Cloud Model-based Outlier Detection Algorithm for Categorical Data

Dajiang Lei[1*], Liping Zhang[2] and Lisheng Zhang[1]

[1]*School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China*
[2]*College of Mobile Telecommunications, Chongqing University of Posts and Telecommunications, Chongqing, China*
**Corresponding Author*
leidj@cqupt.edu.cn, zhanglipingrose@163.com, zhangls@cqupt.edu.cn

Abstract

Most of the existing outlier detection methods aim at numerical data, but there will be a large number of categorical data in real life. Some outlier detection algorithms have been designed for categorical data. There are two main problems of outlier detection for categorical data, which are the similarity measure between categorical data objects and the detection efficiency problem. A cloud model-based outlier detection algorithm for categorical data is proposed in this paper. The algorithm is based on data driven idea and does not require the user to specify parameters. We utilize the synthetic data set and real data set to verify, compare our algorithm with the existing outlier detection algorithms for categorical data, and the experimental result demonstrates that our proposed algorithm has a higher detection rate and lower false alarm rate, while the time complexity is also more competitive.

Keywords: *outlier detection; categorical data; cloud model; certain measure*

1. Introduction

Outlier detection has many applications in fraud detection [1], network intrusion detection [2], insurance fraud [3], medical diagnosis [4]. Efficient outlier detection can help us make good decisions on erroneous data or prevent the negative influence of malicious and faulty behavior. Many data mining techniques try to reduce the influence of outliers or eliminate them entirely. However, the aforementioned manner may result in the loss of important hidden information [5]. Alternatively, outlier detection techniques can lead to the discovery of important information in the data and may construct a new theory or produce a new application domain. The facts are precisely what Knorr had stated, viz., "one person's noise is another person's signal" [6].

With reference to the definition of outliers, one of the most widely accepted definitions of an outlier pattern is provided by Hawkins [7]: "An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism". According to the above definition, many outlier detection algorithms have been proposed. However, most of the research efforts with respect to outlier detection techniques have focused on datasets only consisting of numerical attributes, or ordinal attributes which can be directly mapped into numerical values, also known as continuous values. However, much of the data existed in the databases is categorical, where attribute values can not be naturally mapped as numerical values.

Quite often, it is assumed that the categorical attributes could be easily mapped into numerical attributes. However, mapping categorical attributes to numerical attributes is not a straightforward process. For most outlier detection algorithms, the final results greatly depend on the mapping that is used, *e.g.*, the mapping of a marital status attribute (married or single) or a person's profession (engineer, financial analyst, *etc.*) to a numerical attribute. Under the circumstances, the performance is deteriorated.

In accordance with further analysis corresponding to outlier detection problem, the concept of proximity is significant for most outlier detection algorithm [8]. We usually use the notion of similarity or distance to express proximity. If the measure of proximity corresponding to categorical data instance is imprecise, the final result of outlier detection algorithm is often undependable.

In this paper, we propose a cloud model-based outlier detection algorithm for categorical data, which consists of three stages. In the first stage, unlike the traditional outlier detection algorithm for categorical data, we transform the categorical data to the numerical data by using one probability mapping method. In the second stage, we use the "cloud drop" data in the first stage to generate a multi-dimensional cloud model by back forward cloud generator algorithm. In the final stage, we use forward cloud generator algorithm to compute the certainty degree and make the certainty degree as the outlier degree of data objects. The algorithm is based on data driven and does not require the user to specify parameters. We utilize the synthetic data set and real data set to verify, compare our algorithm with the existing outlier detection algorithms for categorical data, and the experimental result demonstrates that our proposed algorithm has a higher detection rate and lower false alarm rate, while the time complexity is also more competitive.

This paper is organized as follows. Section 2 surveys the related work. Section 3 introduces the principle of cloud model. Section 4 presents a feature extraction method for categorical data. We propose our algorithm in Section 5. The experimental results are shown in Section 6. We conclude the paper in Section 7.

2. Related Work

There have been a few scholars that start paying attention to outlier detection for categorical data or mixed attribute data alone. For the outlier detection algorithm of mixed attribute data, we only intercept part of the algorithm for categorical data. For the outlier detection technology using the same outlier definition pattern, we consider only the most effective algorithm currently.

He *et al.*, used Entropy [9] to express the "uncertainty" and "disorder" of categorical data set in the literatures [10, 11]. The outlier detection algorithm proposed in the literature removes a class of data objects (records) one by one, and calculates the overall "uncertainty" and "disorder" of the data set. If the whole data set has less "uncertainty" or "disorder", the removed data object will be detected as the potential outlier.

He *et al.*, proposed a Local-Search heuristic-based Algorithm (LSA) [10] and forward a Greedy Search Algorithm [10] subsequently, which are used to search for the outlier data set that cause the less entropy of the overall data set. According to the analysis of literature [10], the Greedy Search Algorithm is superior to LSA. The definitions of the outlier degree based on the two methods are consistent in this paper, just the search algorithms are different, so our comparison experiment will only considerate the Greedy algorithm (EBGOD).

He *et al.*, defined a Frequent Pattern Outlier Factor (FPOF) for outlier detection (FPOFOD) of categorical data set in [12]. Frequent Itemset Mining (FIM) comes from the relevant research literatures of Association Rule Mining [13-15]. Giving a user-defined

threshold called minimum support ($minsup$), a frequent itemset is one that appears in the data set at least $minsup$ times. The set of all frequent itemsets given $minsup$ is denoted by FIS. To get the FPOF outlier score, FIS are first calculated, and a set is constituted that meet a certain support degree from the FIS. Then, the support degree of frequent itemsets is calculated, which is contained by the set in each data point. Finally, the ratio of the total support degree in each data point and the number of items in the set is calculated.

