A Fast Audit Doubt Finding Model

Quan Wan\textsuperscript{1,2}, Weibo Li\textsuperscript{1,2,*}, Hairong Wang\textsuperscript{1,2}, Hua Yan\textsuperscript{1,2} and Rui Xiang\textsuperscript{1,2}

\textsuperscript{1}Hubei Key Laboratory of intelligent robot (Wuhan Institute of Technology)
\textsuperscript{2}School of computer science and engineering, Wuhan Institute of Technology, Wuhan, China

*Corresponding author: weiboli@wit.edu.cn

Abstract. In order to improve the audit efficiency and accuracy under the condition of big data, an audit doubt discovery model (CLOWF) based on an adaptive clustering outlier detection algorithm is proposed. The model first uses the slope ratio method to obtain the K value of the K-means++ algorithm, thereby effectively ensuring the clustering effect; At the same time, for the problem of slow pruning speed, a parallel calculation method for the data radius and centroid of each cluster is proposed and designed; Combined with the weighted local outlier factor detection algorithm to calculate the outlier factor, the data greater than the epsilon threshold of outlier is proposed as an indicator to determine the audit doubt. Through auditing application cases, comparing the local outlier factor detection algorithm and the cluster-based outlier detection algorithm, the results show that the execution time of the CLOWF model is the shortest and the accuracy rate is 97.3%.

Keywords: Audit doubts, outliers, cluster analysis, K-means++, LOF.

1. Introduction

In the era of big data, the traditional method of discovering audit doubts consumes a lot of energy, and the effect of sampling may not be valuable. It is a new challenge for computer auditing to quickly and effectively help auditors find audit doubts. Yong Wen [1] and others proposed that outlier detection can quickly locate suspicious data. Ping Cheng [2] uses the density clustering algorithm (Density-Based Spatial Clustering of Applications with Noise, DBSCAN) for clustering, storing high-value clustering results in the audit knowledge base and realizing reuse. Literature [3-5] proposed a clustering algorithm based on K-means to realize the application of outlier analysis in the discovery of audit doubts and improve the efficiency of audit. Lijuan Lu [6] and others based on the three-layer MapReduce parallelized random forest algorithm to predict the probability of audit doubts, the algorithm improves work efficiency. Kuna H.D. [7] and others proposed the application of outlier detection algorithms in audit logs, combined with different data mining algorithms to achieve outlier detection, so as to facilitate the work of auditors.

Outlier analysis is the discovery of abnormal data separated from pattern clusters. Among them, the density-based local outlier factor detection algorithm (LOF) and the cluster-based K-means algorithm are widely used. In view of the low efficiency of the LOF algorithm for big data and the difficulty of
determining the K value of the K-means algorithm, which leads to the high complexity of outlier data detection and poor detection quality, scholars have made some improvements. Jing Tao [8] and others proposed a Cluster-based and LOF Outlier Detection Method (CLOF) algorithm that combines K-means clustering and LOF. The algorithm first uses the K-means algorithm to filter out the initial outlier data from the source data, and then obtains the final outlier data through the LOF algorithm. Literature [9-11] combines DBSCAN algorithm with LOF algorithm and its improved algorithm to detect outliers, which improves the detection efficiency and accuracy to a certain extent. Yang P [12] and others proposed to use an optimized self-organizing feature map algorithm to cluster the data set, so that the calculation of the outlier factor of the data point only needs to be performed in a small cluster. Experiments show that the accuracy and execution time of anomaly detection both prioritize the LOF algorithm.

2. Related theoretical basis

2.1. K-means++ algorithm

In 1967, MacQueen [13] used the K-means algorithm for the first time, after which David Arthur and others proposed the K-means++ algorithm for the random selection of K-means initial cluster centers. Assuming that the data set X contains K clusters \(x_1, x_2, \ldots, x_k\), the number of samples in each cluster is \(n_1, n_2, \ldots, n_k\), and the cluster centers of each cluster are \(c_1, c_2, \ldots, c_k\).

**Definition 1.** Error function (E): Refers to the sum of squares of distances from each data point \(p\) to the respective clustering center \(c_i\), that is, the sum of squares of grouping errors. The formula is as follows:

\[
E = \sum_{i=0}^{K} \sum_{p \in x_i} ||p - c_i||^2
\]

The paper uses formula (1) as the error function to evaluate the performance of K-means++ clustering algorithm.

2.2. Local outlier factor detection algorithm, LOF

The LOF algorithm was first proposed by Breunig [14] and others in 2000. By assigning an outlier factor LOF of neighborhood density to each data point. If outlier factor LOF value close to 1, it is considered a normal point, otherwise it is an abnormal point. Reference to the literature [15], the weighted distance is introduced to calculate, the idea is to highlight the outlier attributes and improve the accuracy of outlier detection.

**Definition 2.** For a given data set \(D\), \(p\) is any data object in the data set, \(N_k(p)\) represents the k distance neighborhood of data object \(p\), then the local density of data object \(p\) is defined as follows:

\[
Aw_k(p) = \left[ \frac{\sum_{\Delta \in N_k(p)} d(p, o)}{N_k(p)} \right]^{-1}
\]

The calculation formula of \(d(p, o)\) is as follows:

\[
d(p, o) = [\sum_{i=1}^{n} \Delta(A_i) \ast d(p, o_i)]
\]

The \(Aw_k(p)\) introduced the definition of the local outlier factor in the LOWF algorithm: The distance-weighted local outlier factor of the data object \(p\), the \(LOWF_k(p)\) is expressed as the local reachability of the neighboring points of the object \(p\) and the average of the ratio of the density to the locally accessible density of the object \(p\).

**Definition 3.** Local outlier factor of object \(p\):
The closer the value of \( \text{LOWF}_k(p) \) is to 1, it means that \( p \) may belong to the same cluster as its neighbors; the greater the value of \( \text{LOWF}_k(p) \) is greater than 1, it means that the density of \( p \) is lower than the density of neighboring points, and \( p \) may be an abnormal point, otherwise \( p \) is a dense point.

3. Audit doubt discovery model based on adaptive clustering outlier detection algorithm

3.1. Adaptive K-Means++ algorithm

In the K-means++ algorithm, the number of clusters \( K \) needs to be determined in advance, and there may be problems: Such as poor clustering effect and difficult selection of \( K \) value. In response to the above problems, the paper proposes an adaptive K-means++ algorithm, which uses the slope ratio to calculate the number of clusters \( K \) to ensure a better clustering effect; in addition, for the clusters obtained by clustering. Open three processes to process three clusters at the same time, and the denser data points in the cluster are deleted, thereby improving the efficiency to a certain extent, and quickly obtaining the initial outlier data set.

3.1.1. Determination of the number of clusters \( K \)

Refer to the slope ratio method proposed by Shuo Zhang [16] and others to obtain the best \( K \) value, so as to solve the problem of difficult determination of \( K \) value. In the paper, the slope ratio method is applied to the K-means++ algorithm, the \( E \) value in the K-means++ algorithm is calculated, and the \( K \) value of the error function convergence boundary is used as the number of clusters.

Definition 4. Slope ratio (S): Refers to the product of the ratio of the error function slope difference of each adjacent \( K \) value to the error function and the reciprocal of \( \Delta K \), and \( \Delta K = K_{i-1} - K_i \), the formula is as follows:

\[
S_i = \frac{E_{i-1} - E_i}{E_{i-1}(K_{i-1} - K_i)}
\]

Among them, \( K_i \) is the i value in the \( K \) value set; \( S_i \) is the slope ratio corresponding to \( K_i \). The \( i \) value in formula (6) is the best \( i \) value, and \( K_i \) is the best \( K \) value.

3.1.2. Parallel processing

K-means++ algorithm is used to obtain \( K \) clusters. The parallel processing flow is shown in Fig.1. The density threshold \( m \) is set and three processes are started. Parallel calculations are performed on the data of three clusters at the same time: If the number of clusters is less than the density threshold, it indicates that the amount of data of this type is small, and all are included in the initial outlier data set; if the number of clusters is greater than the density threshold, each class radius \( R \) and centroid \( X_0 \), the points whose distance from the centroid is greater than the radius \( R \) are included in the initial outlier data set. Calculate in parallel until all clusters are executed, and get the initial outlier data.

Definition 5. Cluster centroid (\( X_0 \)): \( X_i \) refers to the number of data objects in class \( i \), and the number of data objects in class \( n_i \). The formula is as follows:

\[
X_0 = \frac{\sum_{i=1}^{n} X_i}{n_i}
\]

Definition 6. Clustering radius (\( R \)): \( X_i \) refers to the data object in class \( i \), \( X_0 \) refers to the cluster centroid in class \( i \), \( n_i \) the number of data objects in class \( i \), the formula is as follows:
Definition 7. Class center distance (\(d_i\)): \(X_{ij}\) refers to the data object in class \(i\), \(X_0\) refers to the cluster centroid in class \(i\), and \(p\) refers to the dimension of the data. The formula is as follows:

\[
R = \sqrt{\frac{\sum_{i=1}^{n} (X_{ij} - x_0)^2}{n_i}}
\]  

\(d_i = \sqrt{\sum_{j=1}^{p} (x_{ij} - x_0)^2}\)  

The closer the value of \(\text{LOWF}_k(p)\) is to 1, it means that \(p\) may belong to the same cluster as its neighbors; the greater the value of \(\text{LOWF}_k(p)\) is greater than 1, it means that the density of \(p\) is lower than the density of neighboring points, and \(p\) may be an abnormal point, otherwise \(p\) for dense points.

3.2. Audit doubt discovery model

3.2.1. CLOWF-based audit doubt discovery process. In order to find audit doubts, the paper proposes an audit doubt discovery model (Clustering-based and LOF based on weighted distance Outlier Detection Method, CLOWF) based on adaptive clustering outlier detection algorithm, as shown in Fig.2.
(1) Data pre-process: Process the audit data of the test, so that the data is input to the test module according to a unified standard. The processing methods are as follows: 1) Data cleaning: For the default values, the article adopts the delete method to deal with, and at the same time filters the redundant fields, leaving only the field information needed in the mining. 2) Data integration: Due to the existence of duplicate data, the article de-duplicates the data.

(2) Adaptive K-means++ algorithm: Cluster the pre-processed audit data, the initial value of K is set to 2, the change step is 1, and 31 K values are obtained. Use formula (1) to compare these 31 K value calculation errors, and then use the equation (5) and equation (6) calculate the optimal number of clusters K, then use the K-means++ algorithm to obtain K clusters, set the density threshold, start three processes, and delete the dense points in each cluster in parallel. Get the initial outlier data set X.

(3) LOF based on weighted distance (LOWF): For the initial outlier data set X, use formula (3) to introduce weighted distance, and calculate formulas (2) and (4) in turn, if it is greater than the outlier threshold, it is judged as an outlier.

3.2.2. Based on CLOWF algorithm description.

Algorithm: CLOWF based on improved K-means++ and LOWF

Algorithm 1: Improved K-means++
Input: Dataset D, i // Data set D, the number of clusters is i
Output: K // Number of clusters K
for (i < D. size()) calculate Ei; calculate Si; get K
Input: Dataset D, K // Data set D, the number of clusters K
Output: K_clusters // K cluster
K-means++. fit (dataset D); get K_clusters

Algorithm 2: Extract preliminary outliers X // Initial outlier data set X
Input: K_clusters, m //K cluster, density threshold m
Output: Preliminary outliers
for (i <= K) if (n(i) <= m) get preliminary outliers; else {
process p1; calculate center t1, R1; jEi_clusters;
if (dist(dj) >= R1) get preliminary outliers;
process p2; calculate center t2, R2; jEi+1_clusters;
if (dist(dj) >= R2) get preliminary outliers;
process p3; calculate center t3, R3; jEi+2_clusters;
if (dist(dj) >= R3) get preliminary outliers;
}

Algorithm 3: LOWF algorithm
Input: Preliminary outlier set X, k, epsilon // The initial outlier data set X, the number of neighbors k, the outlier threshold epsilon
Output: Outliers
calculate \( \Delta(A_i) \);
for (i< X.size())
calculate the dist(\( x_i \)); calculate \( AW_k(x_i) \); calculate \( LOWF_k(x_i) \);
if (\( LOWF_k(x_i) > \) epsilon) output xi; get outliers

4. Application result analysis
The paper uses the audit case data set to compare and analyze the execution efficiency and the accuracy of outlier data detection of LOF algorithm, CLOF algorithm and CLOWF algorithm. Experimental environment: Operating system: Ubuntu-16.04 LES system, CPU: i5-8300H, memory 8GB, development language: Python.
4.1. Evaluation index
In order to verify the effectiveness of CLOWF algorithms, different algorithms were used in the same audit data set for comparative experiments, using accuracy (ACC), detection rate (FDR), false alarm rate (FAR) and Execution Time (ET) is used as an evaluation index of algorithm performance.

\[
\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN}
\]

\[
\text{FDR} = \frac{TP}{TP + FN}
\]

\[
\text{FAR} = \frac{FP}{TN + FP}
\]

Among them, the definitions of the evaluation indexes of TP, TN, FP, and FN are shown in Table 1.

| Evaluation index | Definition |
|------------------|------------|
| TP (True Positive) | The number of abnormal marked as abnormal |
| TN (True Negative) | The number of normal marked as normal |
| FP (False Positive) | The number of normal marked as abnormal |
| FN (False Negative) | The number of abnormal marked as normal |

4.2. Actual audit case analysis
The audit case data set comes from the audit business data of an audit company from Wuhan. In order to verify the validity of the model, the paper uses the data audited by the auditors as a data set, a total of 185 pieces of data, due to space limitations, the data is micro-processed, as shown in Table 2.

| Event id | Event name | Event RA E functional area | Event created date | Event start date | Meeting participants | Audit results |
|----------|------------|----------------------------|--------------------|------------------|---------------------|--------------|
| 1        | t1         | f1                         | 26/03/2019         | 14/04/2019       | 18                  | 0            |
| 2        | t2         | f2                         | 10/09/2019         | 14/11/2019       | 0                   | 1            |
| 3        | t3         | f3                         | 11/04/2019         | 24/04/2019       | 13                  | 0            |
| 4        | t4         | f4                         | 12/10/2019         | 30/11/2019       | 49                  | 1            |
| 5        | t2         | f2                         | 12/09/2019         | 01/10/2019       | 96                  | 1            |
| 7        | t5         | f5                         | 27/03/2019         | 19/04/2019       | 12                  | 0            |
| 8        | t6         | f1                         | 01/04/2019         | 17/04/2019       | 15                  | 0            |

The paper uses the number of days between the creation time and the start time of the conference and the participants as two-dimensional data for outlier detection. Verify the accuracy and execution time of CLOWF algorithm for outlier detection of audit data, and compare and analyze the accuracy and execution time of different algorithms by simulating different amounts of data.

4.2.1. Verification accuracy. Under the evaluation indicators of ACC, FDR and FAR, compare the accuracy rates of LOF algorithm, CLOF algorithm and CLOWF algorithm on 185 pieces of data with audit results, as shown in Table 3.

| Evaluation index | LOF accuracy/ % | CLOF accuracy/ % | CLOWF accuracy/ % |
|------------------|-----------------|------------------|-------------------|
| ACC              | 94.05           | 96.76            | 97.30             |
| FDR              | 84.21           | 89.47            | 94.74             |
| FAR              | 4.82            | 2.41             | 2.41              |
It can be seen from Table 3 that comparing the LOF algorithm, CLOF algorithm, and CLOWF algorithm. The CLOWF algorithm introduces attribute weights and optimizes the calculation of distance and outlier degree, so the accuracy rate is the highest.

4.2.2. Verify timeliness. Under different data scales, compare the operating efficiency of LOF algorithm, CLOF algorithm and CLOWF algorithm, as shown in Fig.3.

![Figure 3](image)

**Figure 3.** Experimental comparison of execution time under different data scale.

It can be seen from Fig.3 that as the amount of data increases, the execution time of the LOF algorithm increases sharply. When the data reaches 50,000 pieces of data, the amount of calculation of the LOF algorithm increases, resulting in insufficient physical memory. When the amount of data reaches 100,000 pieces of data, compared with the CLOF algorithm, due to the increase in the amount of pruning data, the parallel algorithm improves the pruning speed, and the CLOWF execution time is the shortest.

4.2.3. Resource consumption comparison. Under different data scales, compare the proportion of physical memory usage, as shown in Table 4.

| Data volume/piece | LOF consumption rate/ % | CLOF consumption rate/ % | CLOWF consumption rate/ % |
|-------------------|-------------------------|--------------------------|---------------------------|
| 185               | 1.2                     | 1.8                      | 2.1                       |
| 8335              | 7.8                     | 1.9                      | 2.3                       |
| 15000             | 44.3                    | 3.2                      | 3.6                       |
| 30000             | 91.9                    | 7.6                      | 10                        |
| 40000             | 93.3                    | 16.8                     | 19.1                      |
| 50000             | 94.6                    | 34.9                     | 32.4                      |
| 60000             | Memory overflow         | 51.6                     | 48.6                      |
As shown in Table 4, as the amount of data reaches 50,000 pieces of data, LOF algorithm requires a lot of calculations, resulting in a high proportion of physical memory; CLOF algorithm and CLOWF algorithm prune the data to reduce the data size, and the proportion of physical memory is medium.

4.2.4. Analysis of experimental results. In the case of 185 pieces of data, set the LOF algorithm's nearest neighbor parameter \( k=8 \), outlier threshold \( \epsilon=1.5 \), CLOF algorithm and CLOWF algorithm pruning density threshold \( m=8 \), neighbor parameter \( k=8 \), outlier threshold \( \epsilon=1.5 \), the detection result is shown in Fig.4.

![Figure 4. Running results: (a) LOF algorithm, (b) CLOF algorithm, (c) CLOWF algorithm.](image)

The above experimental results show that comparing the LOF algorithm and the CLOF algorithm, when the amount of data is close to 100,000, the time efficiency of the CLOWF algorithm is significantly improved, and the accuracy of the audit doubt discovery has been improved to a certain extent.

5. Conclusions

The paper first uses the adaptive K-means++ algorithm to cluster the data to extract the preliminary outlier data, and then uses the LOWF algorithm to calculate the outlier degree factor, and uses the data greater than the epsilon threshold as an audit suspicious point judgment index, thereby proposing a fast audit doubt discovery model. Combining audit application cases, comparing LOF algorithm and CLOF algorithm, CLOWF algorithm effectively improves the efficiency and accuracy of audit doubt discovery.

Acknowledgments

The authors wish to thank Prof. Weibo Li for experimental guidance and audit data provided by Grant Thornton International Ltd (Wuhan). This work was also financially supported by Hubei Province University Industry-University-Research Cooperation Key Funding Project (C2010033) fund.

References

[1] Yong Wen, Application of data mining in risk-oriented auditing [J], China Internal Audit, 2013 (06): 61 - 63.
[2] Ping Cheng, Shan Chen, Audit sampling based on DBSCAN clustering method in the era of big data [J], Chinese Certified Public Accountant, 2016 (04): 76 - 79.
[3] Xu Chen, Hongping Chen. Discussion on the application of outlier analysis in the discovery of audit doubts——The Python implementation based on the K-Means clustering algorithm [J], Friends of Accounting, 2019 (17): 72 - 76.
[4] Yuefan Sun, Di Song. Audit sampling method based on outlier analysis in big data environment [J], Finance and Accounting, 2015 (14): 71 - 73.
[5] Yuan Chang, Nana Kang. Application of K-means clustering in the discovery of audit doubts[J],
Chinese Certified Public Accountant, 2018 (05): 84 - 86.

[6] Lijuan Lu, Congrong Yu, Donggui Liang, Weizheng Zhang. Audit big data doubtful prediction based on parallel random forest [J], Computer and Digital Engineering, 2019, 47 (01): 174 - 179.

[7] Kuua HD, Garcia Martinez, R, et al, Outlier detection in audit logs for application systems [J], Information Systems, 2014, 44: 22 - 33.

[8] Jing Tao, Outlier detection method based on clustering and density [D], South China University of Technology, 2014.

[9] Jinghua Wang, Xinxian Zhao, Guoyan Zhang, et al, NLOF: A new density-based local outlier detection algorithm [J], Computer Science, 2013, 40 (08): 181 - 185.

[10] Ben L V, Mao Ting G. Improvement of detection algorithm density-based local outlier [J], Modern Computer, 2015.

[11] Xue Xiao, Shanliang Xue, Outlier data detection algorithm based on improved OPTICS clustering and LOPW [J], Computer Engineering and Science, 2019, 41 (05): 885 - 892.

[12] Yang P, Wang D, Wei Z, et al, An outlier detection approach based on improved self-organizing feature map clustering algorithm [J], IEEE Access, 2019, 7: 115914 - 115925.

[13] Macqueen J, Some methods for classification and analysis of multivariate observations[C]// Proc of Berkeley Symposium on Mathematical Statistics & Probability, 1965.

[14] Breunig M M, Kriegel H P, Ng R T, et al, LOF: Identifying density-based local outliers[C]. //Acm Sigmod International Conference on Management of Data, ACM, 2000.

[15] Peiyong Duan, Chong Cui, Jiejue Zhang, An improved local outlier data detection algorithm[J], Journal of Natural Science of Heilongjiang University, 2017, 34 (04): 474 - 480.

[16] Shuo Zhang, Xin Jin, Zhaofeng Li, Jian Gao, Outlier detection algorithm based on grid LOF and adaptive K-means [J], Command Information Systems and Technology, 2019, 10 (01): 90 - 94.