after center initialization, elements are assigned to the concerned clusters. When an empty cluster is found at this early stage, the process of re-initialization takes place and the process is repeated until all non-empty initial clusters are formed. The limitation is that this method are time costly and may not be applicable by keeping the *k*-means inherently simple structure and the advantages are based on the optimization formulation of the problem and also, a novel iterative method.

In [2] a data clustering approach using modified *k*-means algorithm based on the improvement of the sensitivity of initial center (seed point) of clusters is proposed. It partitions the whole space into different segments and calculates the frequency of data point in each segment. Obtained segment which shows maximum frequency of data point will have the maximum probability to contain the centroid of cluster. The number of cluster's centroid (k) will be provided by the user in the same manner like the traditional *k*-means algorithm and the number of division will be k*k (`k' vertically as well as `k' horizontally). The highest frequency of data point is same in different segments and the upper bound of segment crosses the threshold `k' then merging of different segments become mandatory and then take the highest k segment for calculating the initial centroid (seed point) of clusters. They also define a threshold distance for each cluster's centroid to compare the distance between data point and cluster's centroid with this threshold distance through which we can minimize the computational effort during calculation of distance between data point and cluster's centroid. Thus the modified *k*-means algorithm will decrease the complexity & the effort of numerical calculation by maintaining the easiness of implementing the *k*-means algorithm. This assigns the data point to their appropriate class or cluster more effectively. The advantage is that center and proportional weight difference is used as distance measure and issues are uncertainty, vagueness, and incompleteness.

The main purpose of the paper [3] is to presents a modified *k*-means algorithm with the intension of improving cluster quality and to fix the optimal number of cluster. The proposed method works for both the cases i.e. for known number of clusters in advance as well as unknown number of clusters. In addition to suggesting user has the flexibility either to fix the number of clusters or input the minimum number of clusters required. In the former case it works same as *k*-means algorithm. In the latter case the algorithm computes the new cluster centers by incrementing the cluster counter by one in each iteration until it satisfies the validity of cluster quality.

In this paper [4] select the number of clusters for the *k*-means algorithm has been proposed. This method can suggest multiple values of *k* to users for cases when different clustering results could be obtained with various required levels of detail. The method could be computationally expensive if used with large data sets because it requires several applications of the *k*-means algorithm before it can suggest a guide value for *k*.

Eigenvectors can be used in clustering analysis [5]; they are composed of time-domain statistical indexes. Centers of cluster can be attained by iterative computation with *k*-means algorithm. The basic principle of *k*-means algorithm is that squared sum between all samples in cluster domain and cluster centers should be minimum. Though *k*-means algorithm is simple and the fast converging, it also has some limitations. Simulated annealing algorithm is based on randomized searching algorithm and global optimization algorithm. While employing the optimization algorithm, the question of local minimum in *k*-means algorithm can be avoided. The resulting cluster of *k*-means algorithm is used as initial solution; as a result, the optimal cluster centers are attained by simulated annealing. The advantages are that clustering centers can be attained by iterative computation with *k*-means algorithm and each group of raw data is normalized with mean zero and de-noised with wavelet transform. The issue is lack of prior knowledge and thereby difficulty in classification.

In [6] the author described about the explosive growth of data from terabytes to peta bytes that need for following consideration:

- Data collection and availability
- Automated data collection tools, web, database systems, computerized society
- Major sources of abundant data
- Business: Web, transactions, e-commerce and stocks
- Science: Scientific simulation, bioinformatics and remote sensing, etc.
- Society and everyone: news, digital cameras, etc.

Performance of iterative clustering algorithms which converges to numerous local minima depends highly on initial cluster centers. Generally initial cluster centers are selected randomly. In this paper [7] author propose an algorithm to compute initial cluster center for *k*-means clustering and this algorithm is based on two observations that some of the patterns are very similar to each other and that is why they have same cluster membership irrespective to the choice of initial cluster centers. Also, an individual attribute may provide some information about initial cluster center. The initial cluster centers computed using this methodology are found to be very close to the desired cluster centers, for iterative clustering algorithms. It is applicable to clustering algorithms for continuous data. Though it is conceptually simple and quite straightforward to code, this algorithm complicated during implementation which turn out to be an issue.

