Proceeding of IC-ITS 2015 e-ISBN:978-967-0850-07-8
International Conference on Information Technology & Society
8-9 June 2015, Kuala Lumpur, MALAYSIA

Design and Development of Enhanced Algorithms to Identify Crime Zones Using Data Mining Techniques

A. Malathi, A. Ganthi Mathi

PG and Research department of Computer Science, Government Arts College, Coimbatore, Tamil Nadu, India.

Kumaraguru College of Technology, Coimbatore – 641 018, Tamil Nadu, India.

malathi.arunachalam@yahoo.com

Abstract. Clustering are popular data mining techniques which are intended to help the user to discover and understand the grouping of the data in the set according to a certain similarity measure and predict future structure or group respectively. Two well-known clustering techniques, namely, K-means and DBScan (Density-Based Spatial Clustering Application) algorithm are considered. For each type of crime the results of the clustering process are used to identify crime zones. The various locations were grouped as high crime zone, medium crime zone and low crime zone. In this paper two algorithms K-Means and DBSCAN algorithms are enhanced to identify the crime zones.

Keywords: Clustering, K-Means, DBScan and Crime Zones.

INTRODUCTION

Clustering and classification are popular data mining techniques which are intended to help the user discover and understand the structure or grouping of the data in the set according to a certain similarity measure (Jain *et al.*, 1999) and predict future structure or group respectively. The popularity is behind the fact that they are the first data mining techniques to examine the structure and patterns in data (Hand *et al*., 2001).

In this main focus is the identification of crime zones. For this purpose, two well-known clustering techniques, namely, K-means (Kaufman and Rousseeuw, 1990) and DBScan (Density-Based Spatial Clustering Application) algorithm (Ester *et al.*, 1996) are considered. The results of the clustering process are used to identify trend city-wise for each type of crime.

REVIEW OF LITERATURE

Clustering is an important area of application for a variety of fields including data mining. Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). The clustering problem has been addressed in many contexts and by researchers in many disciplines. This reflects its broad appeal and usefulness as one of the steps in exploratory data analysis.

There exist a large number of clustering algorithms in the literature. The choice of clustering algorithm depends both on the type of data available and on the particular purpose and application. The clustering algorithms are categorized as i (Jain and Dubes, 1998).

IDENTIFICATION OF CRIME ZONES

DBSCAN (Density Based Spatial Clustering of Applications with Noise) is one of the most common clustering algorithms proposed by Ester et al. (1996). It is a density-based clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding nodes.

Data Set

The dataset was created by varying only the data size and missing percentage parameters. The datasize was varied with values ranging from 1000 to 5000 in steps of 1000 and the missing percentage was varied with values ranging from 10 to 40% in steps of 10. From the results, the winner for each data mining method is selected and is used in the proposed crime analysis tool.

Enhanced Dbscan

Every data mining task has the problem of parameters. Every parameter influences the algorithm in specific ways. For DBSCAN, as mentioned in the previous section, the parameters epsilon and MinPnts are needed to be provided by the user. The present project work uses k-distance graph to estimate these parameters. The procedure used to determine the parametes Eps and MinPts is explained in this section.

Let ‘d’ be the distance of a point ‘p’ to its k^{th} nearest neighbor, then the d-neighborhood of ‘p’ contains exactly k+1 points for almost all points ‘p’. The d-neighborhood of ‘p’ contains more than k+1 points only if several points have exactly the same distance ‘d’ from ‘p’ which is quite unlikely. Furthermore, changing ‘k’ for a point in a cluster does not result in large changes of ‘d’. This only happens if the k^{th} nearest neighbors of p for k = 1, 2, 3, …, are located approximately on a straight line which in general is not true for a point in a cluster.

1. Starting with the root node, the algorithm moves down the tree recursively, in the same way that it would if the search point were being inserted (i.e. it goes left or right depending on whether the point is less than or greater than the current node in the split dimension).
2. Once the algorithm reaches a leaf node, it saves that node point as the "current best"
3. The algorithm unwinds the recursion of the tree, performing the following steps at each node:
    1. If the current node is closer than the current best, then it becomes the current best.
    2. The algorithm checks whether there could be any points on the other side of the splitting plane that are closer to the search point than the current best. In concept, this is done by intersecting the splitting hyperplane with a hypersphere around the search point that has a radius equal to the current nearest distance. Since the hyperplanes are all axis-aligned this is implemented as a simple comparison to see whether the difference between the splitting coordinate of the search point and current node is less than the distance (overall coordinates) from the search point to the current best.
        1. If the hypersphere crosses the plane, there could be nearer points on the other side of the plane, so the algorithm must move down the other branch of the tree from the current node looking for closer points, following the same recursive process as the entire search.
        2. If the hypersphere doesn't intersect the splitting plane, then the algorithm continues walking up the tree, and the entire branch on the other side of that node is eliminated.
4. When the algorithm finishes this process for the root node, then the search is complete.

