IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

# A Comparative Analysis of Clustering Approach for Predicting Road Traffic Accident Dataset

**Srividhya[1], Mr. C. Palanichamy[2]**

Scholar, Department Computer Science Engineering, Chendhuran College of Eng and Tech, Pudukkottai[1]

Supervisor, Assistant Prof, Dept Computer Science Engineering, Chendhuran College of Eng and Tech, Pudukkottai[2]

**Abstract:** Road traffic accidents are the majority and severe issue, it results death and injuries of various levels. The traffic control system is one of the main areas, where critical data regarding the society is noted and kept as secured. Various issues of a traffic system like vehicle accidents, traffic volumes and deliberations are recorded at different levels. In connection to this, the accident severities are launched from road traffic accident database. Road traffic accident databases provide the origin for road traffic accident analysis. In this research work, road accident dataset is taken to consideration, the city having higher number of vehicles and traffic and the city having higher number of vehicles and traffic and the cost of these loss and accidents has a great impact on the socioeconomic growth of a society. Traditional machine learning algorithms are used for developing a decision support system to handle road traffic accident analysis. The algorithms such as k-means, Gaussian mixture model and Hierarchical clustering are implemented in MATLAB the result of these algorithms were compared. In this work, the algorithms were tested on a sample database of more than thousand five hundred items, each with 5 accident attributes. And the final result proves that the hierarchical algorithm was accurate and provides 99%.

**Keywords:** Data Mining, Gaussian mixture model, Hierarchical clustering, k-means, Road Accident data set.

## I. INTRODUCTION

Road traffic accidents are a social and public health challenge, as they almost always result in injuries and/or fatalities (Anderson 2009). The World Health Organization estimates over 1 million people are killed each year in road collisions. This is equal to 2.1% of the annual global mortality and an estimated social cost of $518 billion. To significantly reduce traffic fatalities and serious injuries on public roads, need to review the characteristics of traffic accidents and identify the hidden patterns behind the accidents‟ records, referring mainly to the actual knowledge contained in the collision data rather than the raw data records themselves. For example, road safety managers or residents may be interested in the accident patterns near their common unities and not `data records.

In this research work is to propose an efficient agglomerative hierarchical clustering algorithm method. It does not require feature selection and extraction. This method aims to reduce the attributes in traffic accident analysis. The computational time will vary depends on the attribute selection in hierarchical clustering here to use chameleon hierarchical clustering algorithm to measure the accuracy and response time. In this study is to address the efficiency issue of hierarchical clustering which is one of the main stream clustering methods as it is generally applicable to most types of data. In comparison with partition clustering algorithms such as K-means, hierarchical approaches have higher cost, with a complexity of O($N^2$ log N), but they do not require any predefined parameter hence are more suitable for handling real-world data where finding a suitable set of parameters can be tricky.

Hierarchical clustering can go both ways, aggregating from individual points to the most high-level cluster or dividing from a top cluster to atomic data objects. Our focus is the bottom-up approach which is known as the agglomerative approach, because computational cost can be reduced if the bottom-up process starts from somewhere in the middle of the hierarchy and the lower part of the hierarchy is built by a less expensive method such as partitional clustering. This idea would not work well on the top-down approach which is known as divisive hierarchical clustering because it is notorious for its high cost O($2^N$), and verifying middle level sub-clusters by individual data points would still be expensive.

It is possible to use a hierarchical approach to generate middle level sub-clusters then apply partitional algorithms on these sub clusters. However predefined parameters like K still need to be determined. Another possible way to improve efficiency in hierarchical clustering is to perform feature extraction or selection, which may reduce data dimensionality. However that process often requires domain knowledge of the data. It also makes the clustering outcomes dependent on the performance of the feature extraction or selection algorithms.

In this work is to present an efficient agglomerative hierarchical method which does not require feature extraction or selection. The main goals of this study are:

DOI10.17148/IJARCCE.2017.6683

1. Presenting a methodology of combining agglomerative hierarchical clustering and partitional clustering to reduce the overall computational cost. By this method the number of output clusters needs not to be determined beforehand.
2. Studying the behaviors of our methods with different distributions.
3. Evaluating the performance of our methods based on  the coefficients of correlation.

