ISSN : 2231-0711

Density Based Clustering using Enhanced KD Tree

Logamani. K[#1], Punitha. S. C[#2]

[#1] *Research Scholar*

Department of Computer Science, PSGR Krishnammal College for Women
Coimbatore, India

logamanik@gmail.com

[#2] *Assistant Professor, HOD*

Department of Computer Science, PSGR Krishnammal College for Women
Coimbatore, India

***Abstract*— Clustering is a division of data into groups of similar objects. Each group called cluster, consists of objects that are similar within the cluster and dissimilar to objects of other clusters. Representing data by fewer clusters necessarily loses certain fine details, but achieves simplification, and so may be considered as a form of data compression. It represents many data objects by few clusters models data by its clusters. The system cannot cluster data sets with large difference in densities since the Mints-epsilon combination cannot be chosen appropriately for all clusters. These disadvantages are overcome in the proposed work by using two global parameters. Epsilon which is used to determine the maximum radius of the point neighbourhood, and midpoints which is used to determine the minimum number of points in an Epsneighbourhood, these two parameters set for each cluster depending upon the structure to be created. These are set automatically depending upon structure points created the kd-tree was combined with another kd-tree in order to boost the speed rate of the kd-tree.**

***Keywords*— Data mining, Clustering, Fast K- Means, K-Medoids, Enhanced KD Tree.**

I. Introduction

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics. Cluster analysis itself is not one specific algorithm, but the general task to be solved.

It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem.

The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It will often be necessary to modify data preprocessing and model parameters until the result achieves the desired properties.

The notion of a cluster cannot be precisely defined, which is one of the reasons why there are so many clustering algorithms. There of course is a common denominator: a group of data objects. However, different researchers employ different cluster models, and for each of these cluster models again different algorithms can be given. The notion of a cluster, as found by different algorithms, varies significantly in its properties. Understanding these "cluster models" is key to understanding the differences between the various algorithms.

II. Related Works

In K-Nearest Neighbour Based Spatial Clustering Using kd-tree, Data set within the same cluster share common features that give each cluster its characteristics. In this paper, an implementation of Approximate KNN-based spatial clustering algorithm using the kd-tree is proposed. The major contribution achieved by this research is the use of the kd-tree data structure for spatial clustering, and comparing its performance to the brute-force approach. The results of the work performed in this paper revealed better performance using the kd-tree, compared to the traditional brute-force approach.

In A Parallel Algorithm for Fast Density-based Clustering in Large Spatial Databases, Clustering the grouping of objects depending on their spatial proximity is one important technique of knowledge discovery in spatial databases. One of the proposed algorithms for this is FDC [5], which uses a density-based clustering approach. Since there is a need for parallel processing in very large databases to distribute resource allocation, this paper presents PFDC, a parallel version of FDC. The algorithm is divided into three parts. In the first stage, each processor essentially constructs a kd-tree on its part of the database. This is necessary because it is assumed that not all points loaded by a processor can be clustered at the same time in its main memory.

In An Improvement In The Build Algorithm For Kd-tree Using Mathematical Mean, In this research 4 tree data structures have been proposed. These are B, KD, Range and Quad trees. Made a comparative study of their build processes. Traditionally, the median of data elements is used forthe building of kd trees.Evaluated the build process of B, Range and Quad trees with their asymptotic time complexities. Then proposed a mean based build process for kd-tree which has been shown to have a better time complexity as well as a balanced tree compared to the original median based build process. This would widen the application domain for kd-tree, especially in the areas which require a balanced tree such as spatial database indexing.

III. DENSITY BASED CLUSTERING

The dataset consist of many fields which include numeric, strings. First the dataset is preprocessed to get the normalized values. The strings are removed using appropriate functions, finally the input will consist of values in the range between 0 to 1 and it will contain the attributes to which cluster it belongs to. First the tree structure points it constructed and after it are passed to build the tree. After the tree structure is built, the clustering is done ,the clustering concept is based on the nearest neighbour concept. Then analysis of the clustering is made by analyzing it with the query result.

Enhanced kd -tree for checking the clustering point of the connected data of each cluster within the region. Enhanced kd-tree to finding the structured point for cluster data of dense regions in dimensional space. Using Enhanced kd-tree will reduce speed up time and result in higher efficiency. Selected some points of Enhanced kd-tree which denote the dense centers of dense regions in the data set. Selecting leaf nodes and is done based upon the number of points created and number of structured points which represents higher clustering point data .

A. Normalization

The process of normalization is carried out in order the get the value range in between 0 and 1 in order to make the speed up process higher. The normalization process the strings are all removed using string comparison process and the data which are not relevant are also removes. Sufficient information and data are not fully filled.

B. Clustering using K-means Algorithm

K-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as bary centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

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

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centers.

C. Clustering using K-medoids algorithm

The k-medoidsis a clustering algorithm related to the k-means algorithm and the medoidshift algorithm. Both the k-means and k-medoids algorithms are partitional (breaking the dataset up into groups) and both attempt to minimize the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the k-means algorithm, k-medoids chooses data points as centers (medoids or exemplars) and works with an arbitrary matrix of distances between data points instead of l2. This method was proposed in 1987for the work with l1 norm and other distances.k-medoid is a classical partitioning technique of clustering that clusters the data set of n objects into k clusters known a priori.

D. Clustering using Kd-tree Algorithm

(Figures and tables must be centered in the column. Large figures and tables may span across both columns. Any table or figure that takes up more than 1 column width must be positioned either at the top or at the bottom of the page.)

A kd-tree, or k-dimensional tree, is a data structure used in computer science for organizing some number of points in a space with k dimensions. It is a binary search tree with other constraints imposed on it. Kd-treeare very useful for range and nearest neighbor searches. Generally only be dealing with point clouds in three dimensions, so all of kd-tree will be three-dimensional. Each level of a kd-tree splits all children along a specific dimension, using a hyper plane that is perpendicular to the corresponding axis. At the root of the tree all children will be split based on the first dimension (i.e. if the first dimension coordinate is less than the root it will be in the left-sub tree and if it is greater than the root it will obviously be in the right sub-tree). Each level down in the tree divides on the next dimension, returning to the first dimension once all others have been exhausted. They most efficient way to build a kd-tree is to use a partition method like the one Quick Sort uses to place the median point at the root and everything with a smaller one dimensional value to the left and larger to the right.

IV. Experiments and Results

A. *Working Environment*

Various experiments have been carried out by implementing clustering algorithms such as k-means, k-medoids and kd-tree. These algorithms are implemented usingMATLAB 2012. The results of the experiments are compared using entropy, correlation co-efficient,varience and modified-MSE.

B. *Dataset*

The dataset consist of many fields which include numeric, strings. First the dataset is preprocessed to get the normalized values. The strings are removed using appropriate functions, finally the input will consist of value.

C. *Results*

To evaluate this proposed system, the following measure are used they areCorrelation co-efficient, Variance, Modified_MSE, Entropy-class.

*1) Corelation Co-efficient***:** Correlation coefficient formulas are used to find how strong a relationship is between data. The formulas return a value between -1 and 1,where:

- 1 indicates a strong positive relationship.
- -1 indicates a strong negative relationship.
- A result of zero indicates no relationship at all.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

The following Fig. 1 shows the comparison chart of correlation co-efficient.

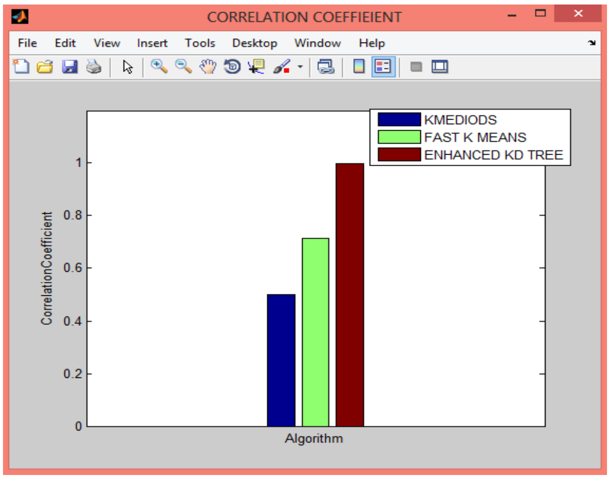

Fig. 1 Comparison chart for Correlation Coefficient

2) *Variance:* Variance measures how far a set of numbers is spread out. A variance of zero indicates that all the values are identical. Variance is always non-negative: a small variance indicates that the data points tend to be very close to the mean (expected value) and hence to each other, while a high variance indicates that the data points are very spread out around the mean and from each other.

An equivalent measure is the square root of the variance, called the standard deviation. The standard deviation has the same dimension as the data, and hence is comparable to deviations from the mean.

$$\text{Mean} = \frac{\Sigma fx}{\Sigma f}$$

$$\text{Variance, } \sigma^2 = \frac{\Sigma fx^2}{\Sigma f} - \left(\frac{\Sigma fx}{\Sigma f}\right)^2$$

The following Fig. 2 shows the comparison chart of variance.

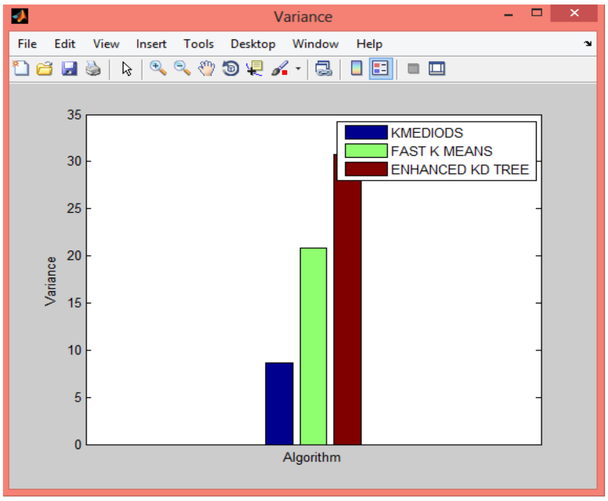

Fig. 2 Comparison chart for variance

3) Entropy –class: Entropy measures how the various semantic classes are distributed within each cluster. Given a particular cluster Sr of size nr, the entropy of this cluster is defined to be

$$\text{entropy} = \sum_{r=1}^{k} \frac{n_r}{n} E(S_r)$$

The following Fig. 3 shows the comparison chart of Entropy- class.

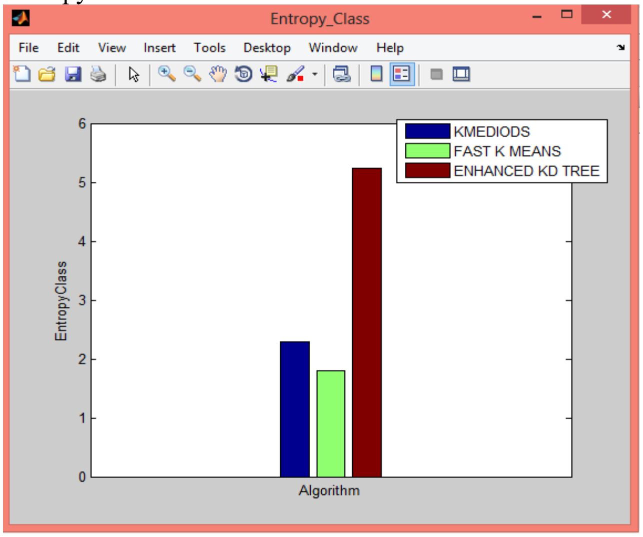

Fig. 3 Comparison chart for Entropy-class

4) Modified -MSE

The mean squared error assesses the quality of an estimator or set of prediction in terms of its variation and degree of bias. Suppose we have a random sample of size n from a population, $X_1, \ldots, X_n$. Suppose the sample units were chosen with replacement. That is, the n units are selected one at a time, and previously selected units are still eligible for selection for all n draws. The usual estimator for the mean is the sample average

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

which has an expected value equal to the true mean μ (so it is unbiased) and a mean square error of ,

$$\mathrm{MSE}(\bar{x}) = \mathrm{E}((\bar{x}-\mu)^2) = (\frac{\sigma}{\sqrt{n}})^2 = \frac{\sigma^2}{n}$$

where σ^2 is the population variance.