Otey *et al.*, also proposed the outlier detection method based on frequent itemsets (InverseNotFPOFOD) in [16]. They assign to each point an anomaly score inversely proportionate to its infrequent itemsets. They also maintain a covariance matrix for each itemset to handle continuous attributes. We omit this part since our focus is on categorical data. This algorithm also first mines the categorical dataset for the frequent itemsets, and then calculates an outlier score for data record based on the above definition. The authors state that the execution time is linear to the number of data, but exponential to the number of categorical attributes.

Koufakou *et al.*, in [17] proposed a new outlier detection algorithm for categorical data, called Attribute Value Frequency (AVF) algorithm (AVFOD). The algorithm generates an outlier score called Attribute Value Frequency (AVF) for each data record. The algorithm assumes that outliers are those points that represent abnormal pattern in the data set, and claims that an “ideal” outlier point in a categorical data set is one whose each and every attribute value is extremely irregular (or infrequent). The algorithm first scan the categorical data set for calculating frequency of each attribute value, and then count the average frequency of all attribute belonging to every data record as the outlier score. The authors claim that their algorithm scans only a single pass over entire data set, companying the low time complexity and the higher outlier detection accuracy.

Despite these algorithms above claim to have lower time complexity and higher detection rate, however, these efforts have not been contrasted to each other using the identical criterion for the same data set. In this paper, we analyze the characteristics of categorical data, and propose a Cloud Model-based Outlier Detection Algorithm for categorical data, and conduct a comprehensive comparison to the above algorithms in detection effect (detection rate and false alarm rate) and detection efficiency (detection time).

3. Principle of Cloud Model

In 1995, Professor Deyi Li, the academician of Chinese Academy of Engineering, pioneered the concept of "cloud" based on probability theory and fuzzy mathematics for the lack of probability theory and fuzzy mathematics in dealing with uncertainty data, and makes a further study for the fuzziness and randomness as well as the correlation between them of the uncertainty data. Professor Deyi Li initiatively proposes a method that uses the “cloud model” to make a unified description for the fuzziness and randomness as well as the correlation between them existing in many prophecy characteristics data. The “cloud model” is treated as a transformation model with ambiguity and uncertainty language description, and the uncertainty is used to measure the relationship between a qualitative concept and its numerical representation. The cloud model can be used to represent the primitives in natural language, viz. linguistic value; the digital characteristics of the cloud, viz. Expectation (Ex), Entropy (En) and Hyper Entropy (He), are used to represent the mathematical properties of the linguistic value. These concepts are a huge breakthrough in uncertain information processing field [18].

3.1. Concept and Digital Characteristics of Cloud Model

Supposing U is a quantitative domain expressed with accurate numerical value, and T is a qualitative concept in U. If the quantitative value $x \in U$, and x is a random realization of the qualitative concept T, the certainty degree of the membership degree of x to T can be expressed as $\mu_T(x) \in [0,1]$, which is a random number with stable tendency, and is expressed as the mathematical formula $\mu : U \to [0,1]$, where $\mu_T(x)$ represents the membership function. The distribution of x in domain U is called Cloud, denoted as $Cloud(X)$, and each x is called a cloud drop [19]. If the domain of the qualitative concept is n-dimensional space, then the concept can be extended to n-dimensional cloud.

The digital characteristics of cloud can be used to express the entire characteristics of cloud model. There are three digital characteristics to describe a qualitative concept of the cloud model: Ex (Expected value), En (Entropy) and He (Hyper Entropy). For the entire characteristics of the multi-dimensional cloud model, it can be represented by the multi-dimensional digital characteristics, as shown in Figure 1. If the cloud model uses the three digital characteristics to represent the entire characteristics of the qualitative concept, it can be written as $Cloud(E_x, E_n, H_e)$.

E_x represents the point that can most represent the qualitative concept in the domain space. In other words, it's the most typical sample of this concept quantization. E_n represents a measurable particle size of the qualitative concept. Generally, the larger entropy is, and the higher abstraction the concept is. Entropy also reflects an uncertainty measurement of the qualitative concept, and jointly determined by the randomness and fuzziness of the qualitative concept. The randomness measurement of the qualitative concept reflects the dispersion degree of the cloud drop that can represent the qualitative concept, and the fuzziness measurement of the qualitative concept reflects the values range of the cloud drop that can be accepted by the concept in the domain space. He is the uncertainty measurement of entropy, namely the entropy of the entropy. It reflects the randomness of the sample that represents the qualitative concept, and reveals the fuzziness and randomness of the entropy.

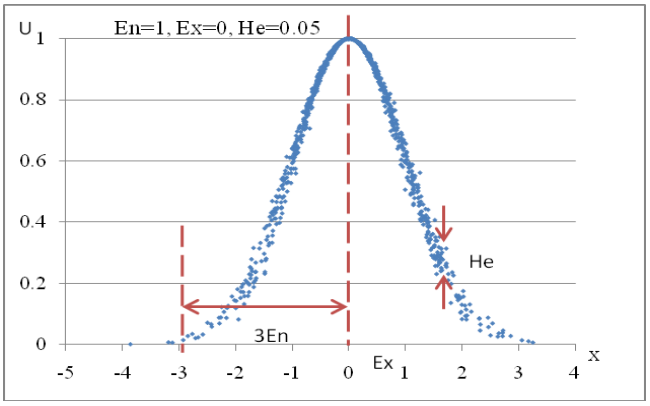

Figure 1. The Cloud Membership and its Digital Characters

3.2. Cloud Generator

Cloud generator is a cloud model generation algorithm by software modules or hardware modules, which can build the interdependent, interrelated and inseparable mapping relationship between the qualitative concept and the quantitative expression. The cloud generator mainly includes forward cloud generator and backward cloud generator, as shown in Figure 2.