Kernel *k*-means is an extension of the standard *k*-means clustering algorithm that identifies nonlinearly separable clusters. In turn to overcome the problem of cluster initialization associated with this method, here in this work [8] author propose the global kernel *k*-means algorithm which is a deterministic and incremental approach for kernel-based clustering. This method adds one cluster at each stage through a global search procedure consisting of several executions of kernel *k*-means from suitable initializations and this algorithm is independent of cluster initialization that identifies nonlinearly separable clusters and also due to its incremental nature and search procedure, it locates near optimal solutions avoiding poor local minima problem. In addition, a modification is also proposed to reduce the computational cost that does not significantly affect the quality of solution. The advantage is this algorithm reduces the computational cost and issue is difficulties in development of theoretical foundations as real implementation behind the assumptions of the method.

In [9] author surveyed clustering algorithms for data sets appearing in statistics, machine learning and computer science. And illustrate their applications in the traveling salesman problem, some benchmark data sets and bioinformatics which is a new field attracting intensive efforts. There are several tightly related topics, cluster validation, and proximity measures are also discussed over here. This survey focuses on clustering in data mining that adds to clustering the complications of very large datasets with very many attributes of different types that imposes unique computational requirements on relevant clustering algorithms. Several algorithms have recently emerged that meet these requirements and were successfully applied to real-life data mining problems which are subject of this survey. It represents the data by fewer clusters may necessarily lose certain fine details but however, achieves simplification.

In [10] author discusses the standard *k*-means clustering algorithm and analyzes the shortcomings of standard *k*-means algorithm as like the *k*-means clustering algorithm has to calculate the distance between each data object and all cluster centers in each of iteration, which makes the efficiency of clustering is not high. This paper proposes an improved *k*-means algorithm in order to solve the above mentioned problem that require a simple data structure to store some information in every iteration, which is to be used in the next iteration. The improved method avoids computing the distance of each data object to the cluster centers that saving the running time. Advantage is that it has reduced running time with the improved speed of clustering and accuracy.

3. MODIFIED HILL CLIMBING AIDED K-MEANS ALGORITHM (MHKMA)

This *k*-means algorithm aims at minimizing a squared error function is given in Equation (1) for the objective function.

$$J = \sum_{i=1}^{k}\sum_{i=1}^{n}\left\| x_i(j) - c_j \right\|^2 \qquad (1)$$

Where, $\left\| x_{j(j)} - C_j \right\|^2$ is a chosen distance measure between a data point $x_j^{(j)}$ and the cluster centre c_j is an indicator of the distance of the n data points from their respective cluster centers. One of the main disadvantages to KMBA [11] is that it requires the number of clusters as an input to the algorithm. The algorithm is incapable of determining the appropriate number of clusters and depends upon the user to identify this in beforehand. For example, if you had a group of people that were easily clustered based upon gender while calling the *k*-means algorithm with k=3 would force the people into three clusters and when k=2 would provide a more natural fit. Likewise, if a group of individuals were easily clustered based upon home state and you called the *k*-means algorithm with k=20 then the results might be too generalized to be effective.

For this reason, it's often a good idea to experiment with different values of k to identify the value that best suits your data. Suppose that we have n sample feature vectors x_1, x_2…x_n all from the same class and we know that they fall into k compact clusters, $k < n$. Let m_i be the mean of the vectors in cluster *i*. We can use a minimum distance classifier to separate them, if the clusters are well separated, i.e.., we can say that *x* is in cluster (*i*) if $\|x - m_i\|$ is the minimum of all the *k* distances.
The following procedure is suggested for finding the k means:
Make initial guesses for the means $m_1, m_2, \ldots. m_{k..}$
Until there are no changes in any mean.

1. Use the estimated means to classify the samples into clusters.
2. For *i* from *1 to k*, replace m_i with the mean of all of the samples for cluster *i.*

Else end.

But finding the value of *i* that best suits of data is very difficult. Hence we moved on to hill climbing. Hill climbing is good for finding a local optimum (a good solution that lies relatively near the initial solution) but it is not guaranteed to find the best possible solution (global optimum) out of all possible solutions (search space) which can be overcome by using steepest ascent Modified Hill climbing finds globally optimal solution. The relative simplicity of the algorithm makes it a popular first choice amongst optimizing algorithms and it is widely used in artificial intelligence, in order to reach a good state from a start state. Selection of next node and starting node can be varied to give a list of related algorithms. This can often produce a better result than other algorithms when the amount of time available to perform a search is limited, such as with real-time systems.

