Comparison of Major Clustering Algorithms Using Weka Tool

R.P.T.H. Gunasekara[1] ,M.C. Wijegunasekara[2], N.G.J. Dias[3]
[1]Department of Computing and Information Systems, Wayamba University of Sri Lanka.
[1]hansigunasekara@yahoo.com
[2]Department of Statistics &Computer Science, University of Kelaniya.
[2]carmel@kln.ac.lk
[3]ngjdias@kln.ac.lk

I INTRODUCTION

Clustering algorithms are used in wide varieties of fields in many contexts. In these cases the behavior of the datasets are different to each other. Their sizes, density or the distribution may vary from one another. In data mining, clustering algorithms are implemented to build clusters with respect to a given dataset. But it is not an easy task to find the most suitable clustering algorithm for the given dataset. Therefore this study is done on several datasets using four clustering algorithms to identify the most suitable algorithm.This study is based on comparison of clustering data mining algorithms by using WEKA machine learning software.

II MATERIALS AND METHODS

This paper analyze the four major clustering algorithms: *k*-means, Expectation Maximization(EM), *makeDensityBased(mDB)* and Hierarchical(H) clustering algorithm and compare the performances of these major clustering algorithms on the aspect of cluster building ability of each algorithm. The results are tested using five datasets of previous experiments on Breast-cancer, mushroom, diabetics, iris and glass datasets using WEKA interface. The widely used machine learning and data mining software, namely WEKA, was first chosen to analyze clusters and compare the performance of each clustering algorithm for several datasets mentioned above to identify the most suitable clustering algorithm.

TABLE I
RESULTS OF EXPERIMENT

	EM	H	*mDB*	*k*-means
Cluster instances	56% 44%	35% 65%	64% 36%	65% 35%
Number of clusters	2	2	2	2
Number of iterations	4	1	4	4
Time taken build clusters (seconds)	0.08	3.93	0.05	0.03
within cluster sumof squared error	-	-	149.25	149.25
log likelihood	-2.16	-	-30.21	-

The results on diabetes dataset which contain 768 record and 9 attributes is recorded in TABLE I. The experiment was repeated for five datasets. All 20 results for five datasets and four algorithms were recorded.

III DISCUSSION AND CONCLUSION

From the 20 results obtained in the experiment it is possible to conclude that there are both advantages and disadvantages among algorithms.

The *k*-means was significantly reflected that it is the best performing algorithm for large datasets like mushroom dataset and the cluster building time taken was only 1.34 seconds. In *k*-means, Hierarchical and EM clustering algorithms user has to decide the number of clusters. But in *makeDensityBased* no need to decide the number of clusters by the user and suitable set of clusters are automatically created. Density based clustering algorithm is not suitable for data with high variance in density. Hierarchical Clustering algorithm didn't support for large datasets because it did not work for mushroom dataset. But Hierarchicalclustering algorithm is more sensitive for noisy datasets. By considering cluster instances created in the above table in breast cancer and glass datasets using Hierarchical clustering algorithm clusters were created for one or two data in the dataset.

EM clustering algorithm gives log likelihood values of the clusters to ensure more reliable clusters. EM algorithm is an extension of *k*-means algorithm which satisfying the formed clusters from *k*-means algorithm by maximizing the expectation using more iterations. Although this is a complex algorithm, it can be developed to parallelization for best performances using cross validation.

For large dataset it is better to select *k*-means algorithm or *makeDensityBased* algorithm. If the dataset is consisting of more noisy data, the most suitable algorithm is hierarchical clustering algorithm. The density based clustering algorithm is more suitable data with low variance.

This study is continued for several clustering algorithms to increase the performance by using parallel programming methodologies.