Clustering-Based Outlier Detection Method

Sheng-yi Jiang[1,2],Qing-bo An[1]

[1] *School of Informatics, GuangDong University of Foreign Studies, 510006,Guangzhou, China*
[2] *Guangdong Province Key Laboratory of Information Security, Sun Yat-sen University, Guangzhou, 510275, China*
Jiangshengyi@163.com

Abstract

Outlier detection is important in many fields. The concept about outlier factor of object is extended to the case of cluster. Based on outlier factor of cluster, a clustering-based outlier detection method, named CBOD, is presented. The method consists of two stages, the first stage cluster dataset by one-pass clustering algorithm and second stage determine outlier cluster by outlier factor. The time complexity of CBOD is nearly linear with the size of dataset and the number of attributes, which results in good scalability and adapts to large dataset. The theoretic analysis and the experimental results show that the detection method is effective and practicable.

1. Introduction

An outlier is an observation that deviates so much from other observations as to arouse suspicion generated by a different mechanism [1].

Traditional pattern recognition aims to find the general pattern for the majority of data and treats outliers as noise. This, however, could result in the loss of important hidden information since one person's noise could be another person's signal. In many fields, outliers are more important than the normal data, as they may demonstrate either deviant behavior or the beginning of a new pattern, and may cause damage to the user. Outlier detection targets the finding of the rare data whose behavior is very exception compared with other data, it becomes a growingly useful tool in some applications, such as credit card fraud detection, pharmaceutical research, loan approval, intrusion detection, marketing and customer segmentation, and so on.

Following of Hawkins' definition, researchers have proposed various outlier definitions and schemes for outlier detection, which describe outlier from different views. In this paper, we generalize the concept of outlier factor of object to the case of cluster and put forward a clustering-based outlier detecting method. We regard the cluster which comes by clustering process as a unit and identify it as "normal" or "outlier" (the id of a cluster is also the id of its objects). The method is made up of two stages: the first stage is grouping dataset with clustering algorithm; the second stage is to identify the gained clusters as "normal cluster" or "outlier cluster" according to their outlier factor.

The rest of this paper is organized as follows .In section 2, we discuss related work on outlier detection. In section 3, we give two definitions and generalize the concept of outlier factor of object to the case of cluster. In section 4, we discuss one-pass clustering algorithm and present a clustering-based outlier detection method. The experimental results are given in Section 5. Finally, section 6 provides the conclusions.

2. Related work

A few clustering algorithms such as *DBSCAN*, *BIRCH*, *ROCK*, *STING*, *WaveCluster* can also handle outliers, but their main concern is to find clusters and the outliers in the context of clustering are often regarded as noise. In general, outliers are typically just ignored or tolerated in the clustering process for these algorithms are optimized for producing meaningful clusters, which prevents giving good results on outlier detection.

Most methods on outlier mining in the early work are based on statistics. These methods can be mainly classified into two categories: *distribution-based* and *depth-based* ones. The *distribution-based* methods use standard distribution to fit the dataset. Outliers are defined on the basis of probability distribution. The main problem with *distribution-based* method is that it assumes that the underlying data distribution is known a prior. However, for many applications, the prior knowledge is not always obtainable, and the cost for fitting data with standard distribution is significantly considerable.

978-0-7695-3305-6/08 $25.00 © 2008 IEEE
DOI 10.1109/FSKD.2008.244

Distance-based scheme [2] declares a point as an outlier if its neighborhood contains less than *pct*% of a whole dataset. This notion generalizes many concepts from *distribution-based* method and enjoys better computational complexity. *Distance-based* scheme is further extended to improve the effectiveness of detection [3-4].

Deviation-based method [5] identifies outliers by inspecting the main characteristics of objects in a dataset and the objects that "deviate" from these features are considered as outliers.

Breuning[6] introduced the concept of "local outlier", and proposed a *density-based* formulation scheme. It uses the "local outlier factor" (*LOF*) to measure how strong an object can be an outlier. Since the *LOF* value of an object is obtained by comparing its density with those in its neighborhood, it has stronger modeling capability than *distance-based* scheme, which is based only on the density of the object itself. There are several enhancement schemes [7-8].

Cluster-based outlier detection techniques were recently developed. Jiang, M.F.[9] proposed an outlier-finding process, named *OFP*, which based on k-means algorithm and regarded small clusters as outliers. Yu, D. [10] introduced an outlier detection method, termed *FindOut*, based on WaveCluster. The main idea in *FindOut* is to remove the clusters from the original data and thus identify the outliers. He, Z. et al. [11] proposed the concept of cluster-based local outlier and outlier detection method *FindCBLOF*, which used "*cluster-based* local outlier factor" for identifying the outlier-ness of each object. The method *OFP* and *FindOut* can only process numerical-attribute data; on the contrary, *FindCBLOF* can only process categorical-attribute data. Jiang, S.[12] presented outlier detection *TOD*, which improves the efficiency of *FindCBLOF* method and can process mixed-attribute data.

3. Notations and Definitions

We suppose that dataset *D* is featured by *m* attributes (m_C categorical and m_N continuous) and D_i is the *i-th* attribute. For simplicity, we set categorical attributes before continuous attributes. We give two definitions and generalize the concept about outlier factor to the case of cluster. To reduce the infection of varied measurement units, it is necessary to standardize numerical attributes.

Definition 1: For a cluster *C*, the cluster summary information (*CSI*) for *C* is defined as: $CSI = \{kind, n, Summary\}$, where *kind* is the type of the cluster *C* with the value of 'normal' or 'outlier', n is the size of the cluster *C* ($n = |C|$), and *Summary* is given in terms of the frequency information for categorical attribute values and the centroid for numerical attributes:

$$Summary = \{< Stat_1, \cdots, Stat_{m_C}, c_{m_C+1}, c_{m_C+2}, \cdots, c_{m_C+m_N} >$$

$$\left| Stat_i = \{(a, Freq_{C|D_i}(a)) | a \in D_i\}, 1 \le i \le m_C, c_j \text{ is centroid}, m_C + 1 \le j \le m_C + m_N \right\}$$

Where, $Freq_{C|D_i}(a)$ is the frequency of value a in D_i.

Definition 2: [13] The distance between clusters C_1 and C_2, $d(C_1, C_2)$ is defined as $d(C_1, C_2) = \sqrt{\sum_{i=1}^{m} dif(C_i^{(1)}, C_i^{(2)})^2 / m}$, where $dif(C_i^{(1)}, C_i^{(2)})$ is the difference between C_1 and C_2 on attribute D_i. For categorical attributes, $dif(C_i^{(1)}, C_i^{(2)})$ is defined as

$$dif(C_i^{(1)}, C_i^{(2)}) = 1 - \frac{1}{|C_1| \cdot |C_2|} \sum_{p \in C_1} Freq_{C_1|D_i}(p_i) \cdot Freq_{C_2|D_i}(p_i),$$

$$= 1 - \frac{1}{|C_1| \cdot |C_2|} \sum_{q \in C_2} Freq_{C_1|D_i}(q_i) \cdot Freq_{C_2|D_i}(q_i)$$

While for the numerical attribute, $dif(C_i^{(1)}, C_i^{(2)})$ is defined as $dif(C_i^{(1)}, C_i^{(2)}) = \left| c_i^{(1)} - c_i^{(2)} \right|$ .where $c_i^{(1)}, c_i^{(2)}$ is respectively corresponding to centroid of C_1 and C_2 on D_i.

Specially, we regard an object as a cluster including only one object and can compute distance between two objects or distance between an object and a cluster.

Definition 3: Let $C = \{C_1, C_2, \cdots, C_k\}$ be the results of clustering on dataset *D*. The outlier factor of cluster C_i, $OF(C_i)$ is defined as *weighted sum* of distances between cluster C_i and the rest of clusters: $OF(C_i) = \sum_{j \ne i} |C_j| \cdot d(C_i, C_j)$.

The outlier factor $OF(C_i)$ measures the outlier degree of cluster, the bigger the value is, the bigger the possibility of being outlier cluster is.

4. Clustering-Based Outlier Detection Method

4.1 One-pass clustering algorithm

The goal of clustering is that the intra-cluster similarity is maximized while the inter-cluster similarity is minimized. Many efficient clustering algorithms have been proposed by the database research community. Clustering algorithm can be selected according to data, objective of clustering and application. In this paper, we use one-pass clustering algorithm [13] divide dataset into hyper spheres with almost the same radius. The algorithm is described as follows.

Step 1: Initialize the set of clusters, S, as the empty set, read a new object *p*.

Step 2: Create a cluster with the object p.

Step 3: If no objects are left in the database, go to step 6, otherwise read a new object p, and find the cluster C^* in S that is closest to the object p. In other words, find a cluster C^* in S, such that for all $\overline{C}$ in S, $d(p,C^*) \le d(p,\overline{C})$.

Step 4: If $d(p,C^*) > r$, go to step 2.

Step 5: Merge object p into cluster C^* and modify the *CSI* of cluster C^*, go to step 3.

Step 6: Stop

4.2 Outlier Detection Method

On the basis of the outlier factor of cluster, we present a clustering-based outlier detection method (*CBOD*), which consists of two stages.

Stage 1. Clustering: Cluster on data set D and produce clustering results $C = \{C_1, C_2, \cdots, C_k\}$.

Stage 2. Determining Outlier Clusters: Compute outlier factor $OF(C_i)(1 \le i \le k)$, sort clusters $C = \{C_1, C_2, \cdots, C_k\}$ and make them satisfy: $OF(C_1) \ge OF(C_2) \ge \cdots \ge OF(C_k)$. Search the minimum b, which satisfies $\frac{\sum_{i=1}^{b} |C_i|}{|D|} \ge \varepsilon (0 < \varepsilon < 1)$, finally, label clusters $C_1, C_2, \cdots, C_b$ with 'outlier' (any object belonged to outlier class is regarded as outlier), while $C_{b+1}, C_{b+2}, \cdots, C_k$ with 'normal'. (Any object belonged to normal class is regarded as normal).

4.3 Complexity Analysis

To simplify the analysis, we assume the final number of the clusters is k, categorical attribute D_i possess of distinct values n_i. For the clustering process, in the worst case, the time complexity of the clustering algorithm is $O(N \cdot k(\sum_{i=1}^{m_C} n_i + m_N))$ and it can be expected to be $O(N \cdot k \cdot m)$ [13].The second stage of the detection method computes the outlier factor of every cluster by calculating distance between any pair of clusters and sort clusters by the outlier factor of every cluster. The time complexity of computation distances is $O(k \cdot (k-1) \cdot (\sum_{i=1}^{m_C} n_i + m_N))$ in the worst case, and it can be expected to be $O(k \cdot (k-1) \cdot m)$. The time complexity of sorting k-clusters is $O(k \cdot \log k)$. Because of $k << N$, in the worst case, the time complexity of *CBOD* is $O(N \cdot k(\sum_{i=1}^{m_C} n_i + m_N))$, and approximates to $O(N \cdot k \cdot m)$.

It can be seen from above that the time complexity of *CBOD* is nearly linear with the size of dataset, the number of attributes and the final number of clusters. This implies the good scalability of our detection method. So it suit to detect outlier in large dataset.

4.4 The strategy to select parameters

- Selecting threshold r

Similar to the method used in literature [13], we present a sampling technique to determine threshold r, the details are stated as follows.

(1) Choose randomly N_0 pairs of objects in the dataset *D*.

(2) Compute the distances between each pair of objects.

(3) Computing the average EX of distances from (2).

(4) Select r in the range of [0.7EX, 0.9EX].

- Selecting parameter ε

The value ε is an approximate ratio of the outlier to the whole training dataset. If we don't know any thing about dataset, we select ε in the range of [0.05, 0.1]. We may determine ε more exactly based on the prior knowledge about the ratio of outlier objects to total objects in training dataset.

5. Experimental results

A comprehensive performance study is conducted to evaluate our algorithm. We test our algorithm on some real-life datasets obtained from the UCI Machine Learning Repository [14]. We use *detection rate* (DR) and *false alarm rate* (FR) to measure performance of the outlier detection methods. The detection rate is defined as the ratio of the detected outlier records to the total outlier records. The false alarm rate is defined as the ratio of the normal records detected as the outlier record to total normal records. *FindCBLOF*[11] and *TOD*[12] are also outlier detection methods containing two stages, which use outlier factor of object and need to scan dataset two pass. We demonstrate the effectiveness of our method against *FindCBLOF* and *TOD* algorithms, and give partial experiment data as follow.

5.1 Lymphography dataset

Lymphography dataset has 148 records with 18 categorical attributes. The records have been divided

into 4 classes: class 1 (with 2 records) and class 4 (with 4 records) are the rare class (outlier class), and the records in the rare class are regarded as outlier. By computing, we obtain *EX*=0.67. The table 1 gives the partial experiment data with different threshold r.

Table 1 Detect rare Records in Lymphography by *CBOD*

Threshold r	number of records (Top Ratio)	number of Rare records (DR)	number of normal records (FR)
0.62-0.54	6(4.05%)	6(100%)	0(0%)
0.53-0.45	5(3.38%)	5(83.33%)	0(0%)
	9(6.08%)	6(100%)	3(2.11%)

Table 2 demonstrates detection results in Lymphography dataset by *FindCBLOF and TOD*. From the experimental results, we can see that the performance of *CBOD* algorithm on Lymphography dataset outperformed that of *FindCBLOF* and *TOD*.

Table 2 Detect rare Records in Lymphography by *FindCBLOF* and *TOD*

FindCBLOF			*TOD*		
number of records (Top Ratio)	number of Rare records (DR)	number of ormal records (FR)	number of records (Top Ratio)	number of Rare records (DR)	number of normal cords (FR)
7(4.73%)	4(67%)	3(2.11%)	7(4.73%)	5(83.33%)	2(1.41%)
22(14.86%)	4(67%)	18(12.68%)	10(6.76%)	6(100%)	4(2.82%)
30(20.27%)	6(100%)	24(16.90%)			

5.2 Wisconsin Breast Cancer dataset

Wisconsin Breast Cancer (WBC) dataset has 699 records with 9 numerical attributes. Each record is labeled as benign (458 records) or malignant (241 records). We follow the experimental technique of Harkins S. [15] by removing some of the malignant records to form a very unbalanced distribution; the resultant dataset had 39 (8%) malignant records and 444 (92%) benign records. By computing, we obtain EX=0.2. The table 3 gives the partial experiment data with different threshold r.

Table 3 Detect Malignant Records in WBC by *CBOD*

Threshold r	number of records (Top Ratio)	number of Rare records (DR)	number of normal records (FR)
0.2	16(3.31%)	16(41.03%)	0(0%)
	48(9.94%)	38(97.44%)	10(2.25%)
	162(33.54%)	39(100%)	123(27.70%)
0.18	25(5.18%)	25(64.01%)	0(0%)
	48(9.94%)	38(97.44%)	10(2.25%)
	145(30.02%)	39(100%)	106(23.87%)
0.16	26(5.38%)	26(66.67%)	0(0%)
	48(9.94%)	38(97.44%)	10(2.25%)
	147(30.43%)	39(100%)	118(26.58%)
0.14	48(9.94%)	26(66.67%)	0(0%)
	44(9.11%)	38(97.44%)	6(1.35%)
	192(39.75%)	39(100%)	153(34.46%)

Table 4 shows the detection results in WBC dataset by *FindCBLOF* and *TOD*. The experimental results show that the performance of *CBOD* algorithm on Wisconsin Breast Cancer dataset is a little superior to that of *FindCBLOF* and *TOD*.

Table 4 Detect Malignant Records in WBC by *FindCBLOF* and *TOD*

FindCBLOF		
number of records (Top Ratio)	number of Malignant records (DR)	number of benign records (FR)
32(6.63%)	27(69.23%)	5(1.13%)
48(9.94%)	35(89.74%)	13(2.93%)
56(11.59%)	38(97.44%)	18(4.05%)
64(13.25%)	39(100.00%)	25(5.63%)
TOD		
number of records (Top Ratio)	number of Malignant records (DR)	number of benign records (FR)
33(6.83%)	31(79.49%)	2(0.45%)
40(8.28%)	35(89.74%)	5(1.13%)
44(9.11%)	36(92.31%)	8(1.80%)
50(10.35%)	38(97.44%)	12(2.70%)

5.3 KDDCUP99 Dataset

KDDCUP99 Dataset contains around 4,900,000 simulated intrusion records with 41 attributes (34 continuous and 7 categorical). We randomly produced a sub-dataset which consists of 38841 normal records and 1618 attack records. By computing, we obtain EX=0.23. Table 5 gives the partial experiment data with different threshold r. Table 6 shows the comparison results among different methods. The experimental results show that the detection results by *CBOD* outperformed those by Eskin, E [16].

Table 5 Detect Attack Records in KDDCUP99 by *CBOD*

Threshold r	number of records (Top ratio)	number of attack records (DR)	number of normal records (FR)
0.16	460(1.14%)	459(28.31%)	1(0.0026%)
	4032(9.97%)	1596(98.64%)	2437(6.04%)
0.18	460(1.14%)	459(28.31%)	1(0.0026%)
	3806(9.41%)	1596(98.64%)	2210(5.69%)
0.20	460(1.14%)	459(28.31%)	1(0.0026%)
	3691(9.12%)	1596(98.64%)	2095(5.39%)
0.22	460(1.14%)	459(28.31%)	1(0.0026%)
	3658(9.04%)	1594(98.52%)	2064(5.31%)

0.24	460(1.14%)	459(28.31%)	1(0.0026%)
	3635(8.98%)	1592(98.39%)	2043(5.26%)

Table 6 The contrast of results with different methods on dataset KDDCUP99

Ref.	Detection rate(DR)	False alarm rate(FR)
Eskin, E [16]	28%-93%	0.5%-10%
TOD	52.41%-98.33%	3.86%-5.00%
CBOD	28.31%-98.64%	0.00%-6.04%

6. Conclusions

In this paper, we generalize local outlier factor of object and propose a clustering-based outlier detection scheme. The theoretical analysis explains that the time complexity of *CBOD* is nearly linear with the size of dataset, the number of attributes and the final number of clusters, *CBOD* suit to detect outlier in large dataset. Finally, we give some experimental results to demonstrate the effectiveness.

Future work is directed at the following issues: (1) to investigate new distance definition to measure the difference among objects more accurately; (2) to devise new methods for labeling outlier classes to obtain more robust detection results.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.60673191),the Natural Science Research Programs of Guangdong Province's Institutes of Higher Education(No.06Z012) and Guangdong University of Foreign Studies Team Research Program of Innovations(No.GW2006-TA-005)

Reference

1. Hawkins, D.: Identification of Outliers, Chapman and Hall, London, 1980
2. Knorr, E. M., Ng,R. T. :Algorithms for Mining Distance-Based Outliers in Large Datasets. In: Proceedings of the 24th International Conference on Very Large Data Bases, New York, NY.1998, pp. 392-403
3. Ramaswamy,S., Rastogi,R., Kyuseok,S. :Efficient Algorithms for Mining Outliers from Large Data Sets. In: Proceedings of ACM SIDMOD International Conference on Management of Data.2000
4. Bay, S. D., & Schwabacher, M.:Mining distance based outliers in near linear time with randomization and a simple pruning rule. In: Proceedings of KDD03.2003
5. Arning,A., Agrawal,R., Raghavan,P.: A Linear Method for Deviation Detection in Large Databases. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press.1996, pp. 164-169
6. Breunig,M.M., Kriegel, H.P., Ng, R.T., Sander, J. :LOF: Identifying density-based local outliers. In: Proceedings of SIGMOD_00, Dallas, Texas.2000, pp.427–438
7. Jiang,S., Li,Q. ,Li,K.,Wang,H., Meng Z. :GLOF:A New Method for Mining Local Outlier. In: Proceedings of ICMLC2003,2003, pp. 157-162
8. Hu,T. , Sung,S. : Detecting pattern-based outliers. Pattern Recognition Letters. 2003, pp. 3059-3068
9. Jiang, M. F., Tseng, S. S., & Su, C. M. :Two-phase clustering process for outliers detection. Pattern Recognition Letters.2001, pp. 691–700
10. Yu, D., Sheikholeslami, G., & Zhang, A.: Findout: finding out outliers in large datasets. Knowledge and Information Systems.2002, pp. 387–412
11. He,Z. , Xu,X., Deng,S. : Discovering cluster-based local outliers Pattern Recognition Letters. 2003, pp.1651-1660
12. Jiang S., Li Q.,Wang H.,Zhao Y. A Two-stage Outlier Detection Method. MINI-MICRO SYSTEMS. 2005, pp. 1237-1240
13. Jiang,S., Xu,Y.: An efficient clustering algorithm. In: Proceedings of the ICMLC2004.2004, pp. 1513-1518
14. Merz,C.J., Merphy,P.:UCI repository of machine learning databases. URL: http://www.ics.uci.edu/ mlearn/ MLRRepository.html.1996
15. Harkins, S., He, H., Willams, G.J., Baster, R.A.: Outlier detection using replicator neural networks. In: Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery, Aix-en-Provence, France.2002, pp. 170–180
16. Eskin,E., Arnold,A., Prerau,M., Portnoy, L. and Stolfo, S. :A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In Data Mining for Security Applications.2002