International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064

Outlier Recognition in Clustering

Balaram Krishna Chavali[1], Sudheer Kumar Kotha[2]

[1]M.Tech, Department of CSE, Centurion University of Technology and Management, Bhubaneswar, Odisha, India

[2]M.Tech, Project Engineer, C-DAC, Mumbai, Maharashtra, India

Abstract: ***Outlier detection is a fundamental issue in data mining, specifically it has been used to detect and remove anomalous objects from data. Outliers arise due to mechanical faults, changes in system behaviour, fraudulent behaviour, network intrusions or human errors. Firstly, this thesis presents a theoretical overview of outlier detection approaches. A novel outlier detection method is proposed and analyzed, it is called Clustering Outlier Removal (COR) algorithm. It provides efficient outlier detection and data clustering capabilities in the presence of outliers, and based on filtering of the data after clustering process. The algorithm of our outlier detection method is divided into two stages. The first stage provides k-means process. The main objective of the second stage is an iterative removal of objects, which are far away from their cluster centroids. The removal occurs according to a chosen threshold. Finally, we provide experimental results from the application of our algorithm on a KDD Cup1999 datasets to show its effectiveness and usefulness. The empirical results indicate that the proposed method was successful in detecting intrusions and promising in practice. We also compare COR algorithm with other available methods to show its important advantage against existing algorithms in outlier detection.***

Keywords: outlier detection, clustering, intrusions

1. Introduction

Data mining, in general, deals with the discovery of non-trivial, hidden and interesting knowledge from different types of data. With the development of information technologies, the number of databases, as well as their dimension and complexity, grow rapidly. It is necessary what we need automated analysis of great amount of information. The analysis results are then used for making a decision by a human or program. One of the basic problems of data mining is the outlier detection. An outlier is an observation of the data that deviates from other observations so much that it arouses suspicions that it was generated by a different mechanism from the most part of data [1]. Inliers', on the other hand, is defined as an observation that is explained by underlying probability density function. This function represents probability distribution of main part of data observations [2].

Many data-mining algorithms find outliers as a side-product of clustering algorithms. However these techniques define outliers as points, which do not lie in clusters. Thus, the techniques implicitly define outliers as the background noise in which the clusters are embedded. Another class of techniques defines outliers as points, which are neither a part of a cluster nor a part of the background noise; rather they are specifically points which behave very differently from the norm [3].

Typically, the problem of detecting outliers has been studied in the statistics community. The user has to model the data points using a statistical distribution, and points are determined to be outliers depending on how they appear in relation to the postulated model. The main problem with these approaches is that in a number of situations, the user might simply not have enough knowledge about the underlying data distribution [4].

Outliers can often be individuals or groups of clients exhibiting behaviour outside the range of what is considered normal. Outliers can be removed or considered separately in *regression modeling* to improve accuracy which can be considered as benefit of outliers. Identifying them prior to modelling and analysis is important [1]. The regression modelling consists in finding a dependence of one random variable or a group of variables on another Variable or a group of variables.

1.1. Practical Applications

The identification of an outlier is affected by various factors, many of which are of interest for practical applications. For example, fraud, or criminal deception, will always be a costly problem for many profit organizations. Data mining can minimize some of these losses by making use of the massive collections of customer data [5]. Using web log files becomes possible to recognize fraudulent behaviour, changes in behaviour of customers or faults in systems. Outliers arise by reasons of such incidents. Thus typical fault detection can discover exceptions in the amount of money spent, type of items purchased, time and location. Many fraud cases can happen, for example, if someone has your name, credit card number, expiration date and billing address. All this information is very easy to obtain even from your home mailbox or any on-line transaction that you had before [6]. So, automatic systems for preventing fraudulent use of credit cards detect unusual transactions and may block such transactions on earlier stages.

Another example is a computer security intrusion detection system, which finds outlier patterns as a possible intrusion attempts. Intrusion detection corresponds to a suite of techniques that are used to identify attacks against computers and network infrastructures. Anomaly detection is a key element of intrusion detection in which perturbations of normal behaviour suggest the presence of intentionally or unintentionally induced attacks, faults and defects [7].

The system applied to real network traffic data is illustrated in Figure 1. The basic steps consist of converting data, building detection model, analysis and summarizing of results. In handwritten word recognition some errors were caused by non-character images that were assigned high character confidence value [8].

