International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 7 (2015) pp. 16655-16664

http://www.ripublication.com

Improved Density-Based K-Clustering (E-DBK) Algorithm For Effectual Investigation Of Crime Data

V.Vinodhini[1] and M.Hemalatha[2]

Karpagam Academy of Higher Education, Eachanari Post, Coimbatore-641021.India.
Department of Computer Science; Karpagam University
1vvprof133@gmail.com 2 csresearchhema@gmail.com

ABSTRACT

Today, security is an aspect that is given top priority by all political and government worldwide and are aiming to reduce crime incidences. Crime plays a major role in all aspects. Crime controlling plays a foremost criterion in criminological aspect. Criminology is an area that focuses on the scientific study of crime and criminal deeds and law enforcement; it is a process that aims to identify crime characteristics. Crime analysis task explores and detects crimes and their relationships with criminals. The high volume of crime datasets and also the complexity of relationships between these kinds of data have made crime analysis an appropriate field for applying data mining techniques. Prophecy of crime characteristics is the first step for developing further analysis. The main study is to provide solutions that can augment the process of crime analysis for prophecy and for reducing crime incidences. The main aim of this paper has been achieved using an enhanced hybrid partitional and density based clustering algorithm. The proposed algorithm focuses on the identification of crime zone and the results can be used to identify trend of each type of crime city-wise. Moreover, the results can also be used to group various states as high crime zone, medium crime zone and low crime zone. From these harmonized groups, the efficiencies of police administration units, that is, states, in terms of crime rate, can also be measured. Experimental results prove that the proposed hybrid algorithm improves crime zone identification and grouping in a well-organized manner when compared with the conservative Density Based Spatial Clustering and KMeans algorithms.

1. INTRODUCTION

Today, security is considered to be one of the main tasks and the problem is

continuing to grow in strength and complexity. Security is an aspect that is given top priority by all political and government worldwide and are aiming to reduce crime incidence (David, 2006)**.** Reflecting to many serious situations like September 11, 2001 attack, Indian Parliament Attack, 2001, Taj Hotel Attack, 2006, Attack in School Campus(Kashmir) and amid growing concerns about burglary, arms trafficking, slay, the importance for crime analysis from previous history to identify crime trends is growing. The CIA, FBI, and other federal agencies are actively collecting domestic and foreign intelligence to prevent future attacks.

Crime is referred to as a wide-ranging concept that is defined in both legal and non-legal sense i.e., legal crime and non-legal crime. From the legal point of view crime is the flouting or breaching of the criminal law (penal code) that governs a particular geographical area (jurisdiction) aimed at protecting the lives, property and the rights of citizens of belonging to that jurisdiction. Crime is an offence against a person (for example slay, and sexual assault), or his/her property (for example, burglary and property damage) or the State regulation (for example traffic violations). In non-legal terms crime is a set of acts that violate socially accepted rules of human ethical or moral behavior; for example acting against a ritual in some society (Akpinar et al., 2005).

Crime analysis is a law enforcement function that involves methodical analysis for identifying and analyzing patterns, trends and disorder. Knowledge discovered or patterns obtained as a result of crime analysis help law enforcement agencies to deploy resources in a more effectual and competent manner. They can also be used to assist detectives and police in identifying and apprehending suspects. An ideal crime analysis tool should be able to identify crime patterns quickly and in an efficient manner for future crime pattern detection and action. However, existing systems faces the following two main issues that need to be addressed properly.

(i) Soaring and increasing large volume of data which puts the process of identifying and analyzing crime data into a difficult way.

(ii) Lack of structured data – The unpredictability in available data (both in format and attributes) makes the task of formal analysis a far more complicated.

Both these challenges aggravated this work to focus on providing solutions that can enhance the process of crime analysis for identifying and reducing crime rate. For this purpose, a crime data mining approach is adopted. Crime data mining can be performed using either descriptive of predictive tasks (Banerjee and Ghosh, 2002). Descriptive data mining tasks describe the general properties of the existing data. They find human-interpretable patterns that describe the data. Examples include association rule discovery, sequential pattern discovery, clustering, characterization, etc. Predictive data mining attempts to do predictions based on inference on available data. They use some variable to predict unknown or future values of other variables. Some predictive data mining techniques are classification, regression, outlier detection, change/evolution analysis, etc. This paper proposes the use of a clustering-based descriptive crime analysis method for identifying crime zones and crime trends.