Modified Hill climbing algorithm attempts to maximize (or minimize) a target function $f(x)$ where x is a vector of continuous and / or discrete values. In each iteration, hill climbing will adjust a single element in x and determine whether the change improves the value of $f(x)$. Then, x is said to be globally optimal.

3.1 Modified Hill climbing aided k-Means Algorithm

The Modified Hill Climbing aided *k*-means Algorithm steps are shown bellow.

Input: *randk* - random value of *k*
Δk - A random move in cluster
Output: *k* - Number of clusters

Pseudo code: Modified Hill Climbing Algorithm

```
do
l_1: iter =true;
        $k_{solved} \leftarrow randk$;
        $l_2$: $new_{solution} \leftarrow k_{solved} + \Delta k$;
         if $(f (new_{solution}) < f (k_{solved}$ ) then
                solution $\leftarrow new_{solution}$;
                $k_{solved} \leftarrow$ solution;
                $k \leftarrow k_{solved}$;
                if(algorithm converged and
                globally optimum) then
                        output k;
                        iter = false;
                else
                        goto $l_2$ ;
        else
                goto $l_1$ ;
while (iter);
```

Input: $E= \{ e_1, e_2 \ldots e_n \}$ - Set of entities to be clustered

k - number of cluster from Modified Hill Climbing Algorithm

MaxIters - Limit of iterations

Output: C= $\{c_1, c_2 \ldots c_n \}$ - Set of clustered centroids

L= {l (e) e= {1, 2…n} - Set of cluster labels of E

Pseudocode: Modified Hill Climbing aided *k*-means Algorithm

```
for each $c_i \in C$ do
        $c_i \leftarrow e_j \in E$ (E.g. random selection);
end
for each $e_i \in E$ do
    $L(e_i) \leftarrow argminDistance (e_i, c_i) j \in \{1,\ldots, k\}$;
end
changed ← false;
iter ← 0;
repeat
for each $c_i \in C$ do
        Update cluster ($c_i$);
end
for each $e_i \in E$ do
     $minDist \leftarrow argminDistance (e_i, c_j) j \in \{1 \ldots k\}$;
if $minDist \neq l (e_i)$ then;
        $l(e_i) \leftarrow minDist$;
        changed ← true;
end
end
iter ← iter+1;
until changed=true and iter $\leq$ MaxIters;
```

In the above algorithm is the best K value is obtained by modified hill climbing and this value is utilized in *k*– means algorithm in order to form effective clusters with uniform cluster density. The following section deals with performance evaluation of implemented system.

4. EXPERIMENTAL RESULTS

We analyze and compare the performance offered by *k*-means clustering algorithm and MHKMA. Here when the no of datasets increased, the error rate is decreased linearly and the dataset which we used for experimentation is Iris, Wine and Vehicle datasets. The error rate of the proposed *k*-means with bat algorithm is low. Based on the comparison and the results from the experiment show the proposed approach works better than the other existing systems. The precision is defined as the proportion of the predicted positive cases in clustering that were correct. The following (from Figure 3 to Figure 5) are the precision graph of Iris, Wine and Vehicle dataset respectively that shows MHKMA is having higher precision than the other techniques.

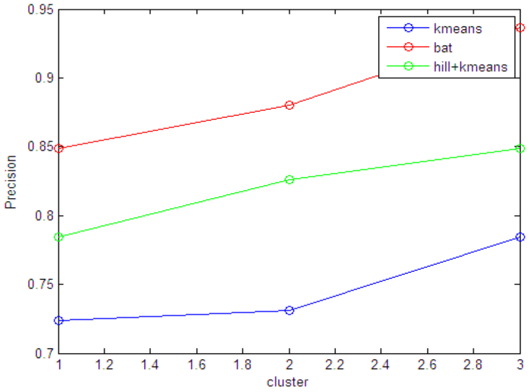

Figure 3: Precision Graph Of Iris Dataset

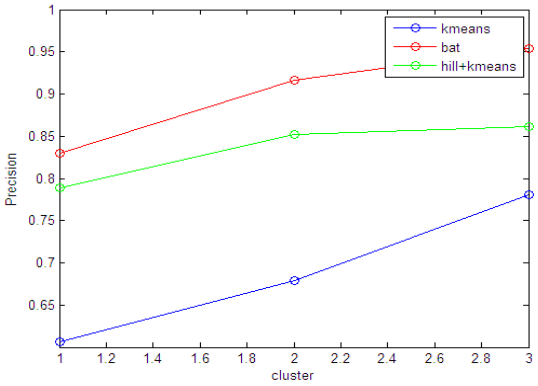

Figure 4: Precision Graph Of Wine Dataset

The sensitivity or true positive rate or recall is defined as the proportion of the actual positive cases in clustering, which were correct. The following (from Figure 6 to Figure 8) are the sensitivity graph of Iris, Wine and Vehicle dataset respectively that shows MHKMA is having higher sensitivity than the other techniques.