FIGURE 1: Search Algorithm Using Kd-Tree

The time requirement of DBScan algorithm is O(n log n) where n is the size of the dataset and because of this it is not a suitable one to work with large datasets. This when combined with k-distance graph to automatically select MinPts and Eps values, increases to $O(n^2 \log n)$. The present project work uses a KD Tree (space partitioning tree) to reduce the time complexity to O(log n) time. While using KD-Tree finding k nearest neighbours for each n data point the complexity is O(kn log n). The k value is very negligible and therefore does not make much different and hence the time complexity becomes O(n log n).

EXPERIMENTAL RESULTS

Performance evaluation of the enhanced algorithm the synthetic crime dataset was performed and the results are reported in this section. The performance metrics used to analyze the clustering algorithms are Silhouette measure and execution time. The performance was analyzed while varying the dataset size, number of clusters and keeping level of randomness and percentage of missingness to zero. This means that the created dataset will have no outliers and missing values.

Impact of Data Size

The influence of data size on silhouette measure and execution time is presented in Figures 2 and 3.

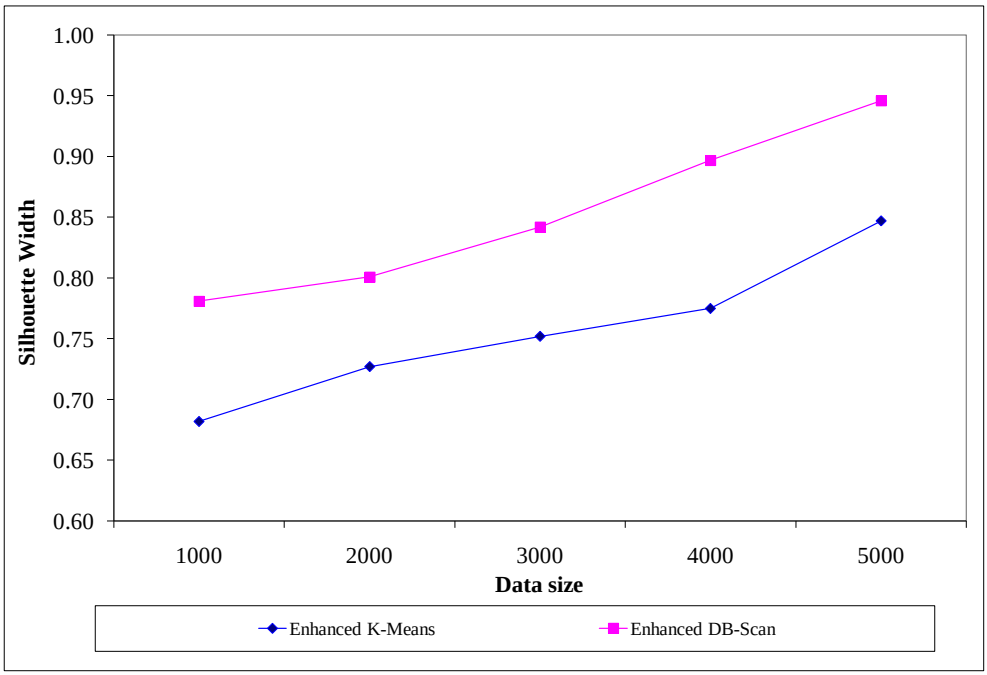

FIGURE 2. Effect of Data Size on Silhouette Width

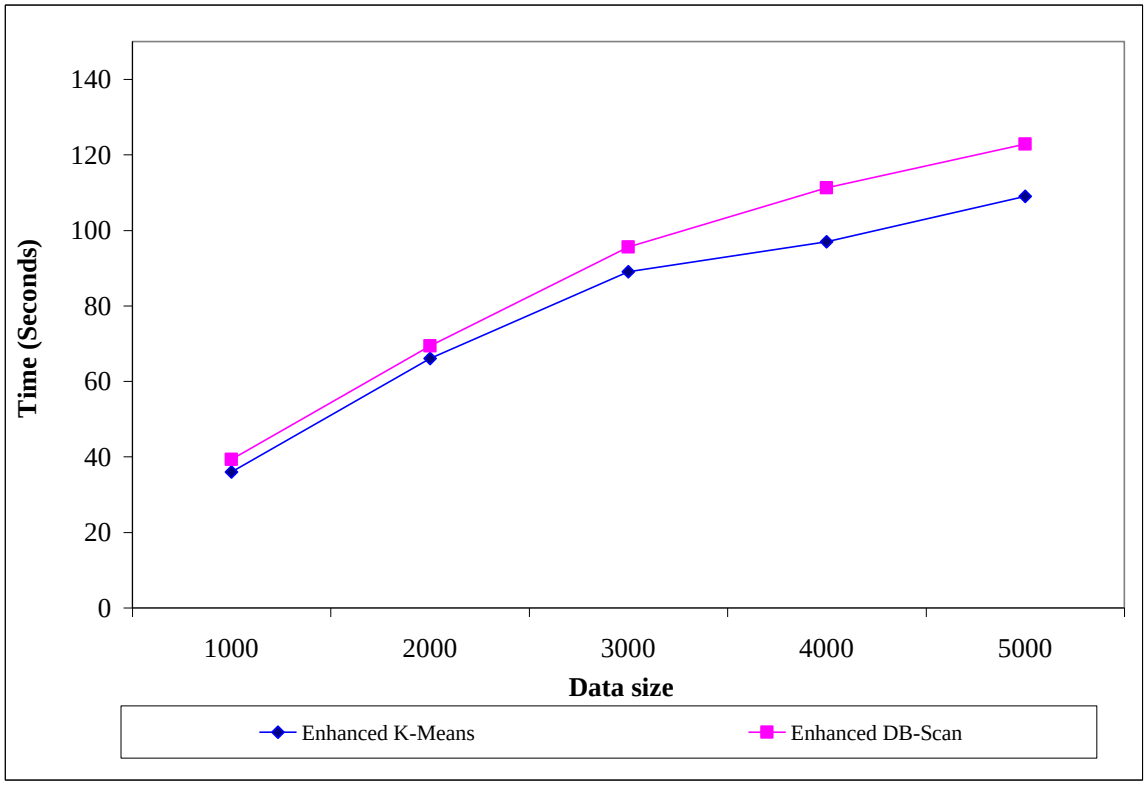