Segmentation and dynamic programming (DP)-based approaches are used for outlier rejection in off-line handwritten word recognition method. The flow diagram is shown in Figure 2. Segmentation splits a word image into partial characters than use character classifier and DP to obtain the optimal segmentation and recognition result. The recognition process assigns a match score to each candidate string and the highest score determines the result.

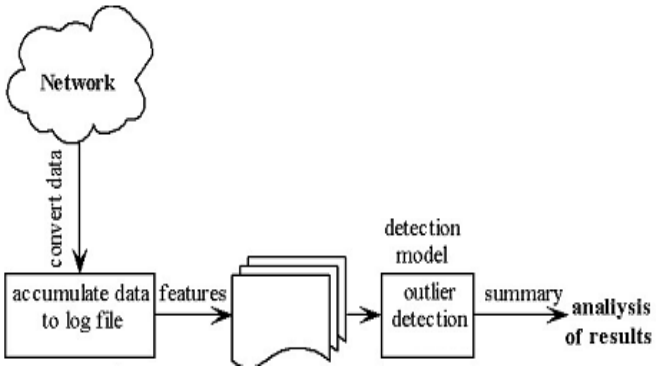


Figure 1: Outlier detection process in Data Mining

The focus of this approach is to assign low character confidence values to non-character images, which means to reject outlier. The neural networks were used to realize outlier rejection, where valid patterns only activate the output node corresponding to the class, which the pattern belongs to. Outliers do not activate any output node [32].

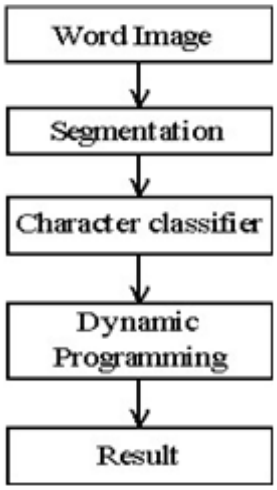


Figure 2: Diagram of handwritten word recognition system.

1.2. Outliers in Clustering

The outlier detection problem in some cases is similar to the classification problem. For example, the main concern of *clustering-based* outlier detection algorithms is to find *clusters* and outliers, which are often regarded as noise that should be removed in order to make more reliable clustering [2]. Some noisy points may be far away from the data points, whereas the others may be close.

1.3 Purpose of this Research

In this work, we consider outliers defined as points, which are far from the most of other data. The purpose of proposed approach is first to apply *k-means algorithm* and then find outliers from the resulting clusters. After that again apply k-means, and so on until the number of points will not be hanged in dataset. The principle of outliers' removal depends on the threshold and the distortion. Threshold is set by user and distortion defined as the ratio of distance for nearest point to the cluster *centroid* divided by distance of furthest point in the same *partition*. If the distortion is less than the threshold, this furthest point is considered to be outlier for this cluster. So, we propose a *clustering-based* technique to identify outliers and simultaneously produce data clustering. Our outlier detection process at the same time is effective for extracting clusters and very efficient in finding outliers.

2. Implementation & Algorithms

Notations of terms:

In this section we formally define the notations used in the reminder of thesis.

N Number of data objects.
M Number of clusters
K Number of attributes
X Set of N data objects $X = \{x1, x2, ..., xN\}$.
P Set of N cluster indices $P = \{p1, p2, ..., pN\}$.
C Set of M cluster representatives $C = \{c1, c2, ..., cM\}$.

2.1. Problem Definition

Clustering, or *unsupervised classification*, will be considered as a combination problem, where the aim is to partition a set of *data object* into a predefined number of clusters. Number of clusters might be found by means of the *cluster validity criterion* or defined by user.

2.2. Clustering Problems

The general clustering problem includes three sub problems: (i) selection of the evaluation function; (ii) decision of the number of groups in the clustering; and (iii) the choice of the clustering algorithm [10].

2.2.1. Evaluation of Clustering

An *objective function* is used for evaluation of clustering methods. The choice of the function depends upon the application, and there is no universal solution of which measure should be used. Commonly used a basic objective function is defined as (2.1):

$$f(P,C)=\sum_{i=1}^{N} d(x_i, c_{p_i})^2$$

where P is partition and C is the cluster representatives, d is a distance function. The Euclidean distance and Manhattan distance are well-known methods for distance measurement, which are used in clustering context. Euclidean distance is expressed as (2.2):

$$d(x_1, x_2)= \sqrt{\sum_{i=1}^{K} (x_1^i - x_2^i)^2}$$

and

$$d(x_1, x_2) = \sum_{i=1}^{K} \left| x_1^i - x_2^i \right|$$

Manhattan distance is calculated as (2.3)

2.3. Classification of Methods

Clustering algorithms can be classified according to the method adopted to define the individual clusters. The algorithms can be broadly classified into the following types: *partitional clustering*, *hierarchical clustering*, *density-based clustering* and *grid-based clustering* [15].

2.3.1 Partitional Clustering

Partition-based methods construct the clusters by creating various partitions of the dataset. So, partition gives for each data object the cluster index *pi*. The user provides the desired number of clusters *M*, and some criterion function is used in order to evaluate the proposed partition or the solution. This measure of quality could be the average distance between clusters; for instance, some well-known algorithms under this category are *k-means*, *PAM* and *CLARA* [13], [14].

One of the most popular and widely studied clustering methods for objects in Euclidean space is called *k-means clustering*. Given a set of *N* data objects *xi* and an integer *M* number of clusters. The problem is to determine *C*, which is a set of *M* cluster representatives *cj*, as to minimize the mean squared Euclidean distance from each data object to its nearest centroid.

Algorithm contains simple steps as follows. Firstly, initial solution is assigned to random to the *M* sets:

$$c_j \leftarrow x_i \,|\, j = random(1, M), i = random(1, N)$$

The number of iterations depends upon the dataset, and upon the quality of initial clustering data. The *k*-means algorithm is very simple and reasonably effective in most cases. Completely different final clusters can arise from differences in the initial randomly chosen cluster centres. In final clusters *k*-means do not represent global minimum and it gets as a result the first local minimum. Main advantage of the k-means method in follows: almost any solution not obtained by a *k*-means method can be improved.

2.3.2. Hierarchical Clustering

Hierarchical clustering methods build a cluster hierarchy, i.e. a tree of clusters also known as *dendogram*. A dendrogram is a *tree diagram* often used to represent the results of a cluster analysis. Hierarchical clustering methods are categorized into *agglomerative* (bottom-up) and *divisive* (top-down) as shown in Figure 4. An agglomerative clustering starts with one-point clusters and recursively merges two or more most appropriate clusters. In contrast, a divisive clustering starts with one cluster of all data points and recursively splits into non overlapping clusters.

3. Outlier Detection Methods

Most outlier detection techniques treat objects with *K* DWWULEXWHV DV SRLQWV LQ *K* space and these techniques can be divided into three main categories. The first approach is *distancebased* methods, which distinguish potential outliers from others based on the number of objects in the neighbourhood [11]. *Distribution-based* approach deals with *statistical methods* that are based on the probabilistic data model.

3.1. Distance-Based Approach

3.1.1. Distance-Based Definitions for Outliers

In *Distance-based* methods outlier is defined as an object that is at least *dmin* distance away from *k* percentage of objects in the dataset. The problem is then finding appropriate *dmin* and *k* such that outliers would be correctly detected with a small number of false detections. This process usually needs domain knowledge [12]. In the present section we define objects as points for simple interpretation and consider definitions as a special case of [18]. Firstly, consider the definition proposed by Knorr and Ng [4], which both a simple and intuitive: ***Definition:*** *A point x in a dataset is an outlier with respect to the parameters k and d, if no more than k points in the dataset are at a distance d or less from x.* To explain the definition by example we take parameter $k = 3$ and distance *d* as shown in Figure. Here are points *xi* and *xj* be defined as outliers, because of inside the circle for each point lie no more than 3 other points. And *x′* is an inlier, because it has exceeded number of points inside the circle for given parameters *k* and *d*.

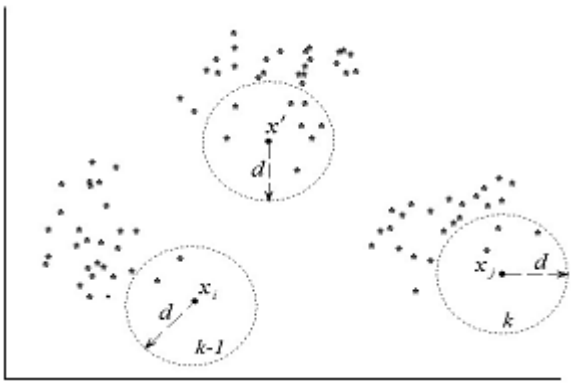


Figure 3: Illustration of outlier definition by Knorr and Ng

This approach does not require any a priori knowledge of data distributions as the statistics methods do. However, this distance-based approach has certain shortcomings:

1. It requires the user to specify a distance *d*, which could be difficult to determine apriori.
2. It does not provide a ranking for the outliers: for instance a point with a very few neighbouring points within a distance *d* can be regarded in some sense as being a stronger outlier than a point with more neighbours within distance *d*.
3. It becomes increasingly difficult to estimate parameter *d* with increasing dimensionality. Thus, if one picks radius *d* slightly small, then all points are outliers. If one picks *d* slightly large, then no point is an outlier. So, user needs to pick *d* to a very high degree of accuracy in order to find a modest number of points, which can be defined as outliers [3].

