

Hybrid Approach to Outlier Detection in Medical Dataset

Archana kadam¹ and Sagar G. Powar²

¹Assistant Professor, Department of Computer Engineering, PCCOE, Pune, India

²Senior Technical Leader, Xpanxion International Pvt. Ltd. Pune, India

E-Mail: kdm.archana@gmail.com

(Received 23 June 2017; Revised 16 July 2017; Accepted 29 July 2017; Available online 10 August 2017)

Abstract - Outlier detection has been a very important concept in the realm of data analysis and the complex relationships that appear with regard to patient symptoms, diagnoses and behavior are the most promising areas of outlier. The data typically consists of records which may have several different types of features such as patient age, blood group and weight. Recently, density-based outlier detection has emerged as a viable and scalable alternative to traditional statistical and geometric approaches. Density Based Outlier Detection Algorithm along with K-means partition technique is used for detecting outliers in Heart Disease dataset which is used to diagnose the abnormal data. This analysis can be used by doctor to predict heart disease of particular patient.

Keywords: K-means clustering technique, density-based algorithm, trajectory outlier detection (TRAOD), Heart disease dataset

I. INTRODUCTION

An outlier is a data object that is grossly different from or inconsistent with the remaining set of data [9]. It has been known that "one person's noise could be another person's signal." An outlier is an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism. The goal of outlier detection is to uncover the different mechanism. There are numerous different formulations of an outlier detection problem which have been explored in diverse disciplines such as statistics, machine learning, data mining and information theory.

Indeed, the outliers may be of particular interest, such as for the detection of credit card fraud and the monitoring of criminal activities in electronic commerce. There are many outlier detection algorithms reported in the literature. They can be classified into distribution-based [10], distance-based [7], density-based [5], [6] and deviation-based [4] algorithms. Most of them are designed to detect outliers from relational tuples (i.e., multi-dimensional point data).

In fact, the study with medical data by using the Data Mining techniques is virtually an unexplored frontier which needs extraordinary attention. The results of the present investigation suggest that a thorough understanding of the complex relationships that appear with regard to patient symptoms, diagnoses and behavior is the most promising area of outlier mining. Data mining approaches in medical domains is increasing rapidly due to the improvement effectiveness of these approaches to classification and prediction systems, especially in helping medical

practitioners in their decision making. In addition to its importance in finding ways to improve patient outcomes, reduce the cost of medicine, and help in enhancing clinical studies.

II. RELATED WORKS

Existing algorithms mainly focus on trajectory dataset. In 2008, Lee *et al.* proposed a two-stage partition-and-detect framework [2]. This framework is based on the line segment Hausdorff distance [8] originated from computer graphics and pattern recognition, and they proposed a TRAOD algorithm. In the first step, a trajectory is partitioned into set of line segments that is t-partitions and then applied Line Hausdorff Distance [8] which is classic method to calculate distance between two trajectories where the trajectory is the imaginary path of the data and co-ordinates of starting and ending points. And the second step TRAOD compares a t-partition with its neighbors to determine whether it belongs to outlying portions or not. The main advantage of this algorithm is the ability to detect outlying sub-trajectories from trajectory database. It has been used in many applications but there are some defects. In the detection phase, TRAOD applies some distance based method for finding the distance between two sub-trajectories and the compare this distance with global threshold value. It means that user has to select some threshold distance globally so the outlying t-partitions detected by this algorithm can be regarded as global outliers. The most tricky and sensitive parameter is D. The literature suggests that user should modify value of D repeatedly and check the repeatedly for searching outliers. A larger value of D generates smaller number of outliers and smaller value of D generates larger number of outliers.

In 2009, Liu *et al.* [3] proposed an algorithm based on R-tree. In this paper, they described how to use R-Tree to speed up outlier trajectories detection. Their method is used to judge the global match of two trajectories by calculating the local matching degree between segment pairs, which takes k consecutive points as the local comparing segment. Firstly, their algorithm needs five parameters to be manually assigned by users with experience, which is difficult to manipulate for majority people. For example, the number of detected sub-trajectories is proportional to k , which has to be assigned by users; the computational cost of their algorithm is largely depended on the neighborhood threshold value ω , and the suitable parameter is determined

by the experts in the special domain. Secondly, the R-tree has not been widely accepted as a spatial-temporal index structure. The R-tree is based on the rectangular region structure, but the ω -neighborhood region proposed in the literature is circular, this requires some extra computations. They did not compare the performance of their algorithm with that of TRAOD since the sub-trajectories which represent local characteristics in their algorithm are intertwined, which is the side effect of the R-tree data structure, so it is not possible to take advantages of the two-stage partition and- detect framework to optimize the performance of their algorithm.