## II. LITERATURE REVIEW

Shafiq Alam et al presents in the agglomerative approach, the clustering process starts with every data element in an individual cluster, The individual cluster are, then merged on the basis of proximity until all the elements are in single cluster. [1] Proposes the work in an agglomerative manner starting from a relatively large number of particles and combining down, to only one final particles
Loris Bazzani et al defines the individual based category compact regions are classified as different entities including groups or persons exploiting a set of heuristics. In people that stand close for a while are joint into groups through a connection graph build by exploring heuristics on the moving regions. [2] Proposes; the hierarchical approach is suitable for decentralized data analysis and prediction.
Y. Chen et al [3] predict the distance between any two nodes Distance prediction  proceeds in the bottom fashion If two nodes belongs to the same cluster this implies their relatively close to each other So we predict the distance between then otherwise two nodes are belong to two different clusters. The hierarchical approach helps to improve the accuracy of the distance prediction.
Dominique Lord proposed the detailed driving data [acceleration, breaking and driver response...] and crash data that would better enable identification of cause and effect. Relationship with regard to crash probabilities are difficulty not available. As a result researches have framed their analytic approaches to study the factor that affect the number of crashes occurring in the specified time period.
Y. Chen et al specifies in the decentralized dataset each node owns co-ordinate points and local error all nodes adjust their network co-ordinates and local error via measuring their latencies to some other nodes in the system. [4] Proposes each attributes in the accident datasets measuring their latencies based on prototype and distances to other attributes in the dataset. Elth Ogston et al defines Centralized clustering is problematical if data is widely distributed dataset ae volatile (or) data items can’t be compactly represented. Decentralized on the other hand is a well- know problem .even in the centralized where each data item can compared to every data item ,perfect cluster can be hand to find. Decentralized creates the additional complication that even if the correct classification can be determined with the in complete inform available, the location of the item belonging to the class also need to be discovered [5].

## III.PROBLEM DESCRIPTION

Reducing the number of traffic accidents remains one of the greatest challenges facing many societies around the world. The cost of traffic accident on society and individuals is very high. Loss of life, disability and suffering are but a few Reducing the number of traffic accidents remains one of the greatest challenges facing many societies around the world. The cost of traffic accident on society and individuals is very high. Loss of life, disability and suffering are but a few  of the impacts of traffic accidents. On average a higher proportion of Indian drivers are involved in road accidents compared to their relative population among licensed drivers. A study of the reasons behind traffic accidents revealed four main factors: factors related to driving (the human factor); vehicle-related factors (physical environmental factors); mechanical factors, and socio-economic factors whether factor and Drinking driving. A traffic collision occurs when a road vehicle collides with another vehicle, pedestrian, animal, or geographical or architectural obstacle. It can result in injury, property damage, and death. Road accidents have been the major cause of injuries and fatalities in worldwide for the last few decades. It is estimated that the amount of data stored in the world’s database grows every twenty months at a rate of 100%. This fact shows that we are getting more and more exploded by data/ information and yet ravenous for knowledge. Data mining is a useful tool to address the need for sifting useful information such as hidden patterns from databases.

### A. Example Traffic accident data

Table: 1 Traffic accident data

| Accident | Gender | Age | Alcohol | Speed |
|---|---|---|---|---|
| 1 | M | Young | Yes | >100 |
| 2 | M | Young | No | 80-90 |
| 3 | M | Middle | No | 70-80 |
| 4 | F | Old | No | <60 |
| 5 | M | Young | Yes | 70-90 |

DOI10.17148/IJARCCE.2017.6683

## IV. METHODOLOGY

### A. Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc. The classic example of this is species taxonomy. Gene expression data might also exhibit this hierarchical quality (e.g. neurotransmitter gene families). Agglomerative hierarchical clustering starts with every single object (gene or sample) in a single cluster. Then, in each successive iteration, it agglomerates (merges) the closest pair of clusters by satisfying some similarity criteria, until all of the data is in one cluster.
The hierarchy within the final cluster has the following properties:

- Clusters generated in early stages are nested in those generated in later stages.
- Clusters with different sizes in the tree can be valuable for discovery.

A Matrix Tree Plot visually demonstrates the hierarchy within the final cluster, where each merger is represented by a binary tree.
Process

- Assign each object to a separate cluster.
- Evaluate all pair-wise distances between clusters (distance metrics are described in Distance Metrics).
- Construct a distance matrix using the distance values.
- Look for the pair of clusters with the shortest distance.
- Remove the pair from the matrix and merge them.
- Evaluate all distances from this new cluster to all other clusters, and update the matrix.
- Repeat until the distance matrix is reduced to a single element.

### B. K-means clustering

The K-means algorithm assigns each point to the cluster whose center (also called centroid) is nearest. The center is the average of all the points in the cluster — that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster.
The main advantages of this algorithm are its simplicity and speed which allows it to run on large datasets. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments. It minimizes intra-cluster variance, but does not ensure that the result has a global minimum of variance.

```
1.   Choose k number of clusters to be determined
2.   Choose k objects randomly as the initial cluster center
3.   Repeat
3.1. Assign each object to their closest cluster
3.2. Compute new clusters, i.e. Calculate mean points.
4. Until
4.1. No changes on cluster centers (i.e. Centroids do not change location any more) OR
4.2. No object changes its cluster (We may define stopping criteria as well)
```

The k-means algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intra cluster similarity is high but the inter cluster similarity is low. Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity

### C. Gaussian Mixture Model

An important step in the implementation of the above likelihood ratio detector is selection of the actual likelihood function, $p(X \mid \lambda)$. The choice of this function is largely dependent on the features being used as well as specifics of the application. For text-independent speaker recognition, where there is no prior knowledge of what the speaker will say, the most successful likelihood function has been Gaussian mixture models. In text-dependent applications, where there is strong prior knowledge of the spoken text, additional temporal knowledge can be incorporated by using hidden Markov models (HMMs) as the basis for the likelihood function. To date, however, use of more complicated likelihood functions, such as those based on HMMs, has shown no advantage over GMMs for text-independent speaker detection tasks as in the NIST SREs.

The GMM method is one way to improve the density of a given set of sample data modelled as a function of the probability density of a single-density estimation method with multiple Gaussian probability density function to model the distribution of the data. In general, to obtain the estimated parameters of each Gaussian blend component if given a sample data set of the log-likelihood of the data, the maximum is determined by the EM algorithm to estimate the optimal model. Principally, the GMM clustering method uses the following algorithm:

Input: Cluster number k, a database, stopping tolerance.
Output: A set of k-clusters with weight that maximize log-likelihood function.

1. Expectation step: For each database record x, compute the membership probability of x in each cluster h = 1,…, k.
2. Maximization step: Update mixture model parameter (probability weight).
3. Stopping criteria: If stopping criteria are satisfied stop, else set j = j +1 and go to (1).

In the analytical methods available to achieve probability distribution parameters, in all probability the value of the variable is given. The iterative EM algorithm uses a random variable and, eventually, is a general method to find the optimal parameters of the hidden distribution function from the given data, when the data are incomplete or has missing values

D. Euclidean Distance Measure:
In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula. By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space. The associated norm is called the Euclidean norm. Older literature refers to the metric as Pythagorean metric.
The Euclidean distance, data vector p and centroid q is computed as

$$d(p,q) = \sqrt{\sum_{k=1}^{n}(q_{ik} - p_{ik})^{2}}$$

## V. IMPLEMENTATION

A. Expected Outcomes
Use of a clustering methodology results are in the optimum number of membership functions.
It was found that, when the number of clusters was increased, the mean silhouette coefficient, which represents the overall quality of the clustering measurement, was decreased.
As explained above, by increasing the number of clusters, the R-value increased and the mean silhouette coefficient was decreased. Therefore, to satisfy two different evaluations for the cluster validity, 12 clusters were selected, which more than the minimum number of 10 clusters was obtained from subtractive clustering.

Twelve clusters were obtained from hierarchical clustering - as the optimum number of clusters, as at this value, the mean silhouette coefficient and R-value converged in the clustering algorithms. Clustering should be applied to the input and output of the training records, which comprised approximately 800 records of the overall used data. The optimum number of clusters and the number of rules should be equal; therefore, 12 rules were created. In addition, each input and output was characterized by 12 membership functions.
Our procedure was able to identify the best model based on precision (R) and response time (t). MLP model via exhaustive search took the greatest amount of time (2.635 seconds) with the best precision (R-value of 0.89).

B. Results
In AHC, when data set with N points is given to be clustered, N×N distance (similarity) matrix is produced. At the beginning, every point represents one cluster. Then algorithm finds most similar cluster pairs and combines them into a single cluster. After combining most similar cluster pair, algorithm finds the next most similar cluster pair and combines them. Combining clusters continue until desired number of cluster is reached.
This method divided into two divisions:

1. Accident predicted attributes
2. Accident unpredicted attributes

Table 1 and 2 shows clustering results of Agglomerative Hierarchical clustering algorithm.

True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) and Accuracy values have been calculated to evaluate efficiency of algorithm.

$$\text{Accuracy} = (TP+TN) / (TP+FP+TN+FN)$$

AHC algorithm has successfully distinguished anomaly attributes from normal attributes. Expected behavior from anomaly cluster is that number of members of anomaly cluster increases after accident, and anomaly cluster is supposed to contain only anomalies. Cluster 2 in AHC algorithm has shown anomaly behavior after accident.

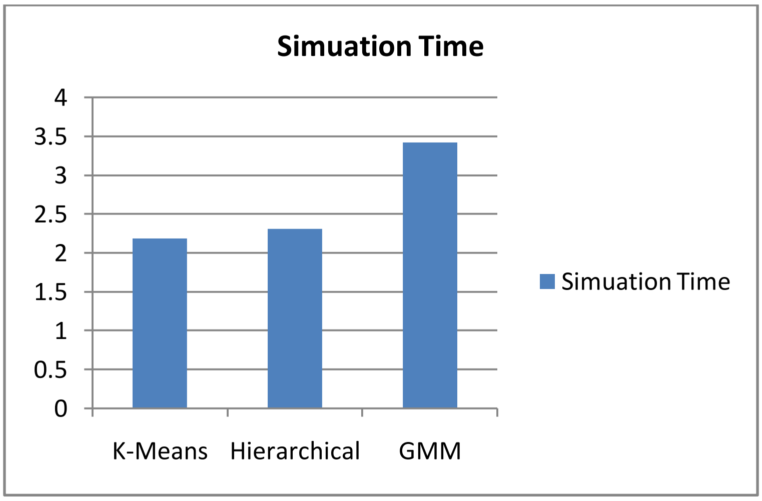
DOI10.17148/IJARCCE.2017.6683

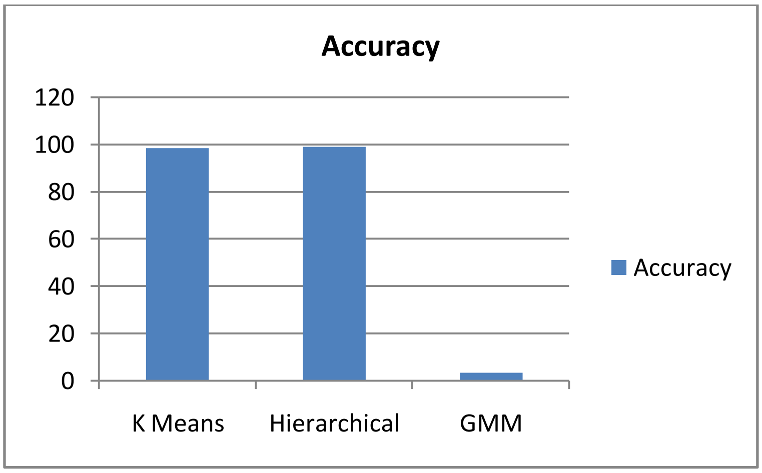
Simulation result 1

| Simulation Time(s) | TP(%) | FP(%) | TN(%) | FN(%) | Accuracy |
|---|---|---|---|---|---|
| 0.47 | 100 | 0 | 100 | 0 | 100 |
| 0.53 | 100 | 0 | 100 | 0 | 100 |
| 0.58 | 100 | 0 | 100 | 0 | 100 |
| 0.63 | 100 | 0 | 100 | 0 | 100 |
| 0.74 | 100 | 0 | 100 | 0 | 100 |

Statistical Rates for Clustering Algorithm
Simulation result 2

| Algorithm | Simulation Time | Accuracy |
|---|---|---|
| K-Means | 2.1882 | 98.7% |
| Hierarchical | 2.3142 | 99.02% |
| GMM | 3.4210 | 94.3% |





## VI. CONCLUSION

The experimental result in this work that indicate that decentralized data systems used to find clustering of large data sets in a reliable amount of time and produce good accuracy. The proposed work will categorized in two methods in traffic accident data that is necessary attribute for prediction and unnecessary attributes for prediction. Using chameleon hierarchical clustering algorithm based on the accuracy and response time to clustering the necessary attribute. The simulation results will produce 100%accuracy to find cluster of predicted attributes in the traffic accident dataset. The amount of response time will vary and comparing with other technique it will produce lesser time for

DOI10.17148/IJARCCE.2017.6683

clustering the large database data. In future needs to improve the accuracy and response time to using different neural network approach or using comparing technique to measure the maximum distance function and produce high accuracy with low response time.

## REFERENCES

[1] Particle Swarm Optimization Based Hierarchical Agglomerative Clustering byShafiq Alam, Gillian Dobbie, Patricia Riddle, M. Asif Naeem.
[2] Decentralized Particle Filter for Joint Individual-Group Tracking by Loris Bazzani, Marco Cristani ,Vittorio Murino.
[3] Pharos : A Decentralized and Hierarchical Network Coo ordinate System for Interne Distance Prediction by Yang Chen , Yongqiang Xiong , X iaohui Shi , Beixing Deng , XingLi.
[4] Pharos: accurate and decentralized network coordinate system by Y. Chen Y. Xiong X. Shi1 J. Zhu B. Deng1 X. Li.
[5] The Statistical Analysis of Crash-Frequency Data: A Review and Assessment of Methodological Alternatives by Dominique Lord.
[6] A Method for Decentralized Clustering in Large Multi - Agent Systems by Elth Ogston , Benno Overeinder, Maarten van Steen, and Frances Brazier.
[7] A Method for Decentralized Clustering in Large Multi-Agent Systems by Elth Ogston , Benno Overeinder, Maarten van Steen, and Frances Brazier.
[8] Traffic Accident Segmentation by Means of Latent Class Clustering by Benoît Depaire Geert Wets, Koen Vanhoof.
[9] A Generic Local Algorithm for Mining Data Streams in Large Distributed Systems by Ran Wolff, Kanishka Bhaduri,Member, IEEE, and Hillol Kargupta, Senior Member, IEEE.
[10] Hierarchically Distributed Peer-to-Peer Document Clustering and Cluster Summarization by Khaled M. Hammouda and Mohamed S. Kamel,Fellow.
[11].K.C. Gowda and G. Krishna, "Agglomerative Clustering Using the Concept of Mutual Nearest Neighborhood," Pattern Recognition, Feb. 1978, pp. 105-112.
[12] Ankerst M., Breunig M. M., Kriegel H.-P.,Sander J.: "OPTICS: Ordering Points To Identify the Clustering Structure", Proc. ACM SIGMOD, Philadelphia, PA, 1999, pp 49-60.
[13] Ester M., Kriegel H.-P., Sander J., Xu X.: "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proc. KDD'96, Portland, OR, 1996, pp. 226-231.
[14] M. Dutta, A. Kakoti Mahanta and A.K. Pujari, QROCK: A quick version of the ROCK algorithm or clustering of categorical data, Pattern Recognition Letters, 26(2005), 2364-2373.
[15] S. Guha, R. Rastogi and K. Shim, ROCK: A robust clusteringalgorithm for categorical attributes, Information Systems, 25 (2000), 345-36.
[16]. G. Karypis, E.H. Han and V. Kumar, CHAMELEON: Hierarchical clustering using dynamic modeling, IEEE Computer, 32 (1999), 68-75.
[17] Y. Song, S. Jin and J. Shen, A unique property of single-link distance and its application in data clustering, Data & Knowledge Engineering, 70 (2011), 984-1003.
[18] D. Krznaric and C. Levcopoulos, Optimal algorithms for complete linkage clustering in d dimensions, Theoretical Computer Science, 286 (2002), 139-149.
[19] P.A. Vijaya, M. Narasimha Murty and D.K. Subramanian, Leaders–Subleaders: An efficient hierarchical clustering algorithm for large data sets, Pattern Recognition Letters, 25 (2004), 505-513.
[20] V.S. Ananthanarayana, M. Narasimha Murty and D.K. Subramanian, Rapid and Brief Communication Efficient clustering of large data sets, Pattern Recognition, 34 (2001), 2561-2563.

DOI10.17148/IJARCCE.2017.6683