GE-International Journal of Engineering Research
Vol. 4, Issue 12, December 2016 Impact Factor- 5.613
ISSN(O): 2321-1717, ISSN(P): 2394-420X
© Associated Asia Research Foundation (AARF) Publication
Website: www.aarf.asia Email : editor@aarf.asia , editoraarf@gmail.com

OUTLIER DISCERNMENT USING K-MEANS

Madhav Bokare
Research Scholar
Priyadarshini Institute of Engineering & Technology,
Nagpur.

V. M Thakare
Professor
Sant Gadge Baba Amravati University,
Amravati.

ABSTRACT
In this paper we propose a clustering based method to capture outliers. We apply K-means clustering algorithm to divide the data set into clusters. We present a unified approach for simultaneously clustering and discovering outliers in data. Our approach is formalized as a generalization of the k-means problem. In this paper shows the outlier of sample data set by using K-means algorithm.

Keywords— *Apriori, Bary, CCT.*

I. INTRODUCTION

Outlier detection is a fundamental issue in data mining; specifically it has been used to detect and remove anomalous objects from data. Outliers, also called contaminant observations, are data points that deviate from other data points that they seem to be generated because of any of the faulty condition in the experimental setup. When observations that are taken in the experimental setup are subjected to analysis there is a possibility of the two conditions based on the outlier. Either (i) outliers negatively influence the results of analysis, or (ii) the search for outliers is the main task of data analysis. In data mining outlier detection is also regarded as the detection of anomaly. In many applications, a set of training values is required to define „normality". Security applications are examples, in which a typical behaviour by people or technical systems has to be detected. Outlier tests are used with the statistical model for generating the observations and presume some knowledge of the number of assumed outliers. Many of them can only cope with a single outlier. Outlier detection is used in various domains in data mining. This has resulted in a huge and highly diverse literature of outlier detection techniques. A lot of these techniques have been developed in order to solve problems based on some of the particular features, while others have been developed in a more generic fashion. It has been argued by many researchers whether clustering algorithms are an appropriate choice for outlier detection. For example, in (Zhang and Wang, 2006) [2], the authors reported that clustering algorithms should not be considered as outlier detection methods. This might be true for some of the clustering algorithms, such as the k-means clustering algorithm (Mac Queen, 1967) [3]. This is because the cluster means produced by the k-means algorithm is (Laan, 2003). But here we propose an algorithm that uses clustering efficiency of the k-means algorithm and uses a hybridized outlier detection technique of finding outlier through clustering.

In this paper we will propose a generalization of the k-means problem with the aim of simultaneously clustering data and discovering outliers. A naive approach to apply the k-means algorithm and list as outliers the top points that are the furthest away from their nearest cluster centres. However, there is a subtle point that needs to be noted: the k-means algorithm itself is extremely sensitive to outliers, and such outliers may have a disproportionate impact on the final cluster configuration. This can result in many false negatives: i.e., data points that should be declared outliers are masked by the clustering and also false positives: data points that are incorrectly labelled as outliers.

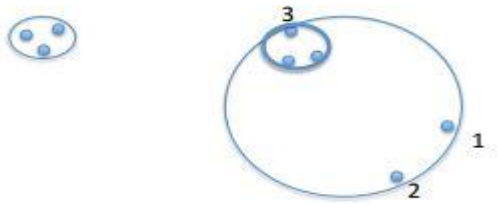

Figure 1: The k-means algorithm is extremely sensitive to outliers. By removing two points (1) and (2), we can obtain much tighter clusters (the

bold circle near 3). The objective of this paper is to obtain a tight clustering and report the outliers in an automat.

II. Related Work

K-means 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 apriori. The main idea is to define k centres, one for each cluster. These centres 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 centre. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycentre 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 centre. A loop has been generated. As a result of this loop we may notice that the k centres change their location step by step until no more changes are done or in other words centres do not move any more. Finally, this algorithm aims at minimizing an objective function know as squared error function given by: [4]

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} (|x_i - v_j|)^2$$

Where,
'||xi - vj||' is the Euclidean distance between xi and vj.
'ci' is the number of data points in ith cluster.
'c' is the number of cluster centres.

Algorithmic steps for k-means clustering:

Let X = {x1,x2,x3,……..,xn} be the set of data points and
V = {v1,v2,…….,vc} be the set of centres.
1) Randomly select 'c' cluster centres.
2) Calculate the distance between each data point and cluster centres.
3) Assign the data point to the cluster centre whose distance from the cluster centre is minimum of all the cluster centres.
4) Recalculate the new cluster centre using:

$$v_{i=}(1/c_i) \sum_{j=1}^{c_i} x_i$$

Where, 'ci' represents the number of data points in i^{th}cluster.

5) Recalculate the distance between each data point and new obtained cluster centres.
6) If no data point was reassigned then stop, otherwise repeat from step 3).

Benefits of K-means:

1) Fast, robust and easier to understand.
2) Relatively efficient: O (tknd), where n is # objects, k is # clusters, d is # dimension of each object, and t is # iterations. Normally, k, t, d << n.
3) Gives best result when data set are distinct or well separated from each other.

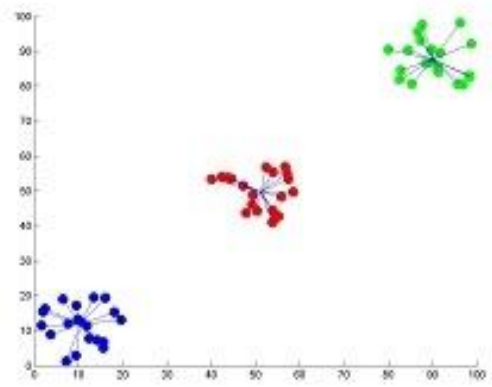

Figure 2 Fig 1: Showing the result of k-means for 'N' = 60 and 'c' = 3

III. Implementation and Result

At toll station the cameras are fixed for security purpose only which is costly. The database of vehicles which are passed through that toll it is in the printed format or documentary format there is no original record that how much vehicles which are under government duty through toll. So it is not difficult to make fake documentation of number of government vehicle passed. Another one there is no one for surveillance on the workers.

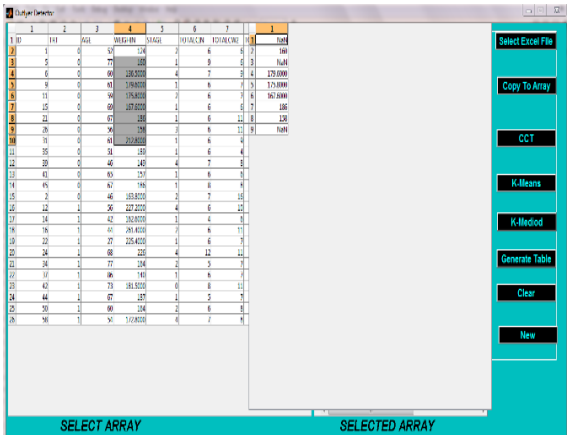

Figure 3 Sample data set

In the figure 3 shows the some options such as Select Excel File, Copy to Array, CCT, K-means. Here we are using sample data set. In left panel the selected data highlighted. Then click on Copy to Array option then CCT and then K-means. Finally output will display as graphical manner.

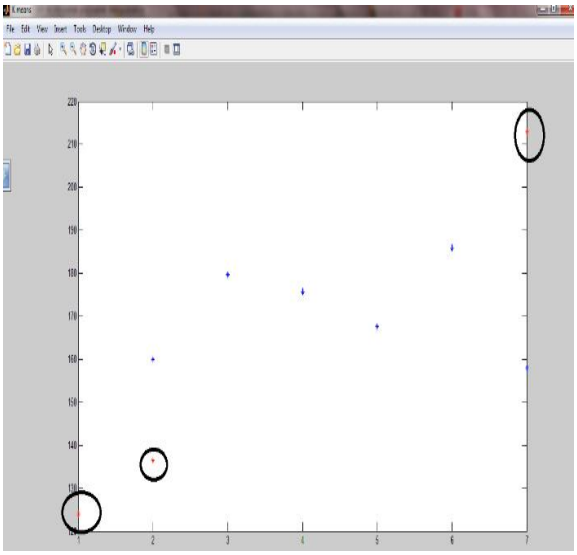

Figure 4Outlier using K-means

IV. CONCLUSIONS

In this paper we have shown the outlier of sampled data set by applying the k-means algorithm..

ACKNOWLEDGMENT

We wish to thank Dr. S. B. Thorat, Director,ITM,and Nanded.

REFERENCES

[1] "An Effective Clustering-Based Approach for Outlier Detection" by Moh"d Belal Al- Zoubi European Journal of Scientific Research Vol.28 No.2 (2009).

[2] "Detecting outlying subspaces for high-dimensional data: the new Task, Algorithms, and Performance, Knowledge and Information Systems", Zhang, J. and H. Wang, 2006.10 (3): 333-355.

[3] "Some methods for classification and analysis of multivariate observations" by MacQueen, J.,1967. Proc. 5th Berkeley Symp. Math. Stat. and Prob, pp. 281 -97.s.

[4] Web-link: https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm