Taylor & Francis
Taylor & Francis Group

Quality Technology & Quantitative Management

ISSN: (Print) 1684-3703 (Online) Journal homepage: https://www.tandfonline.com/loi/ttqm20

Distributed outlier detection in hierarchically structured datasets with mixed attributes

Qiao Liang & Kaibo Wang

To cite this article: Qiao Liang & Kaibo Wang (2020) Distributed outlier detection in hierarchically structured datasets with mixed attributes, Quality Technology & Quantitative Management, 17:3, 337-353, DOI: 10.1080/16843703.2019.1629679

To link to this article: https://doi.org/10.1080/16843703.2019.1629679

Published online: 21 Jun 2019.

QUALITY TECHNOLOGY & QUANTITATIVE MANAGEMENT
2020, VOL. 17, NO. 3, 337–353
https://doi.org/10.1080/16843703.2019.1629679

Taylor & Francis
Taylor & Francis Group

ARTICLE

Distributed outlier detection in hierarchically structured datasets with mixed attributes

Qiao Liang and Kaibo Wang

Department of Industrial Engineering, Tsinghua University, Beijing, China

ABSTRACT
Anomaly detection has been extensively studied over the past decades; however, there are still various challenges due to the complex structures of the real-world datasets. First, only a few methods in the literature provide insight into the datasets that have both categorical and continuous attributes, and even fewer of them are sensitive to the dependencies between the two types of attributes. Second, a real-world dataset tends to be more complex in its structure, and the categorical attributes are usually hierarchically correlated, which has been largely ignored by the existing outlier detection approaches. Following this line of reasoning, we propose a distributed outlier detection method for mixed attribute datasets, especially with hierarchical categorical attributes. The proposed method accounts for the dependencies between categorical and continuous attributes rather than treating them as two separate parts. In addition, the proposed method is able to capture the hierarchical structure among categorical attributes. The experimental results on a real-world dataset and a simulation study show its superior performance in terms of both the detection accuracy and time efficiency.

ARTICLE HISTORY
Accepted 30 May 2019

KEYWORDS
Anomaly detection; hierarchical structures; mixed attribute datasets; outlier detection

1 Introduction

Anomaly detection, or outlier detection, has recently attracted increasing attention with its applications in various fields such as fraud detection (Bolton & Hand, 2002), network intrusion detection (Lazarevic, Ertoz, Kumar, Ozgur, & Srivastava, 2003), clinical diagnosis (Penny & Jolliffe, 2010), and detection of abnormal human activities in sensor network (Nivetha & Venkatalakshmi, 2018). According to the definition from Barnett and Lewis (1994), outliers are those observations that appear to be inconsistent with the remaining portion of the dataset. Outlier detection methods focus on finding the rules or the patterns of how outliers deviate markedly from the other points in the datasets. Many models have been built, and a large number of techniques have been developed for solving this problem. However, the complexity and variety of real-world datasets make it far more challenging to present the data structure accurately using the existing models.

First, most of the research efforts are concentrated on the datasets that have only one type of attribute, i.e. only categorical attributes or only numerical attributes. However, there is an increasing number of cases in which more than one type of attribute is in supervised control. It is even more challenging to address anomaly detection in mixed attribute datasets, and there are fewer prior studies; one reason is that the dependencies among the categorical attributes, among the numerical attributes and between the two types of attributes must all be captured. In this

CONTACT Kaibo Wang kbwang@tsinghua.edu.cn Department of Industrial Engineering, Tsinghua University, RM 612, Shunde Building, Beijing 100084, China

paper, we present an anomaly score function that accounts for all three types of dependencies among the attributes based on the previous outlier score definitions in Otey, Ghoting, and Parthasarathy (2006), Koufakou and Georgiopoulos (2010).

The second issue is the complex structure of the currently available datasets. Most of the attributes are mutually related or even hierarchically linked to one another, e.g. 'Area' and 'Country' are two categorical attributes with obvious hierarchical correlation, and they consist of information that is partially overlapping. The hierarchical property among the categorical attributes in the datasets is worthwhile for further utilization and deep mining into the data.

The primary objective of this research is to propose a fast outlier detection approach to address the challenges above. Specifically, we define an anomaly score function for measuring the degree of inconsistency of each data point. The proposed outlier detection method is able to capture the dependencies between mixed attributes in datasets. Moreover, it can be easily applied to complex-structured datasets with multi-level categorical attributes. We use the MapReduce programming model and Hadoop infrastructure for the distributed model implementation. The proposed algorithm is compared with two state-of-the-art distributed outlier detection methods for mixed datasets proposed by Otey et al. (2006) and Koufakou and Georgiopoulos (2010). The experimental results show that our method, resulting from its utilization of the hierarchical property in the categorical space, exhibits an overall higher detection rate and better time efficiency.

The remainder of this article is organized as follows. In Section 2, we provide an overview of the previous approaches to outlier detection. In Section 3, we present our outlier detection algorithm. Section 4 and Section 5 give the experimental results on a real-world dataset and a simulated dataset, respectively. The overall summary and discussion about future research are provided in Section 6.

2 Related work

The original outlier detection approaches were derived from statistical methods, which judged a potential outlier mainly based on a fitted distribution (Hodge & Austin, 2004). These distribution-based methods are limited because they are suitable only for low dimensional spaces, and a prior knowledge of the data distribution is unachievable in some cases. Distance-based outlier detection methods such as k-NN (Knorr & Ng, 1998; Yu, Luo, Chen, & Bian, 2016) detect a potential outlier based on its distance to the other neighbors under a specific distance metric. Several other mainstream approaches for outlier detection include density-based methods (e.g. LOCI (Papadimitriou, Kitagawa, Gibbons, & Faloutsos, 2002) and LOF (Breunig, Kriegel, Ng, & Sander, 2000)), depth-based methods (Ruts & Rousseeuw, 1996), and clustering-based methods (e.g. RB-MINE (Fan, Zaiane, Foss, & Wu, 2006)). In general, these methods were designed originally for continuous or numerical datasets and cannot be directly applied to scenarios with categorical or even mixed (categorical and numerical) datasets. Furthermore, the approaches based on distance computation between points might be inappropriate to implement in a distributed setting.

There have been several types among approaches for outlier detection in the categorical attribute datasets. The first type of the mainstream methods is the proximity-based method (Hodge & Austin, 2004), which measures the nearness of objects in terms of the distance or density, similar to the methods mentioned above. These methods usually define a specific distance metric to measure the proximity between categorical pairs. For example, a method called ORCA (Bay & Schwabacher, 2003) employed the Hamming distance as distance measurement, and the CNB method (Li, Lee, & Lang, 2007) employed a common-neighbor-based distance function to measure the proximity between a pair of data points.

The second type of outlier detection method for categorical datasets focuses on the frequent pattern or association rules in a dataset. These methods first learn the normal behavior of a system through frequent pattern mining (Han, Cheng, Xin, & Yan, 2007), and then, they label the points that deviate markedly from normality as outliers. For example, the FIM algorithm (He, Xu, Huang, & Deng, 2005) defined a Frequent Pattern Outlier Factor (FPOF) to identify the outlier objects from categorical data. The algorithm proposed by Pai et al. introduced the concept of relative pattern discovery from the new perspective of association analysis and proposed an unsupervised approach to evaluate which observations are anomalous based on the knowledge of relative patterns (Pai, Wu, & Hsueh, 2014). The AVF method (Koufakou, Ortiz, Georgiopoulos, Anagnostopoulos, & Reynolds, 2007; Koufakou, Secretan, Reeder, & Cardona, 2008) defined a specific outlier score for each data point based on the infrequent degree of its attribute values, which allowed better computational efficiency. Other similar methods (Das & Schneider, 2007; Narita & Kitagawa, 2008) detected anomalies according to the association rule between itemsets, meanwhile presenting a remarkable efficiency in speeding up the detection.

Another type of outlier detection approaches for categorical datasets came from the concept of entropy (Lee & Xiang, 2001; Wu & Wang, 2013), which detected outliers through measuring the decrease in entropy after eliminating a point from the dataset. All of the methods above offered us a method for addressing outlier detection in categorical datasets and could be further geared to the same problem in mixed datasets.

Inspired by the previous methods for categorical datasets, there have been some research efforts on outlier detection in datasets with both categorical and numerical attributes. A graph-based outlier detection algorithm (Yu, Qian, Lu, & Zhou, 2006) was proposed to calculate outlier indicators by computing the Euclidean distance for numerical values and Hamming distances for categorical values. Another typical method, LOADED (Ghoting, Otey, & Parthasarathy, 2004), also adopted this ‘divide and conquer’ idea. It used association rules to explore infrequent items among categorical values and calculated the covariance matrix to examine the anomalies in the numerical values. The method proposed by Chen, Miao, and Zhang (2010) employed a new definition of traditional distance metrics by considering neighborhood information and gave a neighborhood-based algorithm to detect the outliers. All of these methods detect anomalies in numerical and categorical values separately, without considering the interactions between the two types of attributes. They are advantageous at detecting global outliers, but they tend to fail in detecting contextual outliers, which deviate significantly from the remaining data points in a subset of data, while looking normal globally (Wang & Davidson, 2009).

To overcome the drawbacks above, some other methods were proposed to obtain better performance on contextual outlier detection. For example, the POD method (Zhang & Jin, 2010) employed a logistic regression to learn the patterns that represent the interactions between different types of attributes in the majority of data and formulated a Mixed-Attribute Data Outlier Factor (MADOF) to quantify the anomaly in mixed attribute datasets. For distributed data, two state-of-the-art distributed outlier detection methods (Koufakou & Georgiopoulos, 2010; Otey et al., 2006) both defined an anomaly score that captured the dependencies between the numerical and categorical attributes. The two methods calculated the first part of the anomaly score based on frequent itemset mining in categorical attribute space, and then, they obtained the second part of the score by focusing on each of the subsets in the mixed attribute space. The calculation was conducted independently on each site in the first stage, and then, the results were combined together to build a global model. However, to the best of our knowledge, all of the existing methods have ignored the scenarios that have hierarchical structure; in other words, they tend to put all of the attributes on an equal status and give them equal weights, which is obviously inappropriate when the attributes are hierarchically linked and share multi-level information.

3 Outlier detection algorithm

Inspired by the two outlier detection methods (Koufakou & Georgiopoulos, 2010; Otey et al., 2006), we plan to define an outlier score function for each data point to reflect its degree of irregularity. The outlier score includes a categorical part and a continuous part, which correspond

Table 1. Notation.

Term	Definition
$\mathcal{D}$	Dataset
n	Number of data points in $\mathcal{D}$
i	The index of a data point
$\mathbf{x}_i$	The i-th data point in $\mathcal{D}$
$\mathbf{x}_i^C$	The categorical part of $\mathbf{x}_i$
$\mathbf{x}_i^N$	The numerical part of $\mathbf{x}_i$
d	Itemset (sets of categorical attributes and values)
$\lvert d\rvert$	Size or length of itemset d
$MAXLEN$	Maximum length of itemset d
$supp(d)$	Support (frequency) of itemset d
σ	Minimum support of itemset d
$\mathcal{I}$	Set of all possible combinations of attributes and their values (distinct single items) in $\mathcal{D}$
a	A numerical attribute
V_i^a	The deviation of point $\mathbf{x}_i$ on attribute a
μ_a^d	The mean value of attribute a in all of the points that contain itemset d
$Range_a^d$	The range of attribute a in all of the points that contain itemset d
δ_a	Maximum deviation on attribute a

to the two types of attribute spaces. The anomaly could arise from irregular values of categorical attributes or continuous attributes and from breaking dependencies between the two types of attributes. The proposed method can be easily implemented in a distributed way. Table 1 shows the notation used in this article.

3.1 Categorical score in previous work

In the first stage of our method, we offer each point a categorical outlier score (or type 1 outlier score) based on its abnormality in categorical attribute space. The basic idea is that if the attribute-value pairs in a point co-occur in a more infrequent way, then the point is going to obtain a higher categorical score. In this subsection, we start with introducing the definition of categorical score from the previous ODMAD model by Koufakou and Georgiopoulos (2010), and our score definition will follow this frame and form. To address hierarchically structured datasets, we propose a modified categorical score in Section 3.2 to make more use of the hierarchical information.

Consider a dataset $\mathcal{D}$ that contains n data points, $\mathbf{x}_i$, $i = 1, \ldots, n$. We denote the categorical part of $\mathbf{x}_i$ by $\mathbf{x}_i^C$ and the numerical part by $\mathbf{x}_i^N$. Calculating the categorical score for each point is derived from the idea of *Frequent Itemset Mining* (Agrawal & Srikant, 2014). Let $\mathcal{I}$ be the set of all possible combinations of attributes and their corresponding values in dataset $\mathcal{D}$. Then, let $\mathcal{S}$ be the set of itemsets d, where an attribute occurs only once in one itemset.

$$\mathcal{S} = \{d | d \in \mathrm{PowerSet}(\mathcal{I}) \wedge d_i.attribute \neq d_j.attribute \forall i, j, i \neq j\}.$$

Thus, we can define the categorical outlier score value for a point $\mathbf{x}_i$ in categorical attribute space as

$$Score_1(\mathbf{x}_i) = \sum_{d \subseteq \mathbf{x}_i \wedge supp(d) < \sigma \wedge |d| \leq MAXLEN} \frac{1}{supp(d) \times |d|}, \tag{1}$$

where $|d|$ represents the length of itemset d, and it is defined as the number of attribute-value pairs in d. The frequency or support of itemset d, $supp(d)$, is defined as the number of data points in dataset $\mathcal{D}$ that contain itemset d. Additionally, σ is a user-defined threshold, and those itemsets for which $supp(d)$ is less than σ are infrequent itemsets. $MAXLEN$ is a user-entered maximum length of itemset d. The score function indicates that a point is more likely to be an outlier if it contains more single values or sets of values that are infrequent. Generally, the occurrence of infrequent attribute-value pairs in a data point shows its abnormality.

Equation (1) implies that we have to go through each infrequent itemset of a data point before calculating its categorical outlier score. However, the efficiency could be improved by using some data mining algorithms such as *Apriori* (Agrawal & Srikant, 2014). The number of itemsets can be efficiently pruned according to the *Apriori* property as follows:

Lemma 1. *If* $supp(d) < \sigma$, *then* $supp(d') < \sigma \; \forall d' \supseteq d$.

Under this property, instead of considering all of the itemsets from length 1 up to *MAXLEN*, we compute only the frequencies of certain categorical itemsets, as follows: if we find that itemset d in point $\mathbf{x}_i$ is infrequent; then, we will not consider any of d's supersets. By making use of the *Apriori* property in itemset mining, valued information is retained to detect an anomaly while the overlapping part of the information is pruned to avoid redundant calculations.

The *Apriori* property indicates that a border exists in this downward itemset mining process. Let *Negative Border* ($\mathcal{NB}$) be a set of those infrequent itemsets such that all of their subsets are frequent. Specifically, the definition of $\mathcal{NB}$ is given by

$$\mathcal{NB} = \{d | d \in \mathcal{S} \wedge supp(d) < \sigma \wedge supp(d') \geq \sigma \forall d' \subset d\}.$$

$\mathcal{NB}$ gives a clear boundary (e.g. Figure 1) where we stop scanning the itemsets with the least loss of information. Based on the definition of $\mathcal{NB}$, the categorical outlier score in Equation (1) is modified as follows:

$$Score_1(\mathbf{x}_i) = \sum_{d \subseteq \mathbf{x}_i \wedge d \in \mathcal{NB} \wedge |d| \leq MAXLEN} \frac{1}{supp(d) \times |d|}. \tag{2}$$

3.2 Modified categorical score

In the previous subsection, there is an emerging hypothesis that all of the categorical attributes are equally important and no obvious hierarchical relationship exists in the dataset. All of the attributes are treated equally in the itemset mining process, while this approach could be untenable in some datasets with far more complex structures. In this subsection, we consider a dataset in which most of the categorical attributes are mutually related or even hierarchically linked to one another. Several categorical attributes of the illustrative dataset in Section 4.1 are ranked from top to bottom according to their hierarchical levels in Figure 2. For example, 'Area'

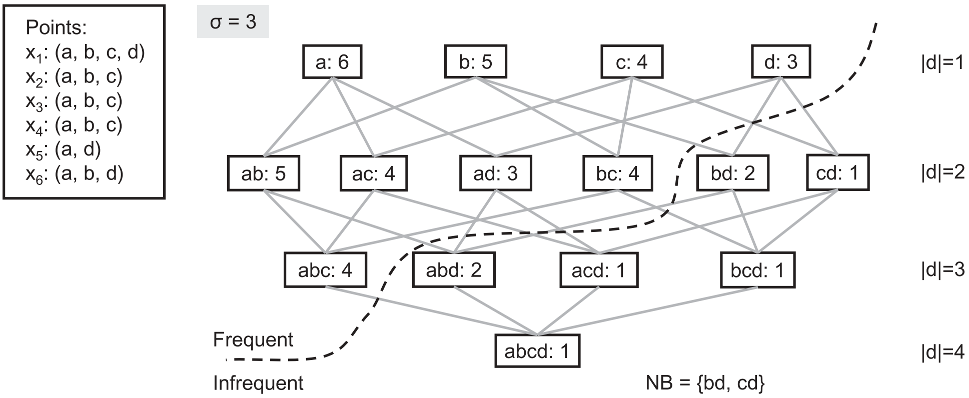

Figure 1. An example dataset with six points in total and its itemset frequency counts.

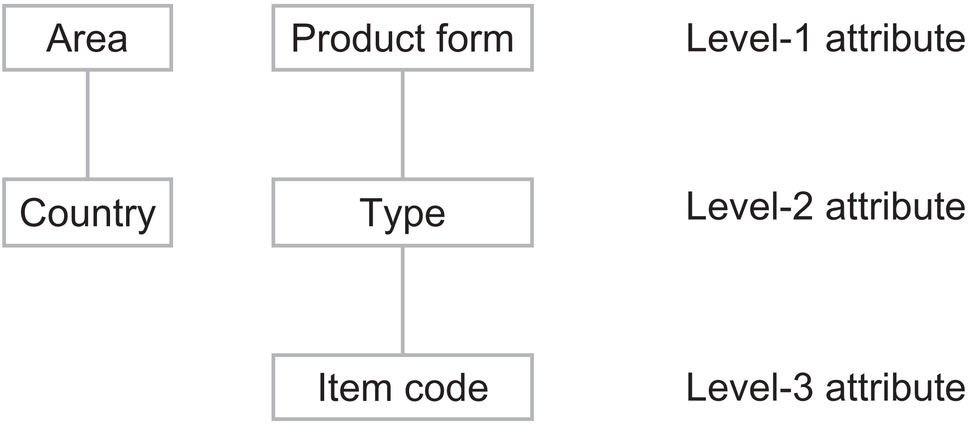

Figure 2. Hierarchical relationship of categorical attributes.

and ‘Country’ are two attributes that have an obvious hierarchical correlation, and they share some information in common. Generally, the attributes with larger level numbers tend to tell more detailed information. By giving the hierarchical structure and ranking the level numbers of these attributes, we can dig deeper into the relationships among the attributes and offer different weights to them.

In Section 3.1, the length of itemset d was defined as the number of categorical attributes in d, while it is under the hypothesis that all of the attributes are at the same level and are treated equally. In our hierarchical model, the length of itemset d should also be related to the levels of the attributes that are included in d. By accounting for the level numbers of the attributes, the length of itemset d is defined as the summation of all of the level numbers of the attributes in d.

The relationship between the itemsets also reflects a hierarchical structure. We define $\mathcal{L}(d)$, or the lower-level itemsets of d, as the set of itemsets in which some of the categorical attributes are descendant nodes of the attributes in itemset d. For example, if $d = \{$Area = ‘Asia’, Type = ‘SAC’$\}$, then $\{$Country = ‘Thailand’, Type = ‘SAC’$\}$ belongs to $\mathcal{L}(d)$ because ‘Thailand’ is a descendant node of ‘Asia’ in the treemap. We can similarly define $\mathcal{U}(d)$, or the upper-level itemsets of d, as a set of the itemsets in which some of the categorical attributes are ancestor nodes of the attributes in itemset d. Based on the definitions of $\mathcal{L}(d)$ and $\mathcal{U}(d)$, we derive a downward close property about $supp(d)$ as shown in Lemma 2, which is similar to the *Apriori* property presented in Lemma 1.

Lemma 2. *If* $supp(d) < \sigma$, *then* $supp(d') < \sigma \; \forall d' \in \mathcal{L}(d)$.

Under this property, the number of itemsets to check can be further decreased as follows: if we find that itemset d in point $\mathbf{x}_i$ is infrequent, then we will not consider any lower-level itemsets of d. By combining the two properties in Lemmas 1 and 2, we define an *Advanced Negative Border* ($\mathcal{ANB}$) as the set of those infrequent itemsets such that all of their subsets and corresponding upper-level itemsets are frequent.

$$\mathcal{ANB} = \{d | d \in \mathcal{S} \wedge supp(d) < \sigma \wedge supp(d') \geq \sigma \forall d' \subseteq d \wedge supp(d'') \geq \sigma \forall d'' \in \mathcal{U}(d)\}.$$

Obtaining $\mathcal{ANB}$ of a dataset is indeed a process of searching itemsets and counting their frequencies in a tree structure (e.g. Example 1). $\mathcal{ANB}$ explains which itemsets are critical to a data point and to which degree those itemsets have an impact. Based on the definition of $\mathcal{ANB}$, we further modify the definition of the type 1 outlier score value for the point $\mathbf{x}_i$ as follows:

$$Score_1(\mathbf{x}_i) = \sum_{d \subseteq \mathbf{x}_i \wedge d \in \mathcal{ANB} \wedge |d| \leq MAXLEN} \frac{1}{supp(d) \times |d|}. \tag{3}$$

Compared to the score formulation of Equation (2), we give more pruning of itemset mining space by digging into the latent hierarchical information among categorical attributes. We will show in the experiment section that this pruning will not reduce the detection accuracy but will lead to higher time efficiency.

Example 1. In the dataset shown in Figure 3, with $\sigma = 2$ and $MAXLEN = 4$, we obtain its *Advanced Negative Border* consisting of four itemsets: $a2$, ac, $b2$, and $b1c$. Thus, the type 1 scores of the points in this dataset are calculated as

$$\begin{aligned}
Score_1(\mathbf{x}_1) &= 0,\ Score_1(\mathbf{x}_3) = 0,\\
Score_1(\mathbf{x}_2) &= \tfrac{1}{supp(a2) \times |a2|} = \tfrac{1}{1 \times 2} = 0.5,\\
Score_1(\mathbf{x}_4) &= \tfrac{1}{supp(b2) \times |b2|} = \tfrac{1}{1 \times 2} = 0.5,\\
Score_1(\mathbf{x}_5) &= \tfrac{1}{supp(ac) \times |ac|} = \tfrac{1}{1 \times 2} = 0.5,\\
Score_1(\mathbf{x}_6) &= \tfrac{1}{supp(b1c) \times |b1c|} = \tfrac{1}{1 \times 3} = 0.33.
\end{aligned}$$

3.3 Continuous score

We have presented how to estimate the anomaly score of a data point in its categorical attribute space. In this subsection, we extend the scope to the whole mixed attribute space, where independencies between different types of attributes are accounted for.

To detect anomalies in continuous attribute space, Otey et al. (2006) applied a covariance score for the continuous attributes under each itemset, and a point's scores under all of the itemsets synthetically reflected its overall deviation from the 'normal' in continuous attribute space. However, to compute the continuous score of a data point, Otey's approach must examine almost every itemset and its corresponding covariance score for the point, which is quite time consuming. Additionally, all of the itemsets should have different weights in affecting the continuous attributes, and thus, we should attempt to avoid simply averaging or adding all of the covariance

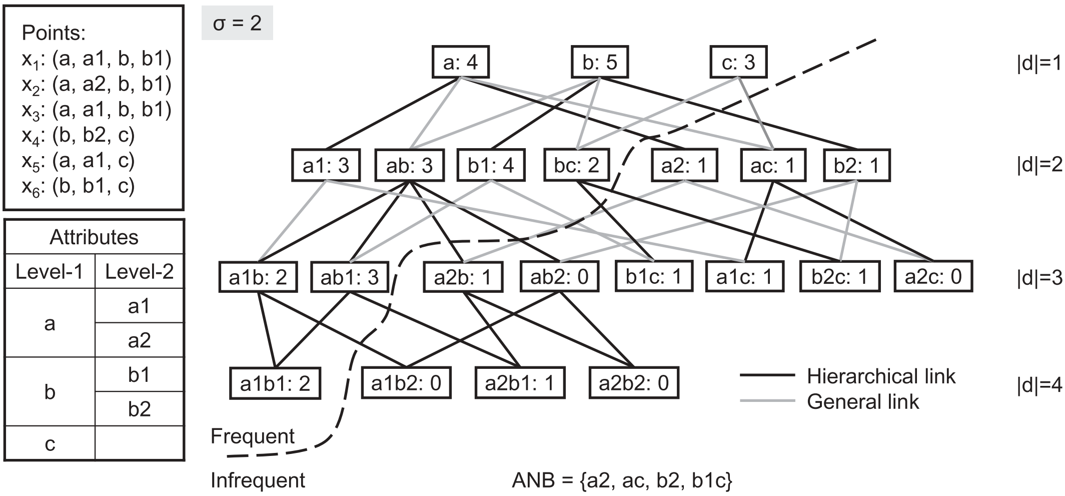

Figure 3. A hierarchically structured dataset with six points in total and its itemset frequency counts.

scores under different itemsets together. In this section, we define a continuous score (or type 2 outlier score) that measures to which degree a data point is deviating from the 'normal' in its continuous attributes. We use the Euclidean distance as a measure of the 'closeness' between the points, and we consider the importance level of different itemsets. Specifically, one way to make a distinction in the importance levels of the itemsets is that we only apply the deviation scores to the most important itemset.

In mixed attribute space, consider a data point $\mathbf{x}_i$ that contains both categorical values and numerical values. The categorical part of $\mathbf{x}_i$ is denoted by $\mathbf{x}_i^C$, and the numerical part of $\mathbf{x}_i$ by $\mathbf{x}_i^N$. Let a represent one of the numerical attributes, and let d be an itemset that can affect the values of attribute a to the greatest extent. We define V_i^a as the deviation of $\mathbf{x}_i$ on attribute a.

$$V_i^a = \begin{cases} \frac{|a_i - \mu_a^d|}{Range_a^d} & \text{if } Range_a^d \neq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where μ_a^d and $Range_a^d$ represent the mean value and range of attribute a under itemset d, respectively, as follows:

$$\mu_a^d = \frac{1}{supp(d)} \times \sum_{i|d\subseteq \mathbf{x}_i^C} a_i, \tag{5}$$

$$Range_a^d = \max_{i|d\subseteq \mathbf{x}_i^C} a_i - \min_{i|d\subseteq \mathbf{x}_i^C} a_i. \tag{6}$$

Thus, the continuous score value for the data point $\mathbf{x}_i$ is defined as

$$Score_2(\mathbf{x}_i) = \sum_{a\in \mathbf{x}_i^N} (1|V_i^a > \delta_a), \tag{7}$$

where δ_a is a threshold for the deviation of $\mathbf{x}_i$ on attribute a. In our definition, a data point with a higher type 1 or type 2 outlier score is more likely to be an anomaly in the dataset. By combining the categorical and continuous scores in mixed attribute space, our method can naturally detect those points that go against the major dependencies between the different attributes.

Before computing V_i^a, we must find the exact itemset d that can affect the values of attribute a to the greatest extent. The process of finding d is essentially a process of feature selection. General methods of feature selection are divided into two categories, filter and wrapper (Kohavi & John, 1997). Filter methods select a subset of the features or attributes that exhibits a top ranking under a specific measurement, such as relevancy, while wrapper methods use the actual target learning algorithm to estimate the accuracy of the feature subsets and they depend on the algorithm used. The existing filter methods, such as CFS (Hall, 2000) and RELIEF (Kononenko, 1994), can be used directly to find the most influencing itemset. On the other hand, we can select the itemset that exhibits the highest accuracy in our algorithm according to the wrapper's concepts.

3.4 Distributed model implementation

We implement the outlier detection algorithm in a distributed fashion using the MapReduce programming model and the Hadoop infrastructure. MapReduce (Dean & Ghemawat, 2008) is a programming model and an associated implementation for processing and generating large datasets. Users design a MapReduce program through two functions: map and reduce. As shown in Figure 4, the users specify a map function that processes a key-value pair to generate a set of intermediate key-value pairs, and a reduce function that merges all of the intermediate values that are associated with the same intermediate key. Hadoop (Abouzeid, Bajda-Pawlikowski, Abadi, Silberschatz, & Rasin, 2009) is an open source distributed infrastructure for the MapReduce implementation. It consists of

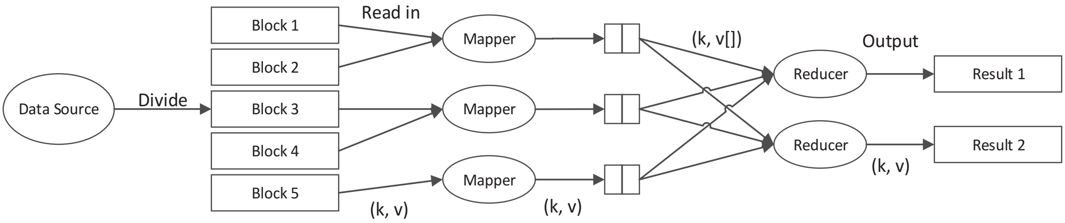

Figure 4. Basic process in MapReduce.

two layers: a data storage layer called the Hadoop Distributed File System (HDFS) and a data processing layer (or MapReduce Framework). In the hybrid system of Hadoop, the advanced properties of MapReduce can be combined with the performance of parallel database systems.

The centralized algorithm is conducted by several MapReduce jobs, which conduct the categorical and continuous outlier score computations separately. In each mapper site, we build an independent local model within a subset of the dataset. Then, the output of the mapper sites with the same key will be combined into a reducer site, where the intermediate results are merged and prepared for the global model.

It takes two passes of a local dataset to calculate the final outlier score for each data point. The first pass is to prepare for the global model construction, during which all of the necessary model parameters, including the itemset frequencies ($supp(d)$), mean value and range of the continuous attribute (μ_a^d, $Range_a^d$), are computed in a MapReduce job. The mapper works on each local site, where it transforms a data point into key-value pairs with itemset d as the 'key' and the numerical part of the point as the 'value'. The key-value pairs with the same key are merged into the same reducer, where it computes the parameters that are required for the next step; for example, to obtain the frequency or support of an itemset d, the reducer counts the number of key-value pairs under d. After a single pass in this stage, all of the parameters in the global model are estimated. The pseudocodes of Mapper and Reducer for this stage are shown in Algorithms 1 and 2, respectively.

Both categorical and continuous outlier scores are calculated in the second pass of data. The centralized algorithm in Section 3.2 and Section 3.3 has shown that the outlier score for a point is calculated based on the number of infrequent itemsets that it contains and the deviation of its continuous values. In this stage, only mappers on local sites are required, because once the global parameters are given, every point's score can be computed without interaction with the other points; thus, the reducer of the MapReduce job in this stage is omitted. Based on the parameters estimated in the first stage, we calculate the outlier score for each data point as shown in Algorithm 3.

A data point is classified as an outlier if its type 1 score exceeds the threshold value c_1 or its type 2 score exceeds the threshold value c_2. The selection of threshold values would influence the detection results. For example, if a threshold value is high, only a few points will be classified as outliers. Hence, both false positive rate (FPR) and true positive rate (TPR) will increase with a decreasing threshold value. In practice, the threshold values are selected to balance the FPR and TPR and achieve effective detection.

Algorithm 1 The mapper for the global model construction

```
Input: dataset $\mathcal{D}$, $MAXLEN$
Output: key-value pair (itemset $d$, $\mathbf{x}_i^N$)
for each $\mathbf{x}_i$, $i = 1, \ldots, n$ do
    for each itemset $d \subseteq \mathbf{x}_i$ and $|d| \leq MAXLEN$ do
        Context.write($d$, $\mathbf{x}_i^N$);
    end for
end for
```

Algorithm 2 The reducer for the global model construction

Input: key-values pair (itemset d, $\{\mathbf{x}_i^N, \mathbf{x}_j^N, \ldots\}$)

Output: $supp(d)$, μ_a^d, $Range_a^d$

```
for each itemset $d$ do
    $supp(d) = 0$; $sum_a^d = 0$; $max_a^d = -\infty$; $min_a^d = \infty$;
    for each $\mathbf{x}_i \supseteq d$ do
        $supp(d) + +$;
        for each attribute $a \in \mathbf{x}_i^N$ do
            $sum_a^d + = a_i$;
            if $a_i > max_a^d$ then
                $max_a^d = a_i$;
            end if
            if $a_i < min_a^d$ then
                $min_a^d = a_i$;
            end if
        end for
    end for
    for each continuous attribute $a$ do
        $\mu_a^d = sum_a^d / supp(d)$;
        $Range_a^d = max_a^d - min_a^d$;
    end for
    Update $supp(d)$, $\mu_a^d$, $Range_a^d$ for $d$ and $a$ in the hash table;
end for
```

Algorithm 3 Outlier score computation

Input: dataset $\mathcal{D}$, $MAXLEN$, σ, δ, $supp(d)$, μ_a^d, $Range_a^d$

Output: $Score_1(\mathbf{x}_i)$, $Score_2(\mathbf{x}_i)$

```
for each $\mathbf{x}_i \in \mathcal{D}$ do
    $Score_1(\mathbf{x}_i) = 0$; $Score_2(\mathbf{x}_i) = 0$;
    $\mathcal{ANB}_1 \leftarrow$ the set of the itemsets with length = 1 in $\mathbf{x}_i$;
    $j = 1$;
    while $j \le MAXLEN$ do
        $\mathcal{ANB}_{j+1} \leftarrow$ the set of the itemsets with length $= j + 1$ that are the supersets or lower-level itemsets of the sets in $\mathcal{ANB}_j$;
        for each itemset $d \in \mathcal{ANB}_j$ do
            Get $supp(d)$ from hash table;
            if $supp(d) < \sigma$ then
                $Score_1(\mathbf{x}_i) + = \frac{1}{supp(d) \times |d|}$;
                Delete all of the supersets and lower-level sets of $d$ from $\mathcal{ANB}_{j+1}$;
            end if
        end for
        $j + +$;
    end while
    for each attribute $a \in \mathbf{x}_i^N$ do
        $d \leftarrow$ the itemset that affects $a$ to the greatest extent;
        Get $\mu_a^d$ and $Range_a^d$ from hash table;
        if $Range_a^d \neq 0$ then
            $V_i^a = \frac{|a_i - \mu_a^d|}{Range_a^d}$;
        else
            $V_i^a = 0$;
        end if
        if $V_i^a > \delta_a$ then
            $Score_2(\mathbf{x}_i) ++$;
        end if
    end for
    Context.write($\mathbf{x}_i$, $Score_1(\mathbf{x}_i)$);
    Context.write($\mathbf{x}_i$, $Score_2(\mathbf{x}_i)$);
  end for
```

4 A real-world case study

4.1 Datasets

This research is greatly driven by a real problem in the information system of a real company. The company needs to manually input new orders into its information system, by recording detailed information about the product (‘Product form’, ‘Type’, ‘Item code’, ‘Usage’), about the logistics plan (‘Order quantity’, ‘Shipping date’), and about the customer (‘Customer ID’, ‘Area’, ‘Country’). In this process, some of the key information about the order could be mistaken by typing errors, and this record becomes an ‘Outlier’; for example, a staff member could mistake product A’s item code for that of B, which is quite likely because the item codes usually look similar.

The original dataset in the information system contains more than 70,000 records of orders in recent years. We further process and integrate the dataset; thus, the final dataset contains 44,341 records of orders. For the purpose of performance comparison, we simulate certain mistakes such as randomly changing a product item code to its similar code, so that we generate 150 records as ‘Outliers’ (approximately 0.34% of the dataset). The final dataset has several characteristics as follows:

- It includes seven categorical attributes and two numerical attributes in total, which indicates that the numbers of the two types of attributes are unbalanced.
- Most of the categorical attributes are hierarchically linked to one another. For example, to represent the product involved in an order, three hierarchically dependent attributes are used from top to bottom, such as ‘Product form’ → ‘Type’ → ‘Item code’. The lower-level attribute shows more detailed information about the product.
- The number of categorical attribute levels is enormous, which leads to a sparse value realization of these attributes. For example, the total number of ‘Item code’ levels can reach up to 3467, which is quite large compared to the scale of the dataset.
- There are three types of outlier records, which are derived from the three most easily mistaken attributes: ‘Item code’ (the first type), ‘Order quantity’ (the second type), and ‘Shipping date’ (the third type). We manually generate 50 records for each type in the dataset; thus, the total number of outlier records is 150.

4.2 Evaluation

To obtain a comparison about properties and performances among different methods, we implemented the proposed method and two other existing state-of-the-art methods, ODMAD (Koufakou & Georgiopoulos, 2010) and Otey’s approach (Otey et al., 2006), on the illustrative dataset above. We evaluate the performance of different outlier detection methods mainly based on two criteria:

- *Outlier Detection rate*, which is the fraction of outliers that are correctly detected by each method.
- *Alarm rate*, which is the fraction of data points that are identified as outliers. In this dataset, the *Alarm rate* is close to the *False Positive rate* because the number of outliers is quite small.

We also compare the running time of the three algorithms on the real-world dataset.

The three methods are all implemented on a Hadoop pseudo-distributed configuration, where the host node has dual 2.5 GHz Intel Core i7 processors and 4 GB of memory, running CentOS 7.

4.3 Results

For each algorithm, we experimented with various parameters and threshold settings, selecting their best parameter combination for model implementation and basing the comparison on their respective best performance. Considering that the two measures, the outlier detection rate and the alarm rate, are mutually restricted, we focus only on the corresponding detection rate when the alarm rate is approximately 10%, which is an acceptable value for the alarm rate in practice. In Table 2, we present the detection results for the three types of outliers using three algorithms.

The experimental results show that our method can detect the three typical mistake patterns in the dataset both effectively and efficiently. Overall, our method achieves an average detection rate of 82.60% with a 10.17% alarm rate, while Oteys has an average detection rate of 74.00% with 9.80% alarm rate, and ODMAD has an average detection rate of 52.60% with 10.45% alarm rate. For the first and third mistake pattern, our method has equal or better detection accuracy than the other two methods. For the second mistake pattern, even though our method does not achieve the best performance, it is still quite close to the best performance. Its sensitivity to outliers in the mixed attribute space results from our utilization of the hierarchical property among categorical attributes, which allows us to allocate different weights to the itemsets in computing the outlier scores according to their positions in the hierarchical structure.

In terms of the time efficiency, our method is apparently better than the other two methods. According to the results shown in Table 2, our method takes only 10.3 min to detect the outliers in the entire dataset with 44,341 records, while the other two methods require more than 30 min to accomplish the same work. The reason behind our efficiency is that we propose the definition of $\mathcal{ANB}$ for the pruning of itemset mining space based on the hierarchical structure embedded in categorical attributes. Specifically, our proposed method can free us from scanning approximately 40% of the infrequent itemsets in the experimental dataset. This time saving would benefit more from a dataset with deeper hierarchical structure involving more attribute levels.

Table 2. Comparison of ODMAD, Otey's, and our method on the real-world dataset.

		Otey's	ODMAD	Our method
First type	Detection amount	43	41	43
(Item code)	Detection rate	86.00%	82.00%	86.00%
Second type	Detection amount	46	38	44
(Order quantity)	Detection rate	92.00%	76.00%	88.00%
Third type	Detection amount	22	0	37
(Shipping date)	Detection rate	44.00%	0	74.00%
Total detection amount		111	79	124
Average detection rate		74.00%	52.60%	82.60%
Total alarm rate		9.80%	10.45%	10.17%
Running time (min)		32.30	30.08	10.30

5 Simulation study

In this section, the proposed outlier detection method is compared with the other two alternatives, ODMAD (Koufakou & Georgiopoulos, 2010) and Otey's approach (Otey et al., 2006), in terms of detecting the outliers in a simulated dataset that deviate in both categorical and continuous spaces. We consider a simulated dataset including four categorical variables (X_1, X_2, X_3, X_4) and two continuous variables (Y_1, Y_2), in which the first three categorical variables (i.e. X_1, X_2, and X_3) are linked hierarchically. For the categorical part, X_1 has 50 attribute levels. X_2 has 20 attribute levels under each value of X_1; thus, it has a total number of $50 \times 20 = 1000$ attribute levels. Similarly, X_3 has three attribute levels under each value of X_2, leading to a total number of $50 \times 20 \times 3 = 3000$ attribute levels. X_4 is a categorical variable with 250 attribute levels.

We generate two types of outlier points by simulating random deviation in categorical space and continuous space, respectively. In the categorical space, X_4 is selected as the shifting variable, and we represent this as categorical shifts. Similarly, we simulate continuous shifts by allocating a random deviation to the continuous variable Y_2.

The experimental dataset consists of 100,000 normal points as well as 100 categorical shift points and 100 continuous shift points. By plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings, the ROC curves of different methods for detecting categorical shifts (Figure 5), continuous shifts (Figure 6), and both of shifts (Figure 7) give us a glimpse of the detection performance. According to the comparison on these ROC curves, our method outperforms other two methods in detecting both categorical and continuous shifts.

As the same with that in the real-world case study, an integrated comparison is presented in Table 3, which mainly includes the detection rates of different methods when the total alarm rate is approximately 10% and their running time. Based on the results in Table 3, when the alarm rates under each method are controlled at a level of approximately 10%, our method achieves the overall best performance with the highest average detection rate and the shortest running time.

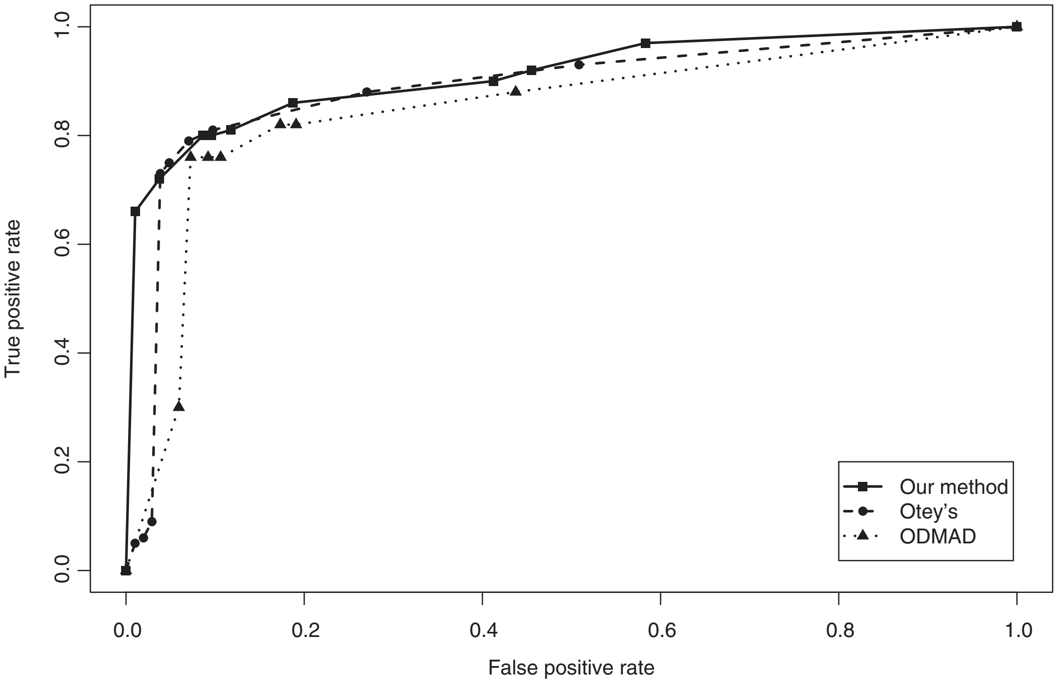

Figure 5. ROC curves for detecting categorical shifts.

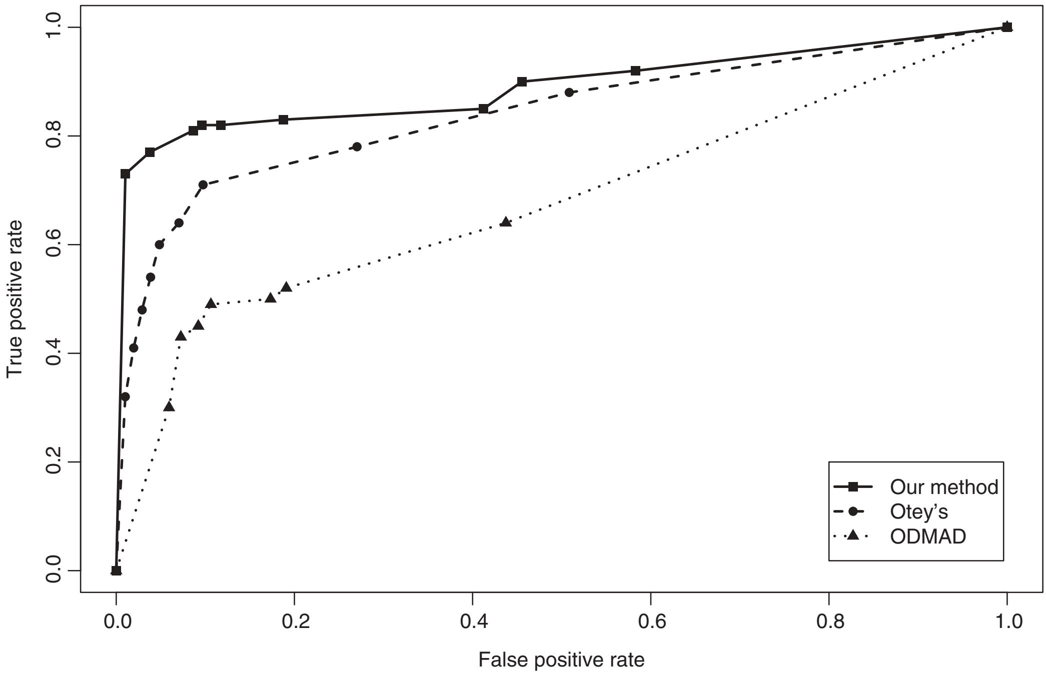

Figure 6. ROC curves for detecting continuous shifts.

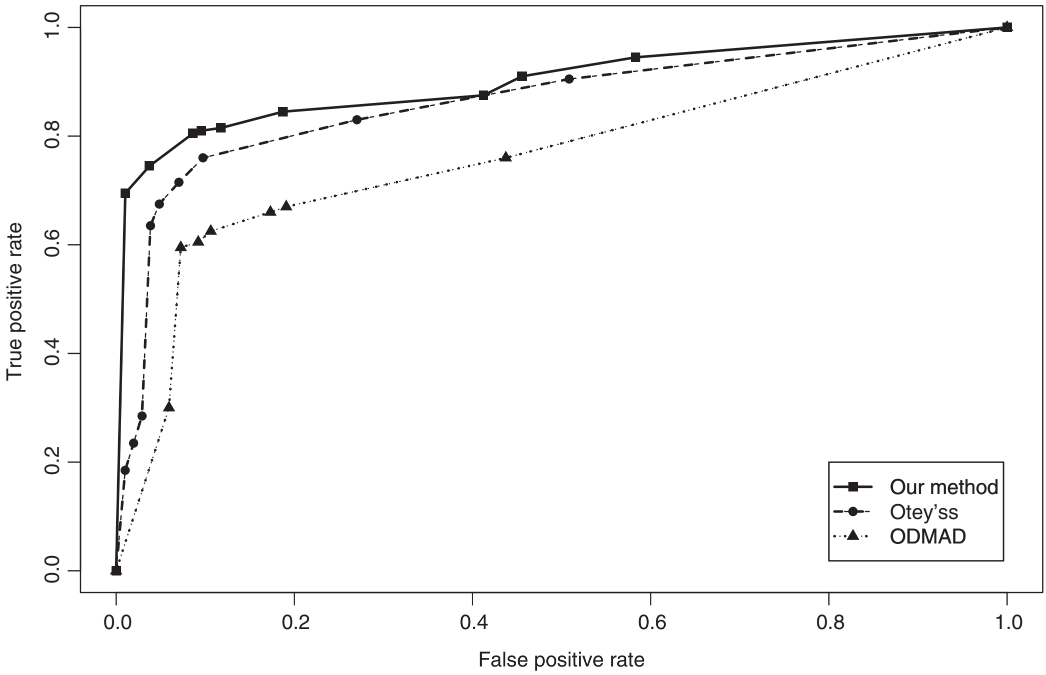

Figure 7. ROC curves for detecting both categorical and continuous shifts.

The results of the simulation study are consistent with that in the real-world case study. Our proposed method is able to capture the hierarchical structure of categorical attributes and the dependencies between mixed attributes, which leads to effective detection of random deviation in

Table 3. Comparison of ODMAD, Otey's, and our method on the simulated dataset.

	Otey's	ODMAD	Our method
Detection rate for categorical shifts	81.00%	76.00%	80.00%
Detection rate for continuous shifts	71.00%	49.00%	82.00%
Average detection rate	76.00%	62.50%	81.00%
Total alarm rate	9.75%	10.62%	9.58%
Running time (s)	49.13	34.67	27.79

both categorical and continuous spaces. In addition, the scanning of itemsets in the categorical space is more efficient in our method due to the pruning of itemset mining space based on $\mathcal{ANB}$.

6 Conclusions

In this study, we have presented a fast distributed outlier detection algorithm for hierarchically structured mixed datasets. The proposed method first identifies outliers based on the categorical attributes as well as their frequent patterns. Next, it concentrates on the subsets of data in the mixed space by utilizing the dependencies between the two types of attributes. We further modify the algorithm based on the downward closure property of the itemsets under hierarchical structures to improve its detection rates and time efficiency. To demonstrate the performance of the proposed method, we experimented with both a real-world-mixed dataset from the order system and a simulated mixed dataset. According to the experimental results, our method outperforms the other two state-of-the-art outlier detection methods, ODMAD and Otey's approach, in terms of the detection accuracy and time efficiency under the same circumstances.

We have implemented our method in a pseudo-distributed fashion using the MapReduce framework, where for now the pseudo-distributed mode has only one computing node. Future work includes increasing the number of computing nodes and evaluating its speedup performance under a real distributed circumstance.

Acknowledgments

The authors would like to thank the editor and referees for their valuable comments, which have helped us improve this work greatly. This research is supported by the National Key R&D Program of China under grant No. 2017YFF0209400 and No. 2016YFF0204100.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Key R&D Program of China [2017YFF0209400];National Key R&D Program of China [2016YFF0204100].

Notes on contributors

Qiao Liang is a Ph.D. student in the Department of Industrial Engineering, Tsinghua University, Beijing, China. She received her B.S. degree in Industrial Engineering from Tsinghua University in 2016. Her research interests include statistical modeling and data analytics for manufacturing and service processes, with a focus on statistical process control based on text analytics.

Kaibo Wang is a Professor in the Department of Industrial Engineering, Tsinghua University, Beijing, China. He received his B.S. and M.S. degrees in Mechatronics from Xi'an Jiaotong University, Xi'an, China, and his Ph.D. in Industrial Engineering and Engineering Management from the Hong Kong University of Science and Technology,

Hong Kong. His research focuses on statistical quality control and data-driven system modeling, monitoring, diagnosis, and control, with a special emphasis on the integration of engineering knowledge and statistical theories for solving problems from the real industry.

ORCID

Kaibo Wang http://orcid.org/0000-0001-9888-4323

References

Abouzeid, A., Bajda-Pawlikowski, K., Abadi, D., Silberschatz, A., & Rasin, A. (2009). Hadoopdb: An architectural hybrid of mapreduce and dbms technologies for analytical workloads. *Proceedings of the Vldb Endowment*, *2*, 922–933.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th international conference on very large data bases* (pp. 487–499). San Francisco, CA: Morgan Kaufmann Publishers Inc.

Barnett, V., & Lewis, T. (1994). *Outliers in statistical data*. Chichester: John Wiley & Sons.

Bay, S. D., & Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Acm sigkdd international conference on knowledge discovery and data mining* (pp. 29–38). New York, NY: ACM.

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, *17*, 235–249.

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). Lof: Identifying density-based local outliers. *Acm Sigmod Record*, *29*, 93–104.

Chen, Y., Miao, D., & Zhang, H. (2010). Neighborhood outlier detection. *Expert Systems with Applications*, *37*, 8745–8749.

Das, K., & Schneider, J. (2007). Detecting anomalous records in categorical datasets. In *Acm sigkdd international conference on knowledge discovery and data mining* (pp. 220–229). New York, NY: ACM.

Dean, J., & Ghemawat, S. (2008). Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, *51*, 107–113.

Fan, H., Zaiane, O. R., Foss, A., & Wu, J. (2006). A nonparametric outlier detection for effectively discovering top-n outliers from engineering data. In *Pacific-asia conference on advances in knowledge discovery and data mining* (pp. 557–566). Berlin, Heidelberg: Springer.

Ghoting, A., Otey, M. E., & Parthasarathy, S. (2004). Loaded: Link-based outlier and anomaly detection in evolving data sets. In *Ieee international conference on data mining* (pp. 387–390). Washington, DC: IEEE.

Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In *Seventeenth international conference on machine learning* (pp. 359–366). San Francisco, CA: Morgan Kaufmann Publishers Inc..

Han, J., Cheng, H., Xin, D., & Yan, X. (2007). Frequent pattern mining: Current status and future directions. *Data Mining and Knowledge Discovery*, *15*, 55–86.

He, Z., Xu, X., Huang, J. Z., & Deng, S. (2005). Fp-outlier: Frequent pattern based outlier detection. *Computer Science and Information Systems*, *2*, 103–118.

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, *22*, 85–126.

Knorr, E. M., & Ng, R. T. (1998). *Algorithms for mining distance-based outliers in large datasets* (pp. 392–403).

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*, 273–324.

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of relief. In *European conference on machine learning* (pp. 171–182). Berlin, Heidelberg: Springer.

Koufakou, A., & Georgiopoulos, M. (2010). A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes. *Data Mining and Knowledge Discovery*, *20*, 259–289.

Koufakou, A., Ortiz, E. G., Georgiopoulos, M., Anagnostopoulos, G. C., & Reynolds, K. M. (2007). A scalable and efficient outlier detection strategy for categorical data. In *Ieee international conference on tools with artificial intelligence* (pp. 210–217). Washington, DC: IEEE.

Koufakou, A., Secretan, J., Reeder, J., & Cardona, K. (2008). Fast parallel outlier detection for categorical datasets using mapreduce. In *Ieee international joint conference on neural networks* (pp. 3298–3304). Washington, DC: IEEE.

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., & Srivastava, J. (2003). A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the 2003 siam international conference on data mining* (pp. 25–36). Society for Industrial and Applied Mathematics.

Lee, W., & Xiang, D. (2001). Information-theoretic measures for anomaly detection. *IEEE Symposium on Security and Privacy* (pp. 130–143). Washington, DC: IEEE.

Li, S., Lee, R., & Lang, S. D. (2007). Mining distance-based outliers from categorical data. In *Ieee international conference on data mining workshops* (pp. 225–230). Washington, DC: IEEE.

Narita, K., & Kitagawa, H. (2008). Detecting outliers in categorical record databases based on attribute associations. In *Progress in www research and development, asia-pacific web conference* (pp. 111–123). Berlin, Heidelberg: Springer.

Nivetha, G., & Venkatalakshmi, K. (2018). Hybrid outlier detection (hod) method in sensor data for human activity classification. *Intelligent Data Analysis*, *22*, 245–260.

Otey, M. E., Ghoting, A., & Parthasarathy, S. (2006). Fast distributed outlier detection in mixed-attribute data sets. *Data Mining and Knowledge Discovery*, *12*, 203–228.

Pai, H. T., Wu, F., & Hsueh, P. Y. S. (2014). A relative patterns discovery for enhancing outlier detection in categorical data. *Decision Support Systems*, *67*, 90–99.

Papadimitriou, S., Kitagawa, H., Gibbons, P. B., & Faloutsos, C. (2002). Loci: Fast outlier detection using the local correlation integral. In *Proceedings of the international conference on data engineering* (pp. 315–326). Washington, DC: IEEE.

Penny, K. I., & Jolliffe, I. T. (2010). A comparison of multivariate outlier detection methods for clinical laboratory safety data. *Journal of the Royal Statistical Society*, *50*, 295–307.

Ruts, I., & Rousseeuw, P. J. (1996). Computing depth contours of bivariate point clouds. *Computational Statistics and Data Analysis*, *23*, 153–168.

Wang, X., & Davidson, I. (2009). Discovering contexts and contextual outliers using random walks in graphs. In *Ieee international conference on data mining* (pp. 1034–1039). Washington, DC: IEEE.

Wu, S., & Wang, S. (2013). Information-theoretic outlier detection for large-scale categorical data. *IEEE Transactions on Knowledge & Data Engineering*, *25*, 589–602.

Yu, J. X., Qian, W., Lu, H., & Zhou, A. (2006). Finding centric local outliers in categorical/numerical spaces. *Knowledge and Information Systems*, *9*, 309–338.

Yu, Q., Luo, Y., Chen, C., & Bian, W. (2016). Neighborhood relevant outlier detection approach based on information entropy. *Intelligent Data Analysis*, *20*, 1247–1265.

Zhang, K., & Jin, H. (2010). An effective pattern based outlier detection approach for mixed attribute data. In *Australasian joint conference on artificial intelligence*, 122–131.