The proposed clustering method combines two clustering algorithms, namely, KMeans and DBSCAN (Density Based Spatial Clustering of Applications with

Noise). Both the selection algorithms have been proved to be proficient in clustering community (Chakraborty *et al.*, 2011; Erman *et al.*, 2006). The KMeans method is an unsubstantiated, non-deterministic and iterative partition-based clustering method, because KMeans cannot work well in noisy environment, while the DBSCAN (Ester *et al*., 1996) method is a simple and effective density-based clustering algorithm which has been proved to work effectively in the field of clustering, i.e, even in noisy environment.

The KMeans algorithm is fast in producing clustering results, whose performance degrades in the presence of noise. On the other hand, DBSCAN algorithm is slow but can cluster efficiently even in presence of noise. Additionally, since DBSCAN works directly on the whole dataset, the amount of memory required is high. On the hope of combining the recompense of the two algorithms an efficient algorithm that is fast, efficient in the presence of noise, a method that integrates the advantages of KMeans and DBSCAN is designed and proposed in this paper.

The proposed hybrid algorithm uses a novel partition & merges procedure to reduce the memory requirement of DBSCAN and improves KMeans quality by removing noise found by DBSCAN algorithm. This algorithm is termed as E-DBK algorithm in this paper (Enhanced density based algorithm). The rest of the paper is organized as follows. Section 2 presents the steps involved in the design and development of the proposed E-DBK algorithm. Section 3 presents the experimental results obtained during performance evaluation, while Section 4 concludes the work with future research directions.

2. E-DBK ALGORITHM

The E-DBK algorithm consists of the following three steps to cluster the crime data for discovering crime trends.

(i) Use a preprocessing step that reduces the dimensionality of the dataset and partition into obtain k parts
(ii) Apply DBSCAN to each partition
(iii) Merge dense regions until the number of clusters is K

DBSCAN is a density-based clustering algorithm and can detect arbitrary shaped clusters. During clustering, the DBSCAN algorithm extracts dense regions by searching for core objects. This search is performed on the whole dataset within the specified Eps threshold. This step contributes to the heavy computations involved in DBSCAN. To reduce the heavy computations involved, a dataset is first partitioned into smaller sized sets and clustering is then performed on each of these partitions. Thus, the neighborhood search is performed only on the small sized partitions, which in turn reduces the number of scans and memory requirement of the algorithm.

In the next step, the KMeans algorithm is performed to obtain the initial K partitions, KP_a, where a ranges from 1... K. In the next step, the DBSCAN algorithm is applied on each KP_a with K, Eps and Minpts as parameters to produce clusters for each partition. This step improves the clustering performance by further reducing the number of scans. The result of applying DBSCAN algorithm to each partition outputs

a group of clusters for each partition. These small clusters can have similar dense regions which are merged in the final step.

The merge step uses a rule-based procedure to obtain the final clustering result. This step has to make sure that the resultant number of clusters is exactly K. The main goal of the merging process is to merge the nearest dense sub-clusters on each partition. For this purpose, a relative inter-connectivity measure is used. Relative interconnectivity was first introduced by Karypis *et al.* (1999) as a dynamic model for merging clusters by representing the objects in k nearest neighbor graph where an edge between one object and another exists if it is one of its k nearest neighbors.

The merge algorithm calculates two measures, namely, relative inter-connectivity and relative closeness, between two clusters, C_1 and C_2. Usage of these two measures makes sure that similar clusters are grouped effectively.

The relative inter-connectivity between a pair of clusters $C_1$and C_2 is defined as the absolute inter-connectivity between C_1 and C_2 normalized with respect to the internal inter-connectivity of the two clusters under consideration. The absolute inter-connectivity between the two clusters is defined as the sum of weight of the edges that connect vertices to C_1 to vertices in C_2.

This is essentially the edge-cut of the cluster containing both C_1 and C_2 such that the cluster is broken into C_1 and C_2. Let this measure be denoted as EC (C1, C2). The internal interconnectivity of a cluster C_i can now be captured easily by the size of its min-cut bisector, $EC(C_1)$, that is, the weighted sum of edges that partitions the graph into two roughly equal parts. Thus, the relative inter-connectivity between C_1 and C_2 is calculated as

$$RI(C_1,C_2) = \frac{|EC(C_1,C_2)|}{\frac{|EC(C_1)|+|EC(C_2)|}{2}} \quad (1)$$

where $|EC(C_1, C_2)|$ represents the absolute inter connectivity and is calculated using Equation (2) and EC, the internal inter connectivity is calculated using Equation (3).

$|EC(C_1, C_2)|$ = Sum of weight of edges that overlap over two clusters (2)

$EC(C_1)$ = Sum of weights of the edges in cluster $C_{1/}$ (3)

Equation (5.1) normalizes the absolute inter-connectivity with the average internal inter-connectivity of the two clusters. The relative closeness between two clusters C_1 and C_2 is defined as the absolute closeness between C_1 and C_2 normalized with respect to the internal closeness of the two clusters C_1 and C_2. The absolute closeness between a pair of clusters can be calculated using different methods that focus on the pair of points between all the representative points from C_1 and C_2 that are closest. A key drawback of these schemes is that by relying only on a single pair of points, they are less tolerant to outliers and noise. Thus, the study measures the closeness by computing the average similarity between the points in C_1 that are

connected to points in C_2. Since these connections can be obtained from k-nearest neighbor graph, their average strength provides a very good measure of the affinity between the data items along the interface layer of the two sub-clusters and at the same time it is forbearing to outliers and noise. Thus, the relative closeness (RC) is measured as

$$RC(C_{R1}, C_{R2}) = \frac{SEC(C_{R1}, C_{R2})}{\frac{|C_{R1}|}{|C_{R1}|+|C_{R2}|} SEC(C_{R1}) + \frac{|C_{Rj}|}{|C_{R1}|+|C_{R2}|} SEC(C_{R2})} \quad (4)$$

where SEC($\mathbf{C_{ri}}$, $\mathbf{C_{rj}}$),i.e., C_{R1},C_{R2} is the and relative closeness.The average weight of the edges are calculated which connect vertices in $\mathbf{C_{ri}}$ to $\mathbf{C_{rj}}$ and SEC($\mathbf{C_{rj}}$) represents the internal closeness.

Generally, two thresholds, T_{RI} and T_{RC} will be used during merging and two clusters will be merged if the following condition is satisfied.

$$RI(C_{ri}, C_{rj}) > T_{RI} \text{ and } RC(C_{ri}, C^{C_{rj}}) > T_{RC} \quad (5)$$

If more than one cluster satisfies the above condition, then C_{ri} is merged with a cluster that has the highest absolute inter-connectivity. After merging all the clusters, the whole procedure is repeated with the combined clusters. The merging process is repeated until none of the adjacent clusters satisfy the above two conditions or until K clusters have been formed. As the goal is to merge two clusters whose RC and RI both are high, instead of using two conditions, the present study proposes the use of a single function that combines the two measures and merge two clusters that maximum this function. The combined function is given in Equation (6).

$$RI(\mathbf{C_{ri}}, \mathbf{C_{rj}}) * RC(\mathbf{C_{ri}} * \mathbf{C_{rj}}) \quad (6)$$

Considering the whole cluster region at this stage will increase the overhead involved during the calculation of RI and RC. To reduce this, only the border objects are considered. As the DBSCAN algorithm already has information regarding this, no extra computations are required. RI and RC between two clusters based on border objects assume that there exists an edge between two border objects if their distance is less than Eps value. All dense regions which maximize Equation (5.5) are merged together. Finally, after merging, if the number of clusters is less than K, then calculate L as difference between K and m, where m is the number of merged clusters. If m > L, then repeat the whole process with new Eps, MinPts, SP, till K clusters are formed.