In scientific research and actual social production, the various branches of the society and natural sciences have fully demonstrated that normal distribution has a high universality. So the unknown data distribution cloud model is usually assumed to be normal cloud, and the normal cloud becomes the most basic "cloud". The normal cloud will be very useful when it is used to express the basic linguistic value, viz. linguistic atom in the natural language [20]. When using cloud generator to generate a cloud model, we generally assume the normal cloud as the generation target. The cloud generator contains forward cloud generator and back forward cloud generator. The model of cloud generator is shown in Figure 2.

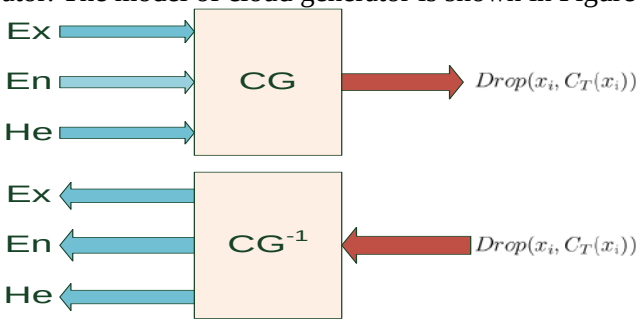

Figure 2. Forward Cloud Generator and Back Forward Cloud Generator

Back forward cloud generator is an uncertainty transform model on transforming between numerical values and linguistic values, which maps from quantitative expressions to qualitative concepts. It efficiently convert the accurate data with a certain number to the qualitative concept represented by appropriate qualitative linguistic value $\{E_x, E_n, H_e\}$, and accordingly represents the entirety of cloud drops reflected by the accurate data. The more accurate data of the cloud drops, the reflection concept is more accurate. If a lot of cloud drops in the cloud model are already known, through the back forward cloud generator, we can restore the three digital Characteristics of the cloud from the given number of cloud drops, which are Expectation E_x, Entropy E_n and Hyper Entropy He. When restoring the digital characteristics of the cloud, it should be noted that the traditional back forward cloud generation algorithm need the value of certainty degree $\mu(x)$ that represents the membership degree of the cloud drops to the cloud model. However, in practical applications, the available data are usually only a set of data values that represents a qualitative concept, while the value of certainty degree $\mu(x)$ that represents the membership degree to qualitative concept is difficult to obtain. Assuming the cloud model is normal cloud, the back forward cloud generation algorithm based on the statistics of the sample only uses the quantitative values of cloud drop x_i to restore the three digital characteristics of the cloud, and does not require the value of certainty degree $\mu(x)$. This algorithm is not only easy to implement, and easy to popularize to high-dimensional cloud, but also the accuracy is higher than the original back forward cloud generation algorithm [21]. When the numerical domain of the cloud drop is one-dimensional space, the back forward cloud generation algorithm is shown as follows.

Algorithm: Back Forward Cloud Generator Algorithm (BFCG)
Input: N cloud drops $x_i(i = 1, \cdots, N)$
Output: Three digital characteristics of cloud model — Expectation E_x, Entropy E_n and Hyper Entropy He
Step 1. According to x_i, calculate mean value of the samples $\overline{X} = \frac{1}{N}\sum_{i=1}^{N} x_i$, absolute central moment of the first order samples $S = \frac{1}{N}\sum_{i=1}^{N}|x_i - \overline{X}|$ and sample variance $S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{X})^2$;

Step 2. Calculate $E_{\hat{x}} = \overline{X}$;
Step 3. Calculate $E_{\hat{n}} = \sqrt{\frac{\pi}{2}} \times \frac{1}{N} \sum_{i=1}^{N} |x_i - E_{\hat{x}}|$;
Step 4. Calculate $H_e = \sqrt{S^2 - E_{\hat{n}}^2}$

Forward cloud generator is an uncertainty transform model on transforming between linguistic values of a basic concept and numerical representation, which maps from qualitative concepts to quantitative expressions. The forward cloud generator shapes cloud drops by analyzing the digital characteristics of the cloud, and the cloud drops will be gathered into a cloud when it reaches a certain number. By analyzing the qualitative concept expressed with linguistic value, the forward cloud generator can get the range and distribution rule of the quantitative data, which process is direct and forward. The inputs of forward cloud generator include the expectation value of the qualitative concept E_x, entropy E_n, hyper entropy He and the number of cloud drops N. The outputs are the quantitative positions of the N cloud drops in the numerical domain space, which also represents the certainty degree of the concept represented by each cloud drop. When the numerical domain of the qualitative concept is one-dimensional, the algorithm of the forward cloud generator is shown as follows.

Algorithm: Forward Cloud Generator Algorithm (FCG)
Input: Three digital characteristics E_x, E_n, He and the number of cloud drops N;
Output: The quantitative values of the N cloud drops, the certainty degree of the concept of each cloud drops
Step 1. Generate a normal random number E_n' with the expectation value E_n and the standard deviation H_e;
Step 2. Generate a normal random number x with the expectation value E_x and the standard deviation absolute value of E_n';
Step 3. Let x be a specific quantitative value of qualitative concept, called a cloud drop;
Step 4. Calculate $y = e^{\frac{-(x-E_x)^2}{2E_n'^2}}$;
Step 5. Let y be the certainty degree of x belongs to the qualitative concept;
Step 6. $\{x, y\}$ reflects the whole contents of this qualitative and quantitative transformation;
Step 7. Repeat Step 1~6 until N cloud drops are generated.