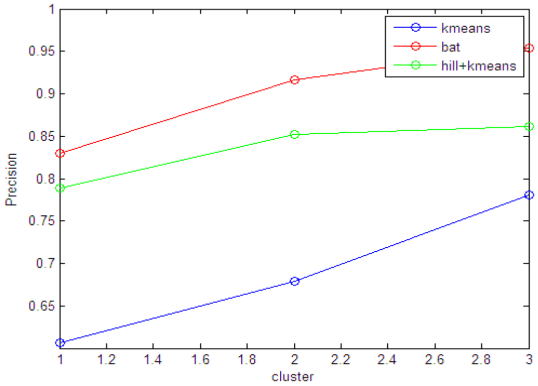

Figure 5: Precision Graph Of Vehicle Dataset

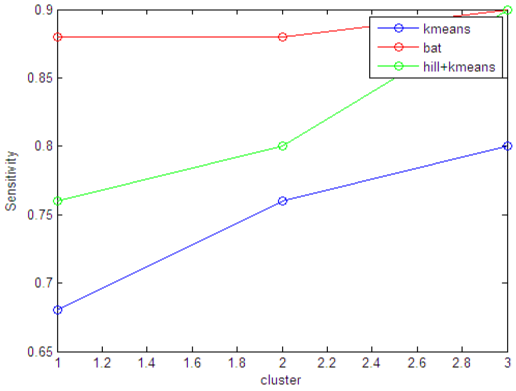

Figure 6: Sensitivity Of Iris Dataset

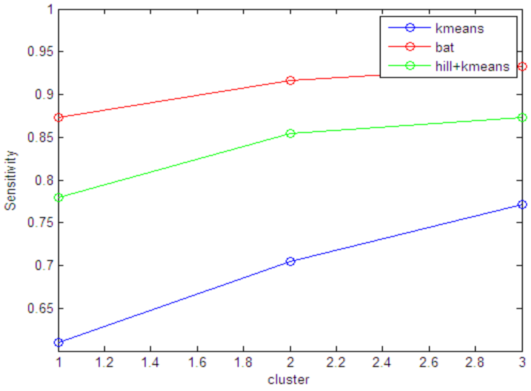

Figure 7: Sensitivity Of Wine Dataset

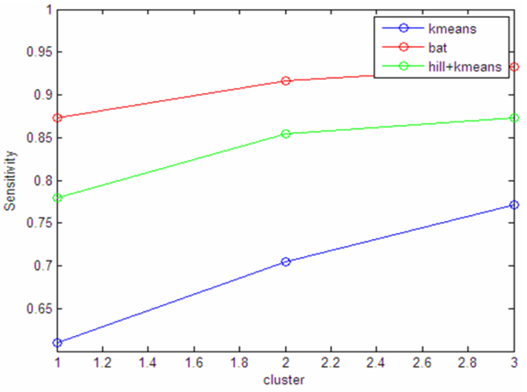

Figure 8: Sensitivity of Vehicle Dataset

The specificity is defined as the proportion of the negative cases in clustering, which were correct. The following (from Figure 9 to Figure 11) are the specificity graph of Iris, Wine and Vehicle dataset respectively that shows MHKMA is having higher specificity than the other techniques.