In 2013, Liu [1], proposed an algorithm called density-based trajectory outlier detection (DBTOD). The concept of density takes account of the distribution of neighborhood objects, and it contains two components: the distance between the sub trajectories and the number of sub-trajectories within the given ranges. DBTOD employs this density-based concept to detect outliers. This paper performs and compares the DBTOD algorithm with the TRAOD algorithm. Based on the development of anomalous trajectory detection of moving objects, this paper introduces the classical trajectory outlier detection (TRAOD) algorithm, and then proposes a density-based trajectory outlier detection (DBTOD) algorithm, which compensates the disadvantages of the TRAOD algorithm that it is unable to detect anomalous defects when the trajectory is local and dense. The results of employing the proposed algorithm to Elk1993 and Deer1995 datasets are also presented, which show the effectiveness of the algorithm. In this paper, we describe a density-based detection approach. Based on this approach, we have developed a DBTOD algorithm. Our algorithm has two advantages. First, it is able to detect both anomalous sub-

trajectories and anomalous local trajectories. Second, it overcomes the sensitive parameter problem of TRAOD.

III. PROPOSED SYSTEM

In this paper we have proposed hybrid approach of density based outlier detection. In this system we are using two algorithms, K means clustering technique and density based algorithm. We first divide the dataset into a number of clusters so as to simplify the execution. The data chunks need to be made noise free. Thus the process of data-preprocessing is necessary. This helps in data redundancy, reducing the noise in the data and also normalizing it.

We first partition the dataset for that we are using K-means Clustering technique. 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 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 group age 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. In other words centroids do not move any more. Then we use detection stage to detect the outliers.

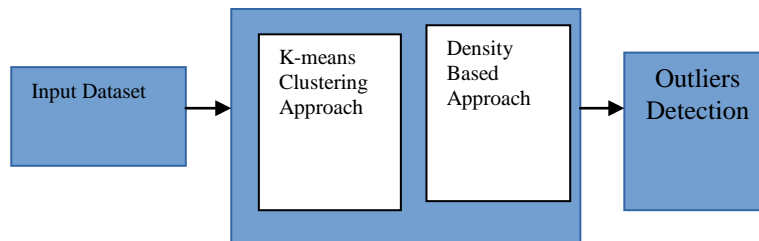


Fig.1 Hybrid Approach

A. System Architecture

An outlier is defined using the local outlier factor(LOF) of each object, which depends on the local density of its neighborhood. Here, the neighborhood is defined by the distance between the two points which are minimum. Data points with high LOF value are detected as outliers. The LOF does not suffer from the problem above. However, the computation of LOF values requires a large number of k-nearest neighbor queries and thus can be computationally expensive. Density Based Outlier Detection Algorithm

(DBOD) is a outlier detection algorithm based on partitioning and detection framework.

Figure 2 shows the functional block diagram of the proposed system model. It consists of two stages, partition and detection.