If the corresponding domain of the concept is n-dimensional space, the algorithm can be easily extended, and then we can get the n-dimensional normal cloud.

4. Categorical Data Feature Extraction

In order to facilitate the description of categorical data, the following symbols and definitions are given. Categorical dataset $D = \{x_1, x_2, \cdots, x_i, \cdots, x_N\}$, where N represents the number of data in the data set. The data set contains d-dimensional categorical attributes, and the attribute set is expressed as $A = \{A_1, A_2, \cdots, A_k, \cdots, A_d\}$, where A_k represents the kth column attributes. There are n_k values in the A_k attribute column of the data set, and the attribute values constitute the set of $\mathcal{A}_k$, $x_i = \{x_{i1}, x_{i2}, \cdots, x_{ik}, \cdots, x_{id}\}$, where x_{ik} represents the value of x_i in the A_k attribute column.

$f_k(x_{ik})$: It represents the number of records with the value x_{ik} in the A_k attribute column of the data set D, if $x_{ik} \ni \mathcal{A}_k$, then $f_k(x_{ik}) = 0$.

$\hat{p}_k(x_{ik})$: It represents the probability with the value x_{ik} in the A_k attribute column of the dataset D, and it can be calculated as $\hat{p}_k(x_{ik}) = f_k(x)/N$.

For the non-specific special handling method of categorical data, the categorical attributes are randomly mapped as numeric attributes. However, for the meanings of most categorical

attributes, it can't be carried out directly, and may result in the loss or distortion of information. For most outlier detection algorithms, the final detection result largely depends on the adopted mapping method. For example, if the categorical attributes of marital status (married or single) or career information (engineer, accountant, *etc.*) are mapped to the numeric attributes, it will lead to information errors, and the detection effect will be a serious decline.

By the more in-depth analysis to outlier detection, literature [8] proposes that most outlier detection algorithms are based on an proximity concept, which has a great significance to the correctness of outlier detection algorithms. We usually use the concept of similarity or distance to represent such a proximity concept, if the adjacent measure method for categorical data objects is inaccurate, then the final result of the outlier detection should also be unreliable.

Based on the above analysis, the adjacent measure for categorical data can't use the direct mapping method. The deeper reason is the categorical data has its own characteristics that the different values of categorical data are not inherent orderly, so it can't map from categorical data to numerical data. In order to calculate the similarity degree of categorical data, the data records of categorical attributes data set and consecutive numerical data set should be treated equally in the non-specific categorical data outlier detection method [22]. Using the method to calculate similarity degree directly, and the similarity function is called matching similarity [23], and is shown as follows.

$$Sim\_Overlap = \frac{1}{d}\sum_{k=1}^{d} S_k(x_{ik}, x_{jk}) \qquad (1)$$

$$S_k(x_{ik}, x_{jk}) = \begin{cases} 1 & x_{ik} = x_{jk} \\ 0 & x_{ik} \neq x_{jk} \end{cases} \qquad (2)$$

where $i \neq j$, and $S_k(x_{ik}, x_{jk})$ represents whether the two attribute values are matching. An obvious drawback of matching similarity is that it does not distinguish the values of all attribute columns, and all matching items and non-matching items are treated equally. Because the distribution frequency of some attribute values in some attribute columns are high, while some attribute values of some attribute columns are very rare, so their contribution to the similarity need to distinguish [24].

The outlier detection technology for categorical data described in this paper does directly calculate the similarity between the categorical data. Through further analysis, we can see these methods use the own features of categorical data to detect the outliers. Some of the feature extractions use the overall characteristics of the categorical data set, as the entropy-based greedy search outlier detection method, where the entire entropies of the data set are an overall characteristic. Frequent itemset-based, inverse infrequent itemset-based and attribute value frequency-based outlier detection method simultaneously use the overall characteristics and the partial characteristics of the data set, which calculate the respective characteristic values by their own characteristics with respect to the overall characteristics.

By the analysis of the above feature extraction methods for categorical data set in the outlier detection, we propose a feature extraction method with reference to its own data set. This feature extraction idea is based on the horizontal and vertical matching characteristics for categorical data. The horizontal refers the matching numbers between the data items of each data record in the horizontal direction and the other data records in the entire data set, and the vertical refers the matching numbers between the data items of each data record in the attribute column direction and the values of each attribute column.

Horizontal statistical characteristics $h_m(x_i)$: We compare data record x_i with other each record x_j in the data set D and calculate the number of matching values $match(x_i, x_j)$, then

the sum of all matching values can be calculated, and then we normalize it. The formula is expressed as follows.

$$h_m(x_i) = \frac{1}{N} \sum_{i \neq j, j=1}^{N} match(x_i, x_j) \quad (3)$$

$$match(x_i, x_j) = \sum_{k=1}^{d} S_k(x_{ik}, x_{jk}) \quad (4)$$

Vertical statistical characteristics $v_m(x_i)$: In each attribute column A_k of data set D, we calculate the value frequency $f_k(x_{ik})$ of each attribute value of data record x_i in $\mathcal{A}_k$, then the sum of all frequency values can be calculated, and then we normalize it. The formula is expressed as follows.

$$v_m(x_i) = \frac{1}{N * d} \sum_{k=1}^{d} f_k(x_{ik}) \quad (5)$$

The horizontal and vertical characteristics of categorical data can't be used to calculate the similarity or distance between the categorical data records, but their values can reflect the outlier characteristics of categorical data. So the horizontal and vertical feature values are larger, and the data is more normal; the feature values are smaller, and the data is more likely to be the outliers.