For a Gaussian distribution this is the best unbiased estimator (that is, it has the lowest MSE among all unbiased estimators) but not say for a uniform distribution.

The following Fig. 3 shows the comparison chart of Modified- mse.

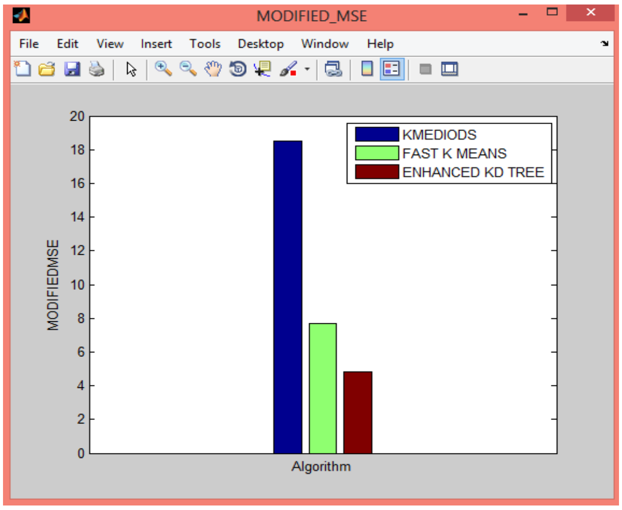

Fig. 4 Comparison chart for Modified- mse

V. CONCLUSION

In this paper, it is concluded that Enhanced kd tree demonstrate better performance than k-means and k-medoids. The results of both k- mean and k-medoid clustering algorithms the comparative study of both clustering algorithm are shown according to the number of clusters formed according to number of links and execution time taken by both clustering algorithm to make the clusters .An essential problem in data clustering and presented some solutions for it. It developed clustering algorithm by using Enhanced kd-tree for clustering.

In this paper, clustering data using the structured point and then using nearest neighbour. It is used kd-tree algorithm which incurs while reading the data points and also to cluster core points and based on the number of nearest neighbours found as well as find the nearest core point for a analyzing query data. The paper deals with enhanced kd-tree,one main thing in enhanced kd-tree is the high tree construction time. To overcome this issue a new technique must be used in order to overcome this issue in tree construction part.

REFERENCES

[1] ArunK.Pujari, “Data Mining Techniques”, Universities of Press,2010.

[2] AtulLuthra, “ECG Made Easy”,Japee Brothers Publishers, 2007.

[3] Jaiwei Han Micheline Kamber, “Datamining –Concepts and Techniques”, Elsevier Science India, 2001.

[4] G. Ball and D. Hall, “A Clustering Technique for Summarizing Multivariate Data,” Behavioural Science, vol. 12, pp. 153-155, 1967

[5] Dr.TVelmurugan,“EfficiencyofkMeanandMedoidsAlgorithmsforArbitraryData Points”,Int.J.Computer Technology &Applications,Vol 3 (5), 1758-1764

[6] V.Chitraa, Dr. Antony Selvadoss Thanamani, “An Enhanced Clustering Technique for Web Usage Mining,International“ , Journal of Engineering Research & Technology (IJERT)Vol. 1 Issue 4, June - 2012.

[7] RadhikaKyadagiri ,Prof. D. Jamuna ,Masthan Mohammed, “An Efficient Density based Improved K- Medoids Clustering

[8] algorithm” ,International Journal of Computers and Distributed Systems Vol. No.2, Issue 1, December 2012 [4].R. Suguna,D.

[9] T. Velmurugan and T. Santhanam, “Computational Complexity between K-Means and K-Medoids Clustering Algorithms for Normal and Uniform Distributions of Data Points”, Journal of Computer Science 6 (3): 363-368, 2010. “PDCA12-70 data sheet,” Opto Speed SA, Mezzovico, Switzerland.

[10] A. P. Reynolds, G. Richards, and V. J. Rayward-Smith ,“The Application of K-medoids and PAM to the Clustering of Rules”

[11] Hae-Sang Park, Jong-Seok Lee and Chi-HyuckJun ,“ A K-means-like Algorithm for K-medoids Clustering and Its Performance”.

[12] Warren Hunt, Gordon Stoll, andWilliam Mark. Fast kd-tree Construction with an Adaptive Error-Bounded Heuristic. In Proceedings of the2006 IEEE Symposium on Interactive Ray Tracing, 2006.

[13] La´aszl´oSz´ecsi. An Effective Implementation of the kd-Tree. In Jeff Lander, editor, Graphics Programming Methods, pages 315–326.Charles River Media, 2003.

[14] Pravin M. Vaidya. An O(N log N) Algorithm for the All-Nearest-Neighbors Problem. Discrete and Computational Geometry, (4):101–115, 1989.

[15] Constrained K-means Clustering with Background Knowledge ;KiriWagsta, Claire Cardie, Seth Rogers, Stefan Schroedl, Daimler Chrysler In: Proceedings of the Eighteenth International Conference on Machine Learning, 2001, p. 577-584,

[16] Fast kd-Tree Construction for 3D-Rendering Algorithms Like Ray Tracing, Sajid Hussain and Håkan .Grahn, G. Bebis et al. (Eds.): ISVC 2007, Part II, LNCS 4842, pp. 681–690, 2007,

[17] D. Eppstein, M. Goodrich, M. T. Sun; The Skip Quadtree: A Simple Dynamic Data Structure for Multidimensional Data,SoCG 2005,

[18] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Communications of the ACM, 51(1):117–122, 2008.

[19] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu. An optimal algorithmfor approximate nearest neighbor searching. Journal of the ACM, 45:891–923, 1998.

[20] L.Kaufman and P. J. Rousseeuw 1990.FindingGroups in Data: An Introduction to Cluster Analysis,John Wiley & Sons, 1990

[21] M. Ester, H.P Kriegal, J. Sander , and X. Xu 1996. ADenity-Based Algorithm for Discovering Clusters inLarge Spatial Databases with Noise. KDD 96 –Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining. 1996

[22] HrishavBakulBarua, Dhiraj Kumar Das &SauravjyotiSarmah“A Density Based Clustering Technique For Large Spatial Data Using Polygon Approach”, IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 3, Issue 6,PP 01-09, 2012.

[23] ChaudhariChaitali G. “Optimizing Clustering Technique basedon Partitioning DBSCAN and Ant Clustering Algorithm”, International

Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-2, pp. 212 – 215, December 2012

[24] Glory H. Shah, C. K. Bhensdadia, Amit P. Ganatra “An Empirical Evaluation of Density-Based Clustering Techniques”,International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, pp. 216 – 223, March 2012.

[25] A.K. Jain and R. C. Dubes, Algorithms for Clustering Data.Englewood Cliffs, NJ: Prentice-Hall, 1988.

[26] M. Ester, H. P. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", In: Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD“96), Portland, AAAI Press pp. 291-316, 1996.

[27] J. Sander, M. Ester, H. P. Kriegeland X. Xu, "Density- Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications", Int. J. of Data Mining and Knowledge Discovery, Kluwer AcademicPublishers vol. 2, pp. 169-194, 1998.

[28] R. Xu, and C. D. Wunsch, C. D., ”Survey of Clustering Algorithm”, IEEE Trans. of Neural Networks, Vol. 16, No. 3, pp. 645-678, 2005

[29] Q. B. Liu, S. Deng, C. H. Lu, B. Wang, Y.F. Zhou, “Relative Density based k-Nearest Neighbors ClusteringAlgorithm”, Proc. of 2nd Int. Conf. on Machine Learning and Cybernetics, Wan, 2-5 Nov. 2003, pp. 133-137.