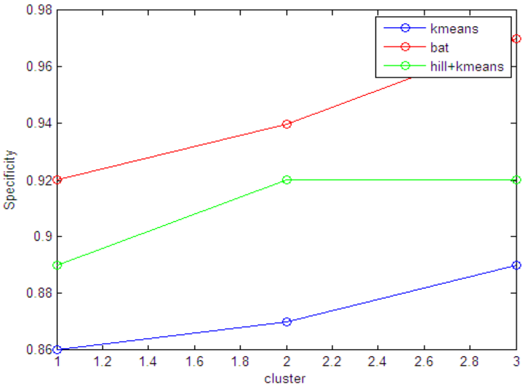

Figure 9: Specificity of Iris Dataset

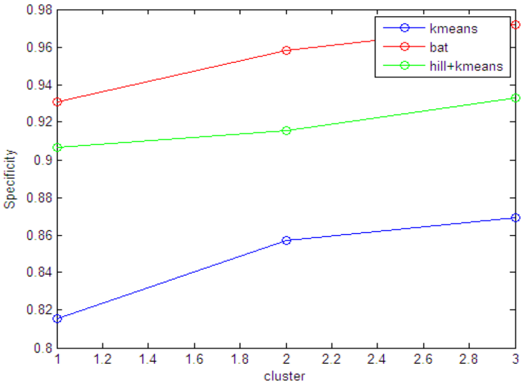

Figure 10: Specificity of Wine Dataset

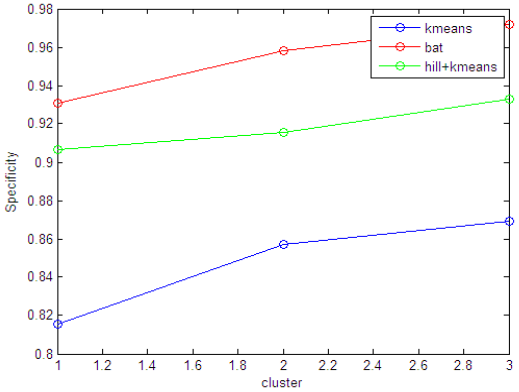

Figure 11: Specificity of Vehicle Dataset

The following Table 1, Table 2 and Table 3 shows the proposed algorithm compare with existing algorithms measurements precision, sensitivity and specificity respectively. In these tables all the three datasets (iris, wine and vehicle) from UCI repository proposed algorithm gives the better results.

Table 1: Comparison Of Precision

S. No	Algorithm / Dataset	Precision		
		Iris	Wine	Vehicle
1.	*k*-means algorithm	0.743	0.69	0.69
2.	Bat algorithm	0.82	0.833	0.833
3.	*k*-means + Hill-climbing	0.853	0.9	0.9

Table 2: Comparison Of Sensitivity

S. No	Algorithm / Dataset	Precision		
		Iris	Wine	Vehicle
1.	*k*-means algorithm	0.746	0.693	0.693
2.	Bat algorithm	0.82	0.833	0.833
3.	*k*-means + Hill-climbing	0.863	0.906	0.906

Table 3: Comparison Of Specificity

S. No	Algorithm / Dataset	Precision		
		Iris	Wine	Vehicle
1.	*k*-means algorithm	0.873	0.85	0.85
2.	Bat algorithm	0.91	0.92	0.92
3.	*k*-means + Hill-climbing	0.886	0.953	0.953

The bellow chart (Figure 12 to Figure 14) shows the precision, sensitivity and specificity respectively. From this charts it shows proposed algorithm gives the best result. All the experimental results we have used Matlab 7.8 version.

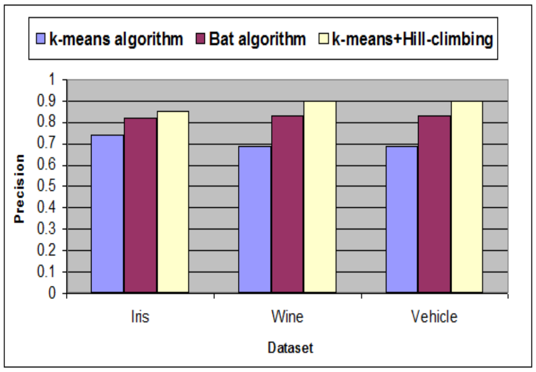

Figure 11: Precision For 3-Datasets

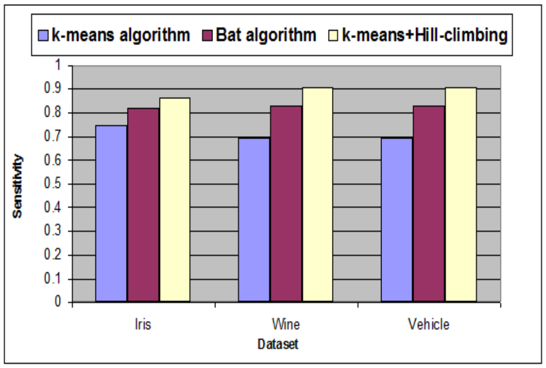

Figure 12: Sensitivity For 3-Datasets

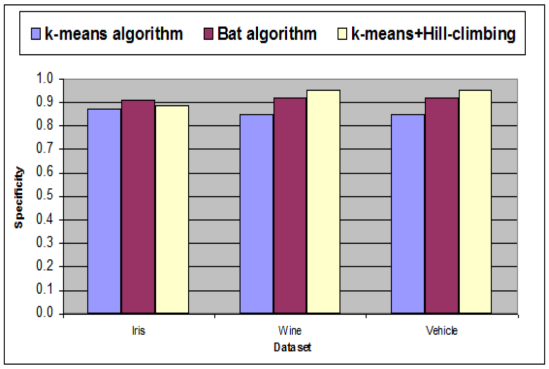

Figure 13: Specificity For 3-Datasets

5. CONCLUSION & FUTURE WORK

The *k*-means clustering algorithm is one of the partition clustering algorithms that primarily depends on two factors namely initial clusters and *k* value. Basically in *k*-means clustering, initial clusters are based on algorithm randomly selected centroids and randomly chosen *k* values. In the existing work of KMBA, the problem of initial centroid is solved. However, the value of *k* is a matter of fact that is not yet considered. Hence we propose *k*-means algorithm that uses modified hill climbing algorithm to resolve before mentioned problem. We evaluated the performance of ordinary *k*-means with KMBA and MHKMA, from which we can find the proposed algorithm, is more effective in terms of reduced number of iterations with equal cluster density in all clusters reducing running time complexity of the deployment. The experimental result shows the better clustering results than the other two algorithms. Though we say the proposed system reduces number of iterations. It may also lead to a pitfall of hill climbing called plateau where results of some plateau level values of *k* produces same results. The algorithm prematurely converge leaving the best *k* value unnoticed but only selecting the better *k* value and effective solution need to be provided in future regarding this problem.

REFRENCES:

[1] Wei Li, “Modified *k*-means clustering algorithm”, *IEEE computer society Congress on Image and Signal Processing*,2008, pp. 618-621.

[2] Ran Vijay Singh and M.P.S Bhatia, “Data Clustering with Modified *k*-means Algorithm”, *IEEE International Conference on Recent Trends in Information Technology*, ICRTIT 2011, 2011, pp 717-721.

[3] Ahamed Shafeeq B M and Hareesha K S “Dynamic Clustering of Data with Modified *k*-Means Algorithm” *International Conference on Information and Computer Networks*, ICICN 2012, pp 221-225.

[4] D T Pham, S S Dimov, and C D Nguyen “Selection of *k* in *k*-means clustering”, *Mechanical Engineering Science*, 2004, pp. 103-119.

[5] Ye Yingchun, Zhang Laibin, Liang Wei, Yu Dongliang , and Wang Zhaohui, “Oil Pipeline Work Conditions Clustering Based on Simulated Annealing *k*-means algorithm”, *World Congress on Computer Science and Information Engineering*, 2009, pp. 646-650.

[6] Jiawei Han and Micheline Kamber, “Data Mining: Concepts and Techniques”, *Morgan Kaufmann Publishers*, second Edition, 2006.

[7] Khan, S.S., Ahmad, A., “Cluster center initialization algorithm for *k*-means clustering”, *Pattern Recognition Letter*. 25, 2004, pp. 1293–1302.

[8] Grigorios F. Tztzis and Aristidis C. Likas, “The Global Kernel *k*-means Algorithm for Clustering in Feature Space”, *IEEE Trans. On Neural Networks*, Vol. 20, No. 7, July 2009, pp. 1181-1194.

[9] R. Xu and D. Wunsch, II, “Survey of clustering algorithms”, *IEEE Trans. Neural Networks.*, vol. 16, no. 3, 2005, pp. 645– 678.

[10] Shi Na., Liu Xumin, Guan Yon , "Research on *k*-means Clustering Algorithm: An Improved *k*-means Clustering Algorithm", *Third International Symposium on Intelligent Information Technology and Security Informatics(IITSI)*, April 2010, pp.63-67.

[11] Komarasamy G and Amitabh Wahi, “An Optimized *k*-means Clustering Technique Using Bat Algorithm”, *European Journal of Scientific Research*, ISSN 1450-216X Vol.84 No.2, August 2012, pp.263 – 273.