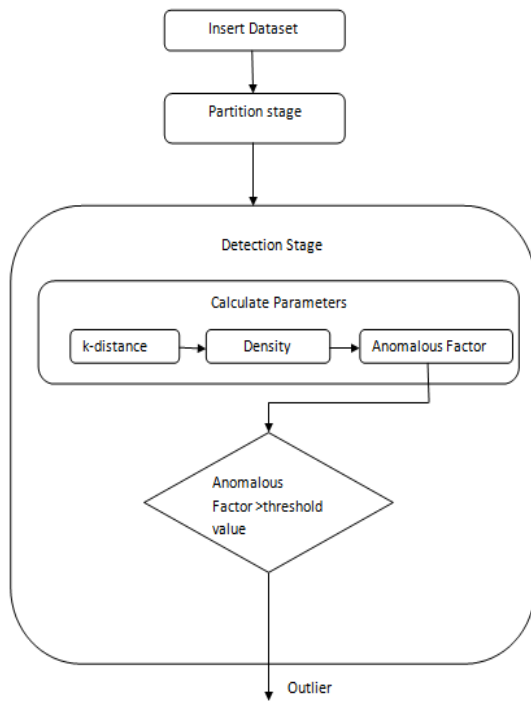


Fig. 2 Hybrid System Architecture

B.Partition Stage

K-means clustering technique

There are many algorithms for partitioning datasets. The *k*-means clustering is a popular method used to divide *n* patterns $\{x_1, \dots, x_n\}$ in *d* dimensional space into *k* clusters. The result is a set of *k* centers, each of which is located at the centroid of the partitioned dataset. This algorithm can be summarized in the following steps:

1. Choose the number of clusters *k* and input a dataset of *n* patterns $X = \{x_1, \dots, x_n\}$.

Randomly select the initial candidates for *k* cluster centers matrix $V(0)$ from the dataset.

2. Assign each pattern to the nearest cluster using a distance measure. For each pattern x_i , compute its membership $m(C_j | x_i)$ in each cluster C_j . The membership function $m(C_j | x_i)$ defines the proportion of pattern x_i that belongs to the *j*th cluster C_j . The *k*-means algorithm uses a hard membership function, that is the membership $m(C_j | x_i) \in \{0, 1\}$. If the pattern x_i is closest to cluster C_j (i.e., the distance between x_i and cluster center v_j is minimal), then $m(C_j | x_i) = 1$; otherwise $m(C_j | x_i) = 0$.

3. Recompute the centroids (centers) of these *k* clusters to find new cluster centers v_j , and compute the sum of square error *E*.

4. Repeat steps 2 and 3 until convergence. Typical convergence criteria are: no more reassignment of patterns

to new clusters, the change in error function *E* falls below a threshold, or a predetermined number of iterations have been reached.

C.Detection Stage

We use our density-based detection algorithm to compute the outliers.

Calculate *k*-distance of each point in partition *L*. The results are used to determine which objects consist of the neighborhood of *L*.

1. Calculate the distance of all present points' partition to *L*.
2. Choose *k* different minimum distances.
3. Choose the maximum value among the *k* distances, and choose it as the *k*-distance of *L*.

IV. RELEVANT MATHEMATICS

Let *D* represents the data points after the partition stage. *L*, *O*, *S* are arbitrary point in *D*.

1. Given an arbitrary positive integer *k*, *k* which is end user value. the *k* distance of *L*, denoted *k*-distance(*L*), is defined by $\text{dist}(L, O)$ which indicates the distance between *L* and *O*. This satisfies the following two conditions:

- (i) at least *k* points satisfying this condition $\text{dist}(L, O') \leq \text{dist}(L, O)$,
- (ii) at most *k* - 1 points satisfying this condition $\text{dist}(L, O') < \text{dist}(L, O)$.

2. Then calculate *k*-nearest neighbour set of *L* is defined by the set of *k* (*k* > 0) nearest points of *L*, denoted by $kNN(L)$. This satisfies the following two conditions:

- (i) $|kNN(L)| = k$ ($kNN(L)$) is the *k*-nearest neighbors,
- (ii) $L \notin kNN(L)$,
- (iii) if *S* and *S'* represent the *k*th and the (*k* + 1)th nearest neighbour points of *L* respectively, then $\text{dist}(L, S) \leq \text{dist}(L, S')$.

3. For each point *L* in given set, if there exist $S \in kNN(L)$ and $r = \text{dist}(L, S)$, such that given any $S' \in kNN(L)$ and $\text{dist}(L, S') \leq r$, the *k* nearest neighbor is given as $kNB(L) = \{S \in \text{given set} | \text{dist}(L, S) \leq r, L \neq S\}$.

4. Let *k* as the natural number, the reachability distance of point *L* corresponds to *O*, denoted $\text{reach-dist}(L, O)$, is defined by $\max\{k\text{-distance}(O), \text{dist}(L, O)\}$.

5. The local reachability density of point *L* is defined as

$$lrd_k = 1 / \left[\frac{\sum_{O \in kNB(L)} \text{reach} - \text{disk}_k(L, O)}{|kNB(L)|} \right]$$

6. The local anomalous factor of point *L* is defined as

$$llrd_k = \frac{\sum_{o \in kNB(L)} \frac{llrdk(O)}{llrdk(L)}}{|kNB(L)|}$$

7. The TR_i is an outlier if the following is true.

$$Ofrac(TR_i) = \frac{\sum_{Li \in OP(TR_i)} \frac{len(Li)}{len(Mi)}}{\sum_{Li \in P(TR_i)} \frac{len(Li)}{len(Mi)}} \geq F$$

where F is a parameter presented by an end user, $len(Li)$ is the length of a t-partition Li , $OP(TR_i)$ is the set of outlying t-partitions of TR_i , and $P(TR_i)$ is the set of all t-partitions of TR_i .

V. DBOD ALGORITHM

An outlier is defined using the local outlier factor (LOF) of each object, which depends on the local density of its neighborhood. Here, the neighborhood is defined by the distance between the two points which are minimum. Data points with a high LOF value are detected as outliers. The LOF does not suffer from the problem above. However, the computation of LOF values requires a large number of k-nearest neighbor queries, and thus, can be computationally expensive. Density Based Outlier Detection Algorithm (DBOD) is an outlier detection algorithm based on partitioning and detection framework.

Input: the dataset $I = \{TR_1, \dots, TR_{num}\}$, parameter k , anomalous factor threshold $threof$, anomalous proportion factor F

Output: The anomalous set $O = \{O_1, \dots, O_{numout}\}$ and the anomalous sub-points of each share O_i

```

/*Partition phase*/
1 For each  $TR_i \in I$  do
2 Partition  $TR_i$ ;
/*C represents the set of shares */
3 End For
4 For each  $Li \in C$ 
5 Caculate  $k$ -distance( $Li$ );
6 Caculate  $kNB(Li)$ ;
7 Caculate  $llrdk$  of  $Li$ ;
8 Caculate  $llofk$  of  $Li$ ;
9 End For
10 For each  $Li \in C$ 
12 if  $llofk > threof$  then
13 Mark  $Li$  as outlying points;
14 End if
15 End for
16 For each  $TR_i \in I$  do
17 Compute  $Ofrac(TR_i)$ ;
18 if  $Ofrac(TR_i) > F$ 
19 Mark  $TR_i$  as outlier;
20 Output  $TR_i$  with its share;
22 End if
23 End for
24 Return  $O = \{O_1, \dots, O_{numout}\}$ 

```

Fig.3 DBOD Algorithm

2. Find the k-distance neighborhood of the t-partition L. The k distance neighbor $kNB(L)$ includes each point which distances to L less and equal to k-distance(L). The purpose of calculating k-distance neighborhood is to find out the nearest neighbor of each object, the definition of the anomalous point is based on its neighborhood so the K-nearest neighbor is as follows:
 $kNB(L) = \{S \in D | \text{dist}(L, S) \leq k\text{-distance}(L)\}$.

3. Calculate the reachability distance of each point partition L. The reachability distance of L corresponds to O is $\text{dist}(L, O)$, which represents $\text{dist}(L, O) > k\text{-distance}(O)$. However, the reachability distance of point in partition L corresponding to O is the same as the k-distance of point partition O. The purpose of the reachability distance is making sure that all objects in neighborhood are very close to each other. The steps of calculating the reachability distance of point partition L are listed below.

- 3.1 Find out the k-distance neighborhood of point partition L, denoted $kNB(L)$.
- 3.2 Compute the reachability distance of point partition L corresponding to O, denoted $\text{reach-dist}(L, O)$, and it equals $\max\{k\text{-distance}(O), \text{dist}(L, O)\}$, where $O \in kNB(L)$.
4. Compute the local reachability density of point partition L.
5. Compute the local anomalous factor of point partition L.
6. Figure out the anomalous points.

VI. ANALYSIS OF DATASET

In this experiment the medical data related to Heart Disease is considered. This dataset was obtained from Cleveland database, which open source and publicly available. Cleveland dataset is about classification of person into normal, abnormal heart diseases.

VII. EXPERIMENTAL RESULT

In this section, we describe the experiments and the performance result of DBOD algorithm. This result give the comparison between class based partition and K-means partition which shows that density based outlier detection using k-means give better results.