3.1.2. Hybrid-Random Algorithm

Hybrid-random algorithm was developed in [25]. It uses *Donoho-Stahel Estimator* (DSE) for distance-based operations in high-dimensional database. If two similar attributes are being compared, and these attributes are independent and have the same scale and variability, then all objects within distance *d* of a object *xi* lie within the circle of radius *d* centred at *xi*, as shown in Figure 8 on the left. In the presence of different scales, variability, and correlation, all objects within distance *d* of a object *xi* lie within an ellipse as in Figure 8 in the middle. If there is no correlation, then the major and minor axes of the ellipse lie on the standard coordinate axes but if there is correlation, then the ellipse is rotated through some angle θ, Figure on the right.

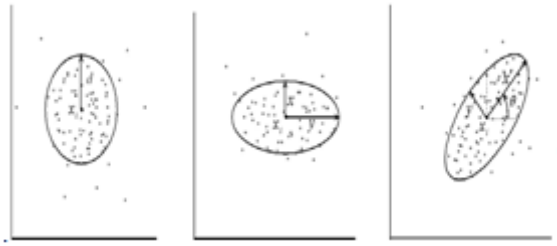


Figure 4: Data objects of the same scale and variability (left). Different scales, variability and no correlation (middle). Different scales, variability and correlation (right)

4. Experimental Results

In this section we present the results of an experimental study on synthetic unlabeled data and then on KDD Cup 1999 real-life labelled datasets [49] prepared for intrusion detection. The purpose of the experiment on real data was to detect intrusions.

4.1. Evaluation Results

We have applied the *Receiver Operating Characteristic* (ROC) analysis to evaluate the performance of the evolved method. In each of ROC plot, the x-axis is the *False Acceptance* (FA) rate, it indicates the percentage of normal connections classified as an intrusion. FA is calculated as a number of inliers detected as outliers divided by all detections. The y-axis is the *False Rejection* (FR) rate; it indicates the percentage of not detected outliers. FR is calculated as a number of not detected outliers divided by all outliers. A data object in the down left corner of the plot with FA and FR axes corresponds to optimal performance, i.e., low FA rate with low FR rate. *Half Total Error Rate* (HTER) is a combination of FR and FA values. We will calculate HTER values and show how they are change with varying threshold and number of clusters. HTER define as (FR+FA)/2. Similar evaluation methodology has been used in [12], [16].

4.2. Experiments with Synthetic Data

We ran our algorithm on the synthetic *DATA_A1* dataset; it contains 3000 objects grouped in 20 clusters. *DATA_A1* is illustrated in Figure 20 on left. At first, we should discuss some formal criteria. Given a dataset *TS* = *DATA_A1*, *C* is a codebook optimized for that dataset. *TS** is a dataset *TS* from which outliers have been removed, and *C** is a codebook optimized for *TS**. *C0* are original cluster centroids of the *TS* dataset, they were used for generating *TS*. *TS**, *C*, *C** we got after testing and used theirs to evaluate the efficiency of results. Thereto we calculated error for *TS* and *C* as *f(TS, C)* it means an average error from data to cluster centroids before any removing. Those errors have not much different from each other for parameter of various threshold values. Training error *f(TS*, C*)* means an average error from resultant data to their cluster centroids, its measure means the more outliers we remove, the less training error we get. Test error *f(TS, C*)* is an average error from original dataset to the resultant cluster centroids. It shows how much the distances are changed after removing. Also were measured the differences between the original clusters centroids and centroids of clustering process *f(C0, C)*, *f(C0,C*)* is the differences between the original clusters centroids and resultant centroids. And the differences between centroids of clustering process and resultant centroids are presented as *f(C, C*)*, it the bigger, the more outliers we remove.

5. Conclusions

This thesis proposes and analyzes a new outlier detection method called COR algorithm. It provides efficient outlier detection and data clustering capabilities in the presence of outliers. This approach is based on filtering of the data after clustering process. It makes those two problems solvable for less time, using the same process and functionality for both clustering and outlier identification. Moreover, we discussed the different categories in which outlier detection algorithms can be classified, i.e. density-based, distribution based and distance-based methods.

Furthermore, we applied algorithm to a real dataset KDD Cup 1999. Experimentally, COR is shown to perform very well on several real datasets. The results indicate that COR works effectively especially where the data has spherical shape. Its performance appears to degrade with datasets containing radial shape clusters and it is not recommended for this type of datasets. This study indicates that for datasets which have non spherical shape, we can improve the outlier detection results by setting the number of iterations. The experimental results demonstrate that the proposed method is significantly better than ODIN and MkNN in finding outliers.

With simple modifications, the method can be implemented for other distance metrics. An important direction for further study is how to apply the COR algorithm to the more general case, where the number of clusters and the threshold value must also be solved.

Also we can control the number of iterations. Moreover, a possible extension of this method would be to compare the performance of our method using different data clustering approaches. The main contribution of the present work is the design of an outlier detection process. Performed experiments demonstrate that COR algorithm was successful in detecting intrusions.

References

[1] G. Williams, R. Baxter, H. He, S. Hawkins and L. Gu, “A Comparative Study for RNN for Outlier Detection in Data Mining”. In Proceedings of the 2nd IEEE

International Conference on Data Mining, page 709, Maebashi City, Japan, December 2002.

[2] Z. He, X. Xu and S. Deng, "Discovering Cluster-based Local Outliers". Pattern Recognition Letters, Volume 24, Issue 9-10, pages 1641 – 1650, June 2003.

[3] C. Aggarwal and P. Yu, "Outlier Detection for High Dimensional Data". In Proceedings of the ACM SIGMOD International Conference on Management of Data, Volume 30, Issue 2, pages 37 – 46, May 2001.

[4] S. Ramaswamy, R. Rastogi and K. Shim, "Efficient Algorithms for Mining Outliers from Large Data Sets". In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Volume 29, Issue 2, pages 427 – 438, May 2000.

[5] C. Phua, D. Alahakoon and V. Lee, "Minority Report in Fraud Detection: Classification of Skewed Data". Special Issue on Learning from Imbalanced Datasets, Volume 6, Issue 1, pages 50 – 59, 2004.

[6] S. Alfuraih, N. Sui and D. McLeod, "Using Trusted Email to Prevent Credit Card Frauds in Multimedia Products". World Wide Web: Internet and Web Information Systems, Volume 5, Issue 3, pages 244 – 256, 2002.

[7] Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava, V. Kumar, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection". In Proceedings of the Third SIAM Conference on Data Mining, May 2003.

[8] J. Liu and P. Gader, "Neural Networks with Enhanced Outlier Rejection Ability for Off-line Handwritten Word Recognition". The journal of the Pattern Recognition society, Volume 35, Issue 10, pages 2061 – 2071, October 2002.

[9] M. Jaing, S. Tseng and C. Su, "Two-phase Clustering Process for Outlier Detection".Pattern Recognition Letters, Volume 22, Issue 6 – 7, pages 691 – 700, May 2001.

[10] P. Fränti and J. Kivijärvi, "Randomised Local Search Algorithm for the Clustering Problem". Pattern Analysis and Applications, Volume 3, Issue 4, pages 358 – 369, 2000.

[11] T. Hu and S. Y. Sung, "Detecting pattern-based outliers". Pattern Recognition Letters, Volume 24, Issue 16, pages 3059 – 3068, December 2003.

[12] V. Hautamäki, I. Kärkkäinen and P. Fränti, "Outlier Detection Using k-Nearest Neighbour Graph". In Proceedings of the International Conference on Pattern Recognition, Volume 3 pages 430 – 433, Cambridge, UK, August 2004.

[13] L. Kaufman and P. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis". John Wiley Sons, New York, USA, 1990.

[14] J. Han and M. Kamber, "Data Mining: Concepts and Techniques". The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, 550 pages, August 2000. (http://www.cs.sfu.ca/~han/DM_Book.html), visited 11.11.2004.

[15] E. Paquet, "Exploring anthropometric data through cluster analysis". Published in Digital Human Modeling for Design and Engineering (DHM), pages, Rochester, MI,June, 2004.

[16] W. Lee, S. Stolfo, K. Mok, "A Data Mining Framework for Building Intrusion Detection Models". In Proceedings of the 1999 IEEE Symposium on Security and Privacy, pages120 – 132, May 1999.