FIGURE 3. Effect of Data Size on Speed of Clustering

From the resutls, it could be seen that the enhanced DBSCAN algorithm scales well with both types of datasets as is evident from the high Silhoutte measure achieved. It can also be seen that the Silhoutte width increases with the dataset size, indicating that the clustering quality increases with large-sized datasets. The clustering quality while using with small sized dataset is also good as evident from the high value (0.6-0.7) obtained by datasets with size 1000 for enhanced K Means and enhanced DBSCAN. A similar trend was observed by entropy performance metric also, where again the Enhanced DBSCAN algorithm showed significant improvement.

While considering execution speed, the performance of Enhanced K Means algorithm outperformed Enhanced DBSCAN algorithm. This change in trend is more pronounced with large sized datasets (that is size > 3000). The efficiency gain obtained by Enhanced K Means algorithm over Enhanced DBSCAN algorithm in term of speed is 8.40%, 4.90%, 6.90%, 12.77% and 11.24% respectively for the five different sized datasets.

Thus, while considering dataset size, improved clustering accuracy is provided by Enhanced DBSCAN algorithm and time efficiency is gained by the Enhanced K Means algorithm.

Impact of Number of Clusters

The influence of number of clusters on silhouette measure, entropy measure and execution time is presented in Figures 4 and 5.

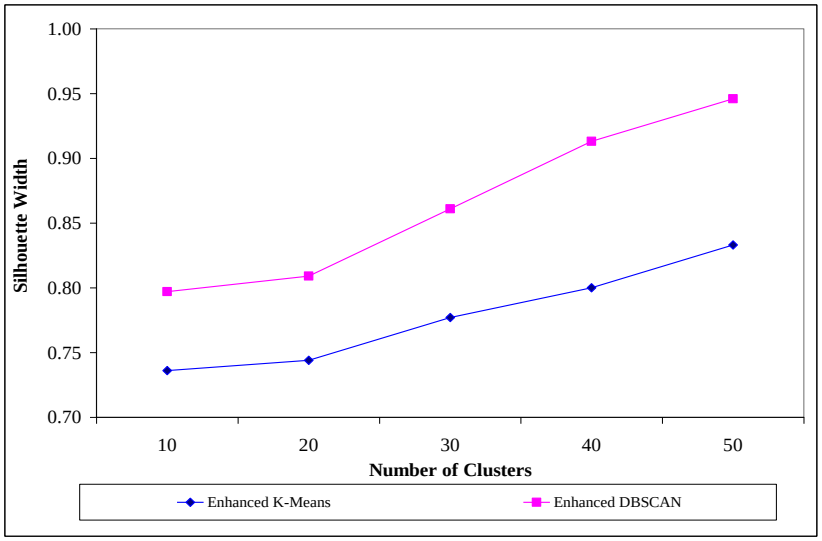

FIGURE 4. Effect of Number of Clusters on Silhouette Width

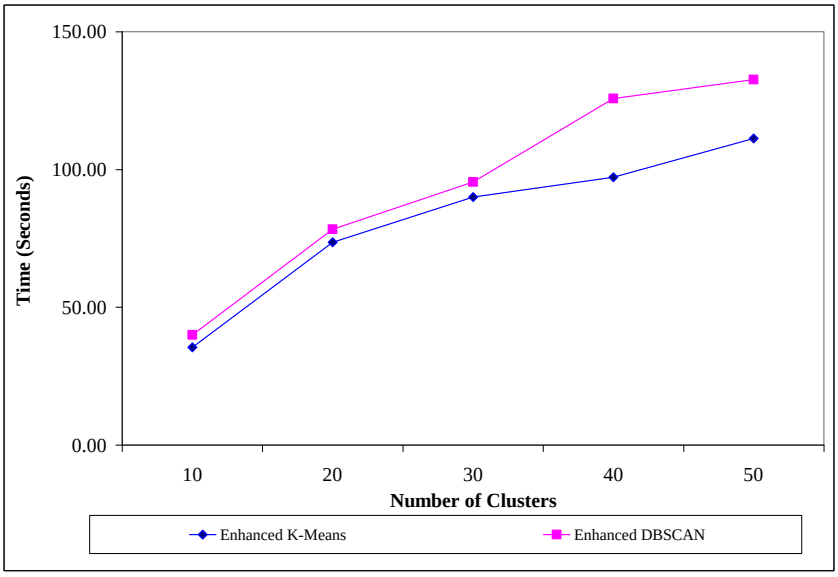

FIGURE 5. Effect of Number of Clusters on Execution Time

Testing the enhanced versions of K Means and DBSCAN with synthetic crime dataset on the impact of number of clusters on Silhoutte measure, the results showed that Enhanced DBSCAN algorithm produced near to unity values indicating that clustering quality has significantly improved. The accuracy was lower with small sized datasets. This shows that the results are consistent. Efficiency in entropy also portrayed similar trend.

From the results, it is evident that with both the algorithms, the time taken to perform clustering increased with the number of clusters. The best performance was achieved with small number of clusters (<20). This indicates that when the number of clusters increases the speed of the algorithm degrades. However, even with similar trend, the Enhanced K Means algorithm is faster than enhanced version of DBSCAN.

This section considered the use of K Means and DBSCAN algorithm to cluster crime data into groups. These groups can be used during crime analysis to detect crime

patterns. Both the algorithms were improved from their traditional manner of working. In both cases, an automated way to predict initial parameter values was used. Apart from this, automated method to select initial centroids was also used. With DBSCAN algorithm a KD-tree algorithm was combined to increase the search time efficiency.

Experiments conducted to analyze the efficiency of the algorithms revealed that Enhanced DBSCAN algorithm is efficient in term of clustering accuracy while enhanced K Means algorithm is efficient in terms of speed of clustering.

CONCLUSION

Encouraged by these results, the Enhanced DBSCAN algorithm was chosen for the clustering crime data in the proposed crime analysis tool. Another important operation during crime analysis is the prediction of future crime trends. The future crime rate prediction for various crimes was efficient in terms of future crime rate prediction. In the Crime data prediction framework the next step is Crime trends classification.

REFERENCES

1. Ester, M. Kriegel, H.-P. & Xu, X. (1996). Knowledge Discovery in Large Spatial Databases: Focusing Techniques for Efficient Class Identification, *Proc. 4th Int. Symp. on Large Spatial Databases, Portland, ME, Lecture Notes in Computer Science, Springer* , Vol. 951, Pp. 67-82.
2. Hand, D. Mannila, H. & Smyth, P. (2001). *Principles of data mining*, P. 594.
3. Jain, A.K. Mutry, M.N. & Flynn, P.J. (1999). *Data Clustering: A Review*. ACM Computing Surveys, Vol. 31, No. 3, Pp. 264-323.
4. Jain, A.K. & Dubes, R.C. (1988). *Algorithms for Clustering Data*. Prentice-Hall.
5. Kaufman, L. & Rousseeuw, P.J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, New York, NY.
6. Phung, D. Adams, B. Tran, K. Venkatesh, S. & Kumar, M. (2009). *High Accuracy Context Recovery using Clustering Mechanisms*, USA, Pp. 122-130.
7. Raiser, S. Lughofer, E. Eitzinger, C. & Smith, J.E. (2010). *Impact of object extraction methods on classification performance in surface inspection systems*. Special Issue Paper, Machine Vision and Applications, Springer-Verlag, PP. 627-641.