Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Algorithm for Outlier Detection Based on Utility and Clustering (ODUC)

Sakthi Nathiarasan A
M. E- Student
Department of Computer Science and Engineering
Adhiyamaan College of Engg , Hosur, India

Abstract— Outlier analysis is one of the applied data mining technique. Outliers are data objects which do not comply with the general behavior or model of data. Statistical approach, distance-based approach, deviation-based approach are some of the outlier detection methods. Clustering data mining technique groups similar data objects into clusters, which indirectly eliminates outliers as noise. Not all the outliers are important for business improvement, this in turn gives birth to utility mining, which considers interestingness of the user while applying any data mining technique. The proposed system is to find outliers based on utilities and k-means clustering. i.e., first pruning the data objects whose utility value is lesser than user's minimum threshold value and the second step is to employ repeated k-means clustering. The repeated k-means clustering at each iteration prune's the data objects which lies near the centroid of the cluster.

Keywords— Outliers, Clusters, Utility, K-means, ODUC

I. INTRODUCTION

Outliers are deviated data objects which shows different characteristics and does not fits to any category of data objects. outliers could be caused by measurement or execution error or by inherent data variability. Outlier mining has wide applications. It can be used in fraud detection for finding unusual usage of credit cards or telecommunication services, in drug discovery, or in medical analysis for finding unexpected response to certain treatment. Outlier detection techniques can be classified as mainly 3 types, which are Distance-based, Deviation-based and Statistical approach. The statistical approach assumes the model underlying distribution that generates data set and then identifies outliers using a test called discordancy test. Distance-based approach is the most widely used technique which generalizes other notions provided by numerous discordancy tests and replaces many of these tests with a single algorithm detecting only outliers of the proposed types. Deviation-based techniques uses linear algorithm for deviation detection using implicit redundancy of the data. Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group or cluster are more similar to each other than to those in other groups or clusters. K-means clustering is the most widely used clustering algorithm. The data objects which does not fits to any clusters are termed as outliers. Clustering eliminates outliers as noisy data's. Not all the outliers are important for business development ,and so we need to include interestingness of the user. For instance cyber fraud detection of detecting steal attacks is important than detecting virus attacks. and here we proposed the algorithm ODUC (Outlier detection based on Utility and Clustering) for outlier detection based on k-means clustering and consideration of utility of the user. The proposed algorithm consists of two phases. The first phase prunes the data objects whose utility value is lesser than user's minimum threshold value. And the second phase consists of iterative k-means clustering. We apply K-means clustering to divide the data objects into clusters. The points or data objects which are lying near to centroid of the cluster are not probable candidate for outlier because they are clusters with strong similarity property and we can prune out such points from each cluster. Next, we calculate a distance-based measure for all remaining points, which is then used as a parameter to identify a point to be an outlier or not. We assume that there are *n* outliers in data set, and top *n* points will be reported as outliers by our method.

II. EXISTING SYSTEM

A. Outlier Analysis

Outlier analysis can be defined as: Given a set of data objects or data points N and the number of outliers K, find top K outlier points which are considerably dissimilar from the remaining data. The outlier analysis involves defining what data can be considered as inconsistent in a given data set and then to mine the outliers so defined by using efficient techniques.

B. Distance-Based outlier Detection

Knorr and Ng [6] were the first to introduce distance-based outlier detection techniques. A distance-based outlier is defined as follows: An object O in a dataset T is a distance-based outlier with parameters P and D, i.e., DB(P,D), if at least a fraction P of the objects in T lie at a distance greater than D from O. Distance-based outlier detection generally

avoids a lot of computation associated with fitting the observed distribution into some standard distribution and choosing discordancy tests. Therefore , a distance-based outlier is also called a unified outlier.

Several efficient algorithms for mining distance-based outliers have been developed. They are outlined as follows