A. Outlier Detection Using Class-Based Partition Technique

The data set is first partitioned into separate files according to their classes. If dataset contains four numbers of classes then the dataset is divided into files having class as name of file. If dataset have classes as $\{0,1,2,3\}$, the four number of files get created with name of file as $\{0.txt, 1.txt, 2.txt, 3.txt\}$. Each point is inserted into respective class file. After the partition only user defined input is read from each file for the outlier detection. Results are as shown in Table 1.

TABLE 1 (CLASS BASED)

Total Instances	250
Inliers	172
Outliers	78

More number of outliers are detected with this approach.

B.Outlier Detection Using K-Means Partition Techniques

The heart disease dataset is used in this system. The K-means clustering is applied on all attributes of dataset. The cluster size is decided by user. But if cluster size increases time required to execute is increased.

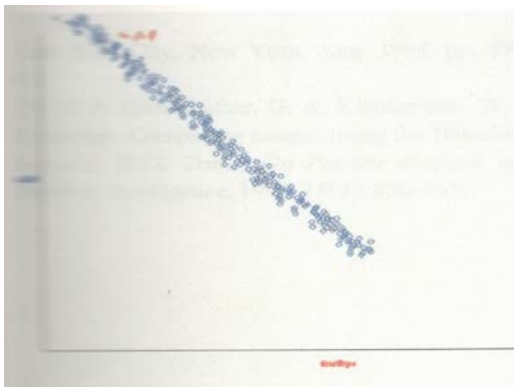




Fig.3 Outlier Detection Using K-Means Partition

Inliers 
 Outliers 

After clustering the detection is performed on the each cluster separately. The graph shows the inliers and outliers in dataset. The input dataset contains total 300 instances and three clusters are taken as input for clustering. Here the count of inliers and outliers are given for each cluster.

TABLE III COUNT OF INLIERS AND OUTLIERS

	Cluster 0	Cluster 1	Cluster 2
Inliers	79	33	158
Outliers	3	3	7

From above two tables table 1 and table 2 it is clear that density based outlier detection using K-means partition is more effective.

VIII. CONCLUSION

In this system, the Hybrid approach for outlier detection is introduced which consist of K-means clustering and density based outlier detection. We tested density based outliers detection algorithm using class based partition and density based outlier detection algorithm using k-means partition for the heart disease dataset. We observed that outliers detected by our hybrid approach are less than a density

based outlier’s detection algorithm using class based partition.

This paper concludes that density based outliers detection algorithm using K-means gives better performance, so it can be used effectively for detecting outlier in medical, criminal, terrorism etc. datasets in future. Results may vary depending on different datasets.

REFERENCES

- [1] Ming-Chuan Hung, Jungpin Wu+, Jin-Hua Chang And Don-Lin Yang “An Efficient k-Means Clustering Algorithm Using Simple Partitioning” [2005].
- [2] Zhipeng Liu1,2,*, Dechang Pi1, and Jinfeng Jiang1 “Density-based trajectory outlier detection algorithm”, April 2013, pp.335–340
- [3] Lee, J., Han, J., Li, X.: Trajectory Outlier Detection: A Partition-and-Detect Framework. In: In Proc. 24th ICDE Int’l Conf., Cancún, México, pp. 140–149 (April 2008)
- [4] L. X. Liu, S. J. Qiao, B. Liu, et al. Efficient trajectory outlier detection algorithm based on R-tree. *Journal of Software*, 2009, Vol.20, No.9:pp. 2426–2435. (in Chinese)
- [5] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," in Proc. 2001 ACM SIGMOD Int’l Conf: on Management of Data, Santa Barbara, California, May 2001, pp. 37-46.
- [6] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in Proc. 2000 ACM SIGMOD Int’l Conf: on Management of Data, Dallas, Texas, May 2000, pp. 427-438.
- [7] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in Proc. 2000 ACM SIGMOD Int’l Conf: on Management of Data, Dallas, Texas, May 2000, pp. 93-104.
- [8] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large datasets," in Proc. 24th Int’l Conf: on Very Large Data Bases, New York City, New York, Aug. 1998, pp. 392-403.
- [9] D. P. Huttenlocher, G. A. Klanderman, W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 1993, Vol.15,No.9:pp. 850–863.
- [10] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed. Morgan Kaufmann, 2006.
- [11] V. Barnett and T. Lewis, *Outliers in Statistical Data*. John Wiley & Sons, 1994.