5. Cloud Model-based Outlier Detection for Categorical Data

In this section, we propose a novel outlier detection algorithm for categorical data, Cloud Model-based Outlier Detection (CMBOD), which consists of three steps. In the first step, we transform original data to numerical data. This transformation does not use the direct mapping method, but the probability mapping, which represents the probability of occurrence of corresponding attributes values in each attribute column A_l ($1 \leq l \leq d$). In the second step, we use the "cloud drop" data in the first step to generate a multi-dimensional cloud model by back forward cloud generator algorithm, which is the transformation from quantitative numerical value to the cloud with qualitative probability and represented as three digital characteristics $\{E_x, E_n, H_e\}$. In the third step, we use forward cloud generator algorithm to compute the certainty degree with respect to each cloud drop belonging to the established cloud model, and make the certainty degree as the outlier degree of data objects. The greater certainty degree represents the greater possibility belonging to the normal data; the smaller certainty degree represents the greater possibility as outliers. Table 1 contains the main notations used in the algorithms of this paper.

In the first stage, we present a data transformation algorithm, which transforms categorical data to numerical data, called "cloud drop" data. In order to describe categorical data in cloud model, we need to transform categorical data to the data with the meaning of cloud model, viz., the attribute values of each data record is belonging to the certainty degree of a domain of discourse, and values taken by each attribute have the physical meaning. Different categorical data values have different underlying meaning. According to the values in each attribute column, we can see the values constitute a set. A specific value in the set has a certain probability characteristics, and the high frequency values have the greater probability values. If the all attribute values of a data record are high, we can think that this categorical data record has a normal mode. Based on the above analysis, we transform each categorical attribute value to the probability of occurrence belonging to the set of categorical values corresponding to attribute column, and the transformed data is called "cloud drops" data set.

Algorithm: Data Transformation Algorithm (TransData)
Input: Categorical data set $D = \{x_1, x_2, \cdots, x_N\}$
Output: Cloud drop data set DC
step 1. Obtain the dimension d and size N of categorical data set D, initialize DC, $DC = zeros(N, d)$;
step 2. Calculate the attribute value set $\mathcal{A}_l$ contained by each A_l, and the frequency value $f_l(x_t)$ corresponding to $x_t \in \mathcal{A}_l$, stored as Map_l with (key x_t,value $f_l(x_t)$), calculate Map_l column by column and stored as $[Map_1, Map_2, \cdot\cdot, Map_d]$;
step 3. Set $i = 1$;
step 4. For x_{il} in $x_i = \{x_{i1}, x_{i2}, \cdots, x_{id}\}$, search for key-value pairs from corresponding Map_l and get $f_l(x_{il}) = f_l(x_t)$, where $x_{il} = x_t$, and store $DC(i, l) = f_l(x_{il})$;
step 5. Set $i = i + 1$, if $i \leq N$, jump to Step 4, otherwise end the loop;
step 6. Normalize the data in DC, $DC = DC/N$;
step 7. Output DC

Table 1. The Terminology of our Proposed Algorithm

Term	Description
D	Categorical data set $D = \{x_1, x_2, \cdots, x_N\}$
DC	Transformed cloud data set by D
N	Size of data set D
d	Number of attributes of data set D
k	Target input number of outliers
x_i	The *i*-th point in D $(1 \leq i \leq N)$
x_{il}	The value of attribute A_l of x_i
A_l	The *l*-th column attributes $(1 \leq l \leq d)$
$\mathcal{A}_l$	The attribute set contained by A_l in D
n_l	Number of different attributes value of $\mathcal{A}_l$ $(n_l = \lvert\mathcal{A}_l\rvert)$
$f_l(x_{il})$	Record number takes value x_{il} of A_l in D
$h_m(x_i)$	Record horizontal feature value in D
$v_m(x_i)$	Record vertical feature value in D

In the second step, we use back forward cloud generation algorithm to establish cloud model with an overall description of the data. In the first step, each categorical data record is transformed into the data of "cloud drop" by transformation algorithm, but these "cloud drop" data are separate from each other, and each cloud drop only represents an implementation belonging to the cloud model, which is reflected by all accurate cloud drops. For this reason, we propose a cloud model algorithm, which uses the cloud drops in the first step and the back forward cloud generation algorithm to calculate the entire cloud drops reflected by these accurate data, viz., the cloud model is reflected by categorical data set. The specific algorithm is shown as follows.

Algorithm: Building the Cloud Model Algorithm (CloudModelBuild)
Input: Cloud drop data set $DC = \{dropx_1, dropx_2, \cdots, dropx_N\}$
Output: Cloud model with the description of categorical data set (Ex,En,He)
step 1. According to $dropx_i$, calculate the sample mean of sample data:
$(\overline{X_1}, \cdots, \overline{X_d}) = (\frac{1}{N}\sum_{i=1}^{N} dropx_{i1}, \cdots, \frac{1}{N}\sum_{i=1}^{N} dropx_{id})$

Absolute central moment of the first order sample:
$(S_1, \cdots, S_d) = (\frac{1}{N}\sum_{i=1}^{N}|dropx_{i1} - \overline{X_1}|, \cdots, \frac{1}{N}\sum_{i=1}^{N}|dropx_{id} - \overline{X_d}|)$
Sample variance:
$(S_1^2, \cdots, S_d^2) = (\frac{1}{N-1}\sum_{i=1}^{N}(dropx_{i1} - \overline{X_1})^2, \cdots, \frac{1}{N-1}\sum_{i=1}^{N}(dropx_{id} - \overline{X_d})^2)$

step 2. Calculate $(E_{\hat{x_1}}, \cdots, E_{\hat{x_d}}) = (\overline{X_1}, \cdots, \overline{X_d})$

step 3. Calculate $(E_{\hat{n_1}}, \cdots, E_{\hat{n_d}}) = (\sqrt{\frac{\pi}{2}} \times \frac{1}{N}\sum_{i=1}^{N}|x_{i1} - E_{\hat{x_1}}|, \cdots, \sqrt{\frac{\pi}{2}} \times \frac{1}{N}\sum_{i=1}^{N}|x_{id} - E_{\hat{x_d}}|)$

step 4. Calculate $(H_{e1}, \cdots, H_{ed}) = (\sqrt{S_1^2 - E_{\hat{n1}}^2}, \cdots, \sqrt{S_d^2 - E_{\hat{n}d}^2})$

In the third stage, we obtain certainty degree of each cloud drop belonging to cloud model. Since the establishment of cloud model, we can represent the whole cloud model of these cloud drops, and then other cloud drops can directly substitute into this model to calculate the certainty degree belonging to the cloud model. From this perspective, if we take the existing data as training data set and establish a cloud model, we can calculate the certainty degree belonging to the cloud model for the data to be test, and the entire detection strategy is a semi-supervised outlier detection method. The method in this paper directly uses all data set to construct cloud model, and then calculates the certainty degree of each cloud drop by unsupervised detection strategy. The specific algorithm is shown as follows.

Algorithm: Computing the Certainty Degree (ComputeCertainty)

Input: Cloud drop data set $DC = \{dropx_1, dropx_2, \cdots, dropx_N\}$
Three digital characteristics (Ex,En,He)

Output: Certainty degree of cloud drop $Certainty = \{\mu(dropx_i), \cdots, (dropx_N)\}$

step 1. Set $i = 1$;

step 2. Calculate the certainty degree of $dropx_i$:
$y = e^{\frac{-(x-E_x)^2}{2E_n'^2}}$, $\mu(dropx_i) = e^{\sum_{l=1}^{d}\frac{-(x_{il}-E_{xl})^2}{2E_{nl}^2}}$

step 3. Set $i = i + 1$, if $i \leq N$, jump to Step 2; otherwise end the loop;

step 4. Store the certainty degree of each cloud drop, and return.
$Certainty = \{\mu(dropx_i), \cdots, (dropx_N)\}$

Through a combination of the above three stages, finally Cloud Model-based Outlier Detection (CMBOD) algorithm proposed in this paper can be established. There is an advantage of CMBOD that it avoids calculating the similarity of categorical data. It directly uses the characteristics of categorical data to construct cloud model, and then identifies the certainty degree of each categorical data belonging to the cloud model. In order to improve the accuracy of detection, we can simultaneously extract the horizontal and vertical features of categorical data in the transformation data algorithm of TransData in the first stage, and the horizontal and vertical features are incorporated into the cloud drop data set. The final cloud model-based outlier detection algorithm is shown as follows.

Algorithm: Cloud Model-based Outlier Detection Algorithm (CMBOD)

Input: Categorical data set $D = \{x_1, x_2, \cdots, x_N\}$, target number of outliers k

Output: k detected outliers $OutSet$

step 1. Obtain cloud drop data DC by algorithm TransData;

step 2. Merge horizontal and vertical feature of each categorical data, generate new cloud drop data $DC = [DC, h_m, v_m]$

step 3. Calculate the three digital characteristics (Ex,En,He) of cloud model with the description of categorical data set by algorithm CloudModelBuild;

step 4. Calculate the certainty degree of cloud drop by algorithm ComputeCertainty

$Certainty = \{\mu(dropx_i), \cdots, (dropx_N)\}$, as the outlier degree of the data;

step 5. Rank $Certainty$ with ascending order;

step 6. Return top k data with the minimum outlier degree as $OutSet$.

6. Experiments and Analysis

In this section, we conduct a series of experiments on the synthetic data set and UCI standard data set to validate the cloud model-based outlier detection algorithm for categorical data proposed in this paper. As a comparison, we implement the four algorithms for categorical data set described in this paper; according to the algorithm described in the literature, we use Matlab to implement Apriori algorithm. The experimental environment of this paper is as follows: Pentium（R）Dual-Core 2.0GHz CPU, 2.0GB memory, Windows XP Professional.

6.1. Experimental Setup

A. Real data set

Wisconsin Breast Cancer: This data set contains 699 records and 9 attributes. Each record is marked as benign or malignant. According to the method of Harkins in literature [25], we take the records marked with malignant as outlier category, and extract the records with the interval of 6 records. Finally, the experimental data set that we have obtained contains 39 outlier records (8%) and 444 normal data records (92%).

Lymphography: This data set contains 148 records and 19 attributes (including categorical attributes). There are a total of 4 class clusters, and the number of data in the 1th and 4th class cluster is less, we take them as outliers (4%).

Post-operative: This data set is used to determine the nursing department of the patients after surgery (including Intensive Care Unit, Home, general hospital floor). This data set contains 90 data records and 9 attributes (including categorical attributes). We take the first and second class cluster as outliers (including a total of 26 outliers, 29%).