1) *Index-based algorithm:* The index-based algorithm uses multidimensional indexing structures such as B-trees, to search for neighbours of each data object O in a dataset within radius D around that data object. As soon as m+1 neighbours are found, where M is the maximum number of data objects within the d-neighbourhood of an outlier, it is clear that data object O is not an outlier. This algorithm has the worst case complexity of O(K * N^2). Where K is the dimensionality and N is the number of items in the dataset.
2) *Nested-loop algorithm*: The nested-loop algorithm has the same computational complexity as the index-based algorithm but it avoids index structure construction and tries to minimize the number of I/O's. It divides the memory buffer space into two halves and the dataset into several logical blocks. By carefully choosing the order of loading the blocks into different halves, I/O efficiency can be achieved.
3) *Cell-based algorithm:* To avoid N^2 computational complexity, a cell-based algorithm is developed for memory-resident datasets, with complexity O(C^k +N), where C is a constant which depends on number of cells, and k is the dimensionality. The algorithm counts outliers on a cell-by-cell rather than object-by-object basis. The algorithm counts number of items in the current cell itself, in the cell and the first layer together, and in the cell and both layers together. Some items in the current cell can be outliers only if the total number of items in the cell and the first layer is less than or equal to the maximum number M of outliers that can be in the d-neighbourhood of an outlier. If this condition does not hold, then all the items in the cell can be removed from further investigation as they cannot be outliers. If the total number of items in the cell and both layers is less than or equal to M , then all items in a cell are outliers, and if it is more than M , then only some of the items in the cell are outliers.

C. Local Distance-Based Outlier Factor:

Zhang et.al. [10] proposed a local distance-based outlier detection method to find outliers from the data set. The local distance-based outlier factor (LDOF) of an object determines the degree to which the object deviates from its neighbourhood. Calculating LDOF for all points in the data set, makes overall complexity O(N2), where N is the number of points in the data set. The high ldof value of a point indicates that the point is deviating more from its neighbours and probably it may be an outlier. The factor ldof is calculated as follows [10]:

ldof of p: The local distance-based outlier factor of p is defined as:

$$ldof(p) = dp/ DP$$

dp (KNN distance of p): Let Np be the set of k-nearest neighbours of object p (excluding p). The k-nearest neighbours distance of p equals the average distance from p to all objects in Np. More formally, let dist(p, q) ≥ 0 be a distance measure between objects p and q. The k-nearest neighbours distance of object p is defined as:

$$dp=(1/k) \sum dist(p,q)$$

DP (KNN inner distance of p): Given Np of object p, the k-nearest neighbours inner distance of p is defined as the average distance among objects in Np.

$$DP=(1/k^2-k) \sum dist(q,q')$$

Calculating ldof values for all points is computationally expensive, since the complexity of this algorithm is O(N2) [10]. In our method, we try to reduce the computation while detecting the outliers by pruning some points which are probably not the outliers.

D. Clustering

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group in many applications.

The data objects which do not fits to any cluster is termed as outliers. It is possible to find outliers by clustering. Clustering methods like CLARANS [7] and CURE [2] may detect outliers. However, since the main objective of a clustering method is to find clusters, they are developed to optimize clustering, and not to optimize outlier detection. The definition of outlier used are subjective to the clusters that are detected by these algorithms. While definitions of distance-based outliers are more objective and independent of how clusters in the input data are identified. While existing work on outliers focuses only on the identification aspect, the work in [5] also attempts to provide intentional knowledge, which is basically an explanation of why an identified outlier is exceptional. The k-means clustering algorithm is a centroid-based technique which takes input parameter K , and partitions a set of 'n' objects into 'K' clusters so that the resulting intra cluster similarity is high whereas the inter cluster similarity is low.

1) *K-means Clustering Algorithm:*

Input: The number of clusters k, and a database containing n objects.

Output: A set of K clusters which minimizes the squared-error criterion.

Method: The k-means algorithm is implemented as follows

1. arbitrarily choose k objects as the initial cluster centers;
2. Repeat
3. (re)assign each object to the cluster to which the object is most similar. Based on the mean value of the objects in the cluster;
4. Update the cluster means, i.e., calculate the mean value of the objects for each cluster;
5. Until no change;

E. Utility Mining:

utility mining, is a new area in data mining which considers external utility factors or interestingness which applying data mining. In utility mining the data objects with utility greater than user's specified threshold are extracted. The utility here refers to the enumerative form of user's preference. An item set x is said to be a high utility item set if and only if $u(x) \geq$ minUtil, where minUtil is a user defined minimum utility threshold. It is also called cost sensitive learning or Active learning.

III. PROPOSED SYSTEM

The proposed algorithm ODUC(Outlier Detection based on Utility and Clustering) consists of 2 phases

Phase 1: pruning all low utility data objects (i.e., data objects whose utility value is lesser than user's minimum threshold value).

Phase 2: performing repeated k-means clustering to find and prune the data objects which lies nearer to the centroid of the cluster during each iteration. In this section we describe our modified LDOF for phase 2, which is an improvement over LDOF(Local distance based outlier factor). The main shortcoming with the LDOF algorithm proposed in [10] is, it is computationally expensive. This is because for each point p in the data set DS, one has to compute ldof. Since we are interested in the only outliers which are very few in numbers, the ldof computations for all the points are of little use and can be altogether avoided. We use K-means algorithm to cluster the data set. Once clusters are formed, we calculate radius of each cluster. Prune the points whose distance from the centroid is less than the radius of the respective clusters. After that for each unpruned points in every cluster we calculate the ldof. We report the top-n points with high ldof value as outliers.

A. *ODUC Algorithm*

Input: Number of clusters K, database containing N data objects, utility values for individual N data objects, minimum utility threshold MUtil, Number of outliers O, Number of iterations IT

Output: O- outliers.

Method : The ODUC is implemented as follows:

1. load the N data objects with their corresponding utility values.
2. Perform phase 1:

```
i=0;j=0;
do {
    if(utility of dataobject[i] <MUtil)
        {
            Prune(dataobject[i]);
        }
    else
        {
            Highutilitydataobject[j]=dataobject[i];
            j++;
        }
    i++;
    }while(i<N)
```

3. Perform phase 2:

```
i: Set  Y ← Kmeans( K,IT,High utilitydataobject)
ii: for each cluster Cj ∈ Y do
iii:        Radiusj← radius(Cj)
iv: end for
v: if |Cj | > O then
vi: for each point pi ∈ Cj do
vii:        if distance(pi, sj) < Radiusj then
viii:           prune(pi)
ix:         else
x:              Add pi to U
```



```
xi:             end if
xii:        end for
xiii: else
xiv:        for each point pi ∈ Cj do
xv:                 Add pi to U
xvi:        end for
xvii: end if
xviii: for each point pi ∈ U do
xix:            calculate ldof(pi)
xx: end for
xxi: Sort the points according to their ldof(pi) values.
xxii: First n points with highest ldof(pi) values are the desired outliers.
```

.

IV. CONCLUSIONS

In this paper, we proposed an efficient outlier analysis algorithm based on utility and clustering. We first identify some data objects which are less useful(less interest) to the user. Then we examine the data objects which are not probable candidates for outliers by using the radius of each cluster and we remove those points from the high utility data objects. Due to the reduction in the size of the data set, the computation time reduced considerably. We used a local distance-based outlier factor to measure the degree to which an object deviates from its neighbourhood. The precision of detecting outliers of our method is at per or higher than the existing methods though we pruned out some of the points.

.

REFERENCES

1) A. Ghoting, S. Parthasarathy, and M. Otey. "Fast mining of distance-based outliers in high-dimensional datasets" *DataMining and Knowledge Discovery,* 16(3):349–364, June 2008.
2) S. Guha, R. Rastogi, and K. Shim. CURE: "An efficient clustering algorithm for large databases". *SIGMOD Rec.*, 27(2):73–84, 1998.
3) W. Jin, A. K. H. Tung, and J. Han. "Mining top-n local outliers in large databases". In KDD '01: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 293–298, 2001.
4) *M. Knorr and R. T. Ng. "Algorithms for mining distance based outliers in large datasets". In Proc. 24th Int. Conf. Very Large Data Bases, VLDB, pages 392–403, 1998.*
5) *M. Knorr and R. T. Ng. "Finding intensional knowledge of distance-based outliers". In VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases, pages211–222, 1999.*
6) *M. Knorr, R. T. Ng, and V. Tucakov.'Distance-based outliers: algorithms and applications". The VLDB Journal, 8(3-4):237–253, 2000.*
7) R. T. Ng and J. Han. "Efficient and effective clustering methods for spatial data mining". pages 144–155, 1994.
8) S. Ramaswamy, R. Rastogi, and K. Shim. "Efficient algorithms for mining outliers from large data sets". pages 427–438, 2000.
9) Y. Tao, X. Xiao, and S. Zhou. "Mining distance-based outliers from large databases in any metric space". *KDD 06*, 20, 2006.
10) K. Zhang, M. Hutter, and H. Jin. "A new local distance-based outlier detection approach for scattered real-world data". In PAKDD '09*: Proceedings of the 13th Pacific-Asia Conferenceon Advances in Knowledge Discovery and Data Mining,* pages 813–822, 2009.