3. EXPERIMENTAL RESULTS

To evaluate the proposed algorithms and to ascertain their efficiency with respect to the existing methods the experiments were conducted using a real time crime datasets

called INSCR (Integrated Network for Societal Conflict Research) dataset. The dataset was downloaded from the INSCR website (http://www.systemicpeace.org/inscr/inscr.htm). The dataset has details regarding the crime in India and was posted and compiled by Marshall and Marshall (2008). The dataset contains records transcribed from the original materials published by the Government of India, Ministry of Home Affairs. The experiments were conducted in two stages. The first stage aimed to evaluate the performance gain obtained by the proposed hybrid model, while the second stage of experiments was used to identify the crime zones using the clustering algorithm.

a) Stage 1 : Performance Evolution
b) Stage 2 : Identifying Crime Patron using clustering algorithm.

3.1. Performance Evaluation of the Clustering Algorithm

The performance of the proposed hybrid clustering algorithm was evaluated using clustering accuracy and speed of clustering. Accuracy (Equation 7) discovers the one-to-one relationship between clusters and classes and uses this measurement to analyze the extent to which each cluster contains data points from the corresponding class.

$$Accuracy = \frac{\max\left(\sum_{C_{dk}, L_{dn}} T(C_{dk}, L_{dn})\right)}{m} x100 \quad (7)$$

where m is the number of data points, C_{dk} denotes the k^{th} cluster and L_{dm} is the n^{th} class. T(C_{dk}, L_{dm}) is the number of data points that belong to class n that are assigned to cluster k. Accuracy is computed as the maximum sum of T(C_{dk}, L_{dm}) for all pairs of clusters and classes and these pairs have no overlaps. Speed of clustering measured in seconds and is the time taken by the clustering algorithms to partition a given input dataset. Table 2 presents the accuracy and speed of the proposed E-DBK model and the conventional KMeans and DBSCAN algorithms.

Table 1 : Accuracy (%) and Speed (Seconds) of Clustering

Algorithm	Accuracy	Speed
KMeans	78.81	15.57
DBSCAN	82.26	18.66
E-DBK	89.22	13.05

The E-DBK algorithm showed an efficiency gain of 13.21% and 7.8% respectively over KMeans and DBSCAN algorithms. Thus, from the results, it is evident that the proposed model is efficient and has improved the clustering accuracy and speed of clustering.

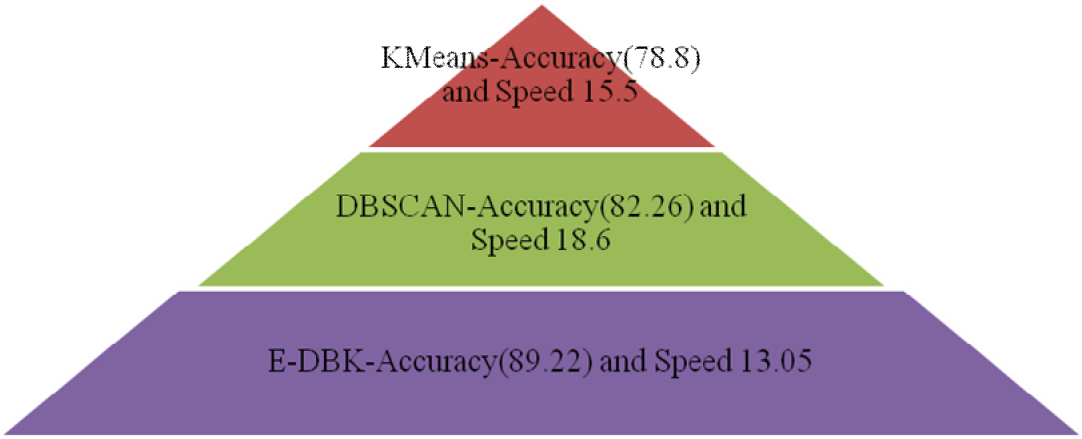

Fig 1: Accuracy (%) and Speed (Seconds) of Clustering

3.2. Identification of Crime Zones Using Clustering

Measuring efficiencies of state police forces has remained a constant area of governmental concern since these states are having diversities in area, population and crime density. So, the experiments were conducted to identify crime zones of states with similar crime density using the proposed clustering techniques while using the INSCR dataset.

Crime data contains the crime records of all 28 states and 7 union territories (UT) of India for year 2006 under 4 crime heads i.e. slaughter, Dacoits, Riots and (arson) inflammable. As a first step crime densities of the four crimes were calculated and the states were grouped into three crime zones such as High Crime Zone (H), Moderate Crime Zone (M) and Low Crime Zone (L) using E-DBK Clustering algorithm.

The result of clustering (crime zone group) along with the crime densities and crime zones is shown in Table 2. After analyzing the table, the following facts were identified.

- Elevated crime zone states-High - Andhra Pradesh, Bihar, Gujarat, Karnataka, Kerala, Madhya Pradesh, Maharashtra and Uttar Pradesh, Middling crime zone states-Medium - Rajasthan, Tamil Nadu and West Bengal
- Stumpy crime zone states -Low - Arunachal Pradesh, Assam, Chhattisgarh, Goa, Haryana, H P, J & K, Jharkhand, Manipur, Meghalaya, Mizoram, Nagaland, Orissa, Punjab, Sikkim, Tripura, Uttaranchal, A & N Islands, Chandigarh, D & N Haveli, Daman & Diu,Delhi, Lakshadweep and Pondicherry.

Thus the number of states identified as high crime zones is 8, medium crime zones are 3 and low crime zone states are 24.

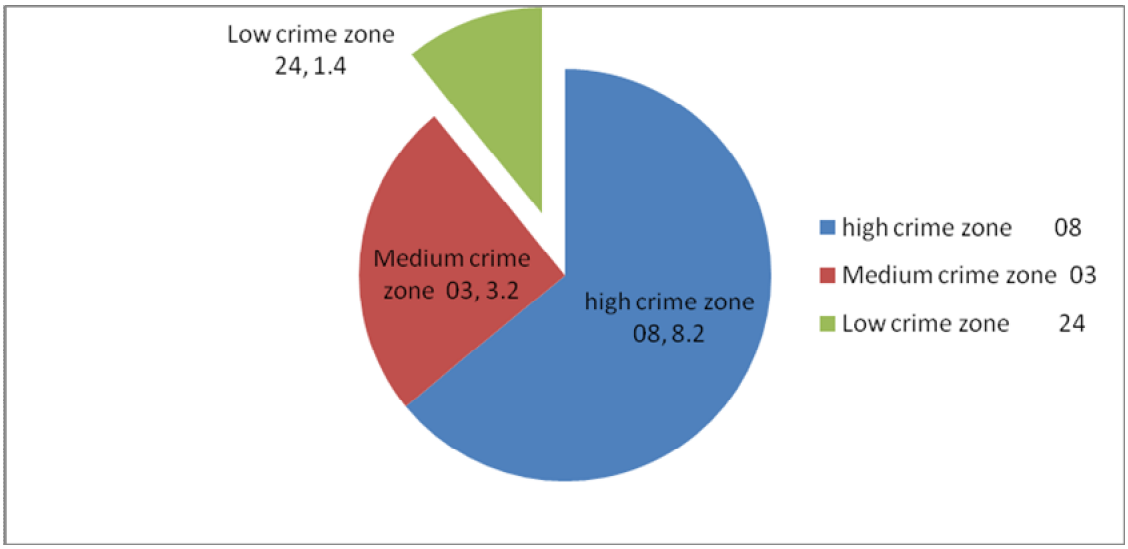

Fig 2: Crime Densities and Crime Zones

Table 2: Crime Densities and Crime Zones

State/UT	Murder	Dacoit	Riots	Arson	Crime Zone
Andhra Pradesh	2766	178	2916	1012	M
Arunachal Pradesh	60	28	6	20	L
Assam	1207	319	2684	488	L
Bihar	3249	5001	8259	2285	H
Chhattisgarh	1098	160	905	262	L
Goa	39	7	63	38	L
Gujarat	1165	290	1534	321	M
Haryana	873	104	1142	156	L
H P	111	7	566	115	L
J & K	487	10	1197	203	L
Jharkhand	1492	536	2650	178	L
Karnataka	5627	5782	6183	2768	H
Kerala	4393	3429	6365	3435	H
Madhya Pradesh	3309	3551	2308	1815	H
Maharashtra	2656	1663	7453	1188	H
Manipur	205	2	60	109	L
Meghalaya	157	57	7	28	L
Mizoram	25	7	0	25	L
Nagaland	123	16	7	16	L
Orissa	1159	239	1535	371	L
Punjab	817	35	3	68	L
Rajasthan	1209	37	1767	551	M
Sikkim	21	0	12	1	L
Tamil Nadu	1363	95	1838	460	L

Tripura	154	18	154	35	L
Uttar Pradesh	5480	218	3774	299	H
Uttaranchal	274	31	489	39	L
West Bengal	1425	177	2385	111	M
A & N Islands	4	0	10	9	L
Chandigarh	12	1	44	5	L
D & N Haveli	9	5	8	5	L
Daman & Diu	6	8	24	5	L
Delhi	476	14	87	33	L
Lakshadweep	0	0	12	4	L
Pondicherry	30	2	194	22	L

4. CONCLUSION

This study presents a hybrid clustering-based algorithm to analyze crime data for identifying crime zones and crime trends. The proposed algorithm combines KMeans and DBSCAN to enhance the clustering process, by combining these two algorithm an enhanced feature of novel based partition is introduced which works well both in speed and accuracy. A K Means algorithm is used in partition and comparing with the neighborhood cluster of the dataset. This step is performed to reduce the search space of the neighborhood clusters. The clustering algorithm instead of examining the whole dataset needs to search the objects only inside a single partition at a time, thus, reducing the number of search space computations. This step results in several small clusters, which are merged to obtain the optimal K clusters using two threshold measures namely, RC and RI. The two measures are combined to form a single function which has to be maximized to perform merging. Experimental results showed that the proposed algorithm is efficient in terms of clustering accuracy and speed of clustering and hence can be used professionally to identify crime zones in various cities of India. The results of clustering can be improved further through proper handling of missing values and outliers. Future research is premeditated in this direction.

REFERENCES

[1] David, G. (2006) Globalization and International Security: Have the Rules of the Game Changed?, Annual meeting of the International Studies Association, California, USA, http://www.allacademic.com/meta/ p98627_index.html.

[2] Akpinar, E. and Usul, N. (2005) Geographic Information Systems Technologies in Crime Analysis and Crime Mapping, Pp.1-12.

[3] Ester, M., Kriegel, H., Sander, J. and Xu, X. (1996) A density-based algorithm for discovering clusters in large spatial databases with noise, Evangelos Simoudis, Jiawei Han, Usama M. Fayyad, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press, Pp. 226–231.

[4] Karypis, G., Han, E.H. and Kumar V. (1999) Chameleon: hierarchical clustering using dynamic modeling, IEEE Comput, Vol. 32, Pp. 68–75.

[5] http://www.systemicpeace.org/inscr/inscr.htm, Last Access Date: November, 2013.

[6] Banerjee, A. and Ghosh, J. (2002) On scaling up balanced clustering algorithms,Proceedings of the 2nd SIAM ICDM, 333-349, Arlington, VA.

[7] Chakraborty, S., Nagwani, N.K. and Dey, L. (2011) Performance ComparisonofIncremental K-means and Incremental DBSCAN Algorithms, International Journal of Computer Applications, Vol. 27, No.11, Pp.14-18.

[8] Erman, J., Arlitt, M. and Mahanti, A. (2006) Traffic Classification Using Clustering Algorithms, Proceedings of the 2006 SIGCOMM workshop on Mining network data, ACM, New York, Pp. 281-286.

[9] Grubin, Don (1998) Sex offending against children: understanding the risk (PDF). London: Home Office. pp.v-vi and p.26-crime types and types of crime abuse.

[10] M Ester,HP Kriegel,J Sander-(1998) A density based algorithm for discovering clsters in large spatial database with noise.

[11] J Agrawal, S. Soni,S.Sharma (2014) Modification of density based spatial clustering algorithm for large database-ieeexplore.ieee.org.