Pageblocks: This data set contains 5473 data records and 10 attributes. There are 5 class clusters in the data set, and a class cluster data among them account for 90% in the entire data set, so we take the rest other class cluster data as outliers. Because this data set is continuous numeric data, we use the discrete method with equal frequency to transform these continuous data to categorical data. We remove half of the outlier data, and take the rest 280 data records as the final outliers.

Adult: This data set contains 48842 data records and 14 attributes. Because this data set is continuous numeric data set, we use the discrete method with equal width to make them discretization, and take the income data that is greater than 50K per year as outlier data (about 24% of the entire data set). In order to get fewer outliers, we extract the records with the interval of 6 records to get the final outliers from the above outlier data set.

B. Artificial data set

For the experiment on artificial data set, our main target is to test the speed and scalability of the comparison algorithms, while the accuracy (detection rate and false alarm rate) of the algorithms do not be considered. We use the software provided by Cristofor in [26] to generate the artificial data set, and the main purpose of the experiment is to test the change of performance of different algorithms when the dimension and the total amount of data set

changed. The data set is generated from two aspects. The first aspect is to fix dimension, and increase the total amount of data. In our experiment, the dimension is 10 and the outlier scale is 30, while the total data from 1K to 800K respectively generates multiple simulation data. The second aspect is to fix the total amount of data, and change dimension. In our experiment, the data amount is fixed to 100K, the outlier scale is 30, and the dimension from 2 to 40 generates multiple simulation data sets.

6.2. Analysis for Results

Tables 2 as well as Figure 3 and 4 show the result of outlier detection of each algorithm on real data set and artificial data set. Based on the analysis of detection results of various algorithms for real data set, for the data set of Wisconsin Breast Cancer and Lymphography, the detection rate and false alarm rate and AUC are acceptable; but for Post-operative and Pageblocks and Adult, the detection rate and false alarm rate and AUC are undesirable. Based on the characteristics of the data set itself, the ratio of the outlier data and the total data in Wisconsin Breast Cancer and Lymphography is less than 10%. At the same time, the data in the set originally is categorical data, which has not been discretized, so the result is better. For Post-operative data set, the outlier ratio accounts for 29%, and it has only 90 records in the whole data set, which is inadequate as training and learning data. Especially for the algorithm that needs to extract the characteristics of categorical data for learning, the effect is poor. Pageblocks and Adult have a great amount of data, but their own data type is numeric type data, which will lose information after discretization, so the result is poor. But for CMBOD algorithm as well as the comparison algorithm proposed in this paper, in the case of a fair comparison, we can see that the detection effect of CMBOD and EBGOD algorithm is better than other outlier detection algorithms.

Figure 3 and Figure 4 show the experiment of various algorithms for the scalability of the data dimensions and data size. Along with data size increase, when data size N= 700K, the detection time of each algorithm are AVFOD 1.33 seconds, CMBOD 22.86 seconds, EntropyBasedGreedyOD 184.94 seconds, FPOFOD 564.8 seconds, InverseNotFPOFOD 3127.22 seconds. We can see the CPU execution time of AVFOD is the shortest, and the reason is the algorithm extracts only the vertical frequency characteristics of the categorical data set. When the data set is relatively simple and the amount of data is sufficient, using the algorithm can meet needs. Although CMBOD proposed in this paper extracts the horizontal and vertical characteristics, but the total time spent is 22.86 seconds, which has much higher efficiency compared to other three algorithms. Through a balanced comparison in two aspects of detection effect and operational efficiency, CMBOD has a larger advantage. For the two outlier detection algorithm based on frequent itemsets, their execution time is longer, which is not practical for large-scale data. For the experiment with increasing data dimension, when the dimension d= 40, and the data size N= 100K, the detection time of each algorithm are AVFOD 2.6178 seconds, CMBOD 13.613 seconds, FPOFOD 21.466 seconds, InverseNotFPOFOD 109.95 seconds, EntropyBasedGreedyOD 107.85 seconds. We can see the running time of InverseNotFPOFOD and EntropyBasedGreedyOD is sensitive to the increase of dimension, which is not suitable for high-dimensional data set. AVFOD and CMBOD increase linearly with time, which can meet the change of dimension.

Table 2. The Results on UCI Data Set

Data set	Metric	CMBOD	EBGOD	AVFOD	FPOFOD	INFPOFOD
Wisconsin Breast Cancer	DR	**0.8462**	**0.8462**	0.7949	0.7948	0.8461
	FR	**0.0135**	0.0158	0.018	0.0202	0.0157
	AUC	**0.994**	0.9933	0.9905	0.989	0.9917
Lymphography	DR	0.6667	**0.8333**	0.6667	0.6667	0.6667
	FR	0.0141	**0.007**	0.0141	0.1408	0.1408
	AUC	0.9883	**0.9977**	0.9906	0.9877	0.9877
Post-operative	DR	**0.36**	0.3077	0.28	0.276	0.276
	FR	**0.2581**	0.3437	0.2903	0.3452	0.3452
	AUC	**0.5768**	0.5228	0.4916	0.4877	0.4913
Pageblocks	DR	**0.5632**	0.4643	0.3964	0.225	0.225
	FR	**0.0305**	0.0345	0.0392	0.0481	0.0481
	AUC	**0.8924**	0.8816	0.8823	0.6371	0.6371
Adult	DR	**0.2346**	0.09154	0.1001	0.07923	0.07441
	FR	0.0386	**0.01849**	0.0494	0.01917	0.01943
	AUC	**0.6574**	0.5365	0.5697	0.5301	0.5274

7. Conclusions

This paper first introduces the basic concept of cloud model. Then, we survey the current typical outlier detection algorithms for categorical data. We analyze the difference between the outlier detection algorithms suitable for categorical data and no suitable for categorical data, and suggest that the outlier detection algorithms suitable for categorical data are required to use the data set own characteristics to detect the outliers. Based on the above analysis and combining with the application of cloud model, we propose a cloud model-based outlier detection algorithm for categorical data. Through the comparative experiments in real data set and artificial data set, the CMBOD algorithm proposed in this paper has advantages in detection correctness (higher detection rate and lower false alarm rate), which also has a better scalability for the data set scale and dimension increase. In addition, the proposed algorithm in this paper does not require the user to provide the parameters, which has better practicality than frequent itemset-based algorithms.

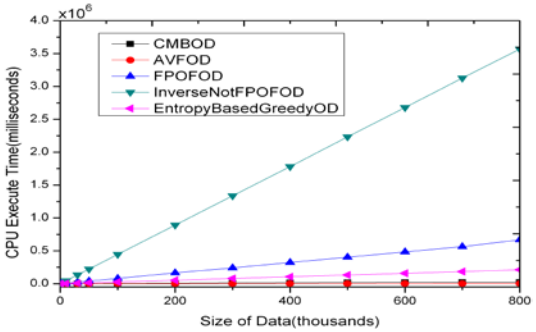

Figure 3. The Performance of Algorithms on the Artificial Data Sets as Number Increasing

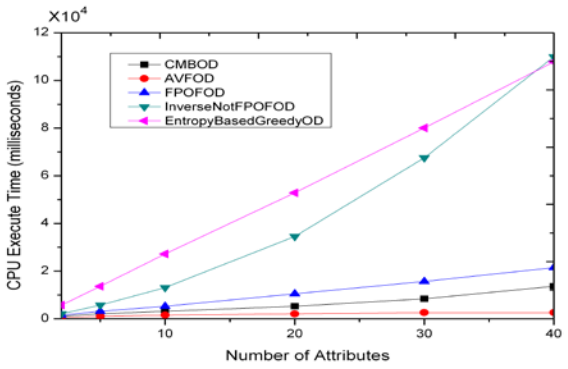

Figure 4. The Performance of Algorithms on the Artificial Data Sets as Dimensionality Increasing

Acknowledgments

This paper is supported by the following foundations or programs, including National Natural Science Foundation of China (61171060, 61073058, 61075019, 61201383), Natural Science Foundation of Chongqing of China (cstc2012jjA40027), Natural Science Foundation of Education Commission of Chongqing of China (KJ130527), Doctoral Scientific Research Foundation of Chongqing of University of Posts and Telecommunications (A2013-01).

References

[1] D. Yue, X. Wu, Y. Wang, Y. Li, C.-H. Chu and Ieee, "A Review of Data Mining-based Financial Fraud Detection Research", Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing, vol. 1-15, **(2007)**, pp. 5519-5522.
[2] Z. Jiong and Z. Mohammad, "Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection", Proceedings of IEEE International Conference on Communications, ICC '06, **(2006)**, pp. 2388-2393.
[3] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review", Statistical Science, vol. 17, no. 3, **(2002)**, pp. 235-255.
[4] V. Podgorelec, M. Hericko and I. Rozman, "Improving mining of medical data by outliers prediction", Proceedings of 18th IEEE Symposium on Computer-Based Medical Systems, **(2005)**, pp. 91-96.
[5] M. Kantardzic, "Data Mining Concepts, Models, Methods, and Algorithms", Wiledy Interscience Publications; IEEE Press, **(2003)**.
[6] E. M. Knorr and R. T. Ng, "Algorithms for Mining Distance-Based Outliers in Large Datasets", Proceedings of the 24rd International Conference on Very Large Data Bases, **(1998)**, pp. 392-403.
[7] D. M. Hawkins, Identification of outliers London; New York: Chapman and Hall, **(1980)**.
[8] H.-P. Kriegel, M. Schubert and A. Zimek, "Angle-based outlier detection in high-dimensional data", Proceedings of 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, **(2008)** August 24-27, pp. 444-452.
[9] C. E. Shannon, "A mathematical theory of communication", ACM SIGMOBILE Mobile Computing and Communications Review, vol. 5, no. 1, **(2001)**, pp. 3-55.
[10] Z. Y. He, S. C. Deng and X. F. Xu, "An optimization model for Outlier detection in categorical data", Proceedings of Advances in Intelligent Computing, **(2005)**, pp. 400-409.
[11] Z. Y. He, S. C. Deng, X. F. Xu and J. Z. X. Huang, "A fast greedy algorithm for outlier mining", Proceedings of Advances in Knowledge Discovery and Data Mining, **(2006)**, pp. 567-576.
[12] Z. He, X. Xu, J. Huang and S. Deng, "FP-Outlier: frequent pattern based outlier detection", Computer Science and Information Systems, vol. 2, no. 1, **(2005)**, pp. 103-118.
[13] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", Proceedings of the international conference on very large data bases, **(1994)**, pp. 487-499.
[14] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. I. Verkamo, "Fast discovery of association rules", Advances in Knowledge Discovery and Data Mining, vol. 12, **(1996)**, pp. 307-328.

[15] J. Han, J. Pei and Y. Yin, "Mining frequent patterns without candidate generation", Proceedings of the 2000 ACM SIGMOD international conference on Management of data, Dallas, Texas, United States, **(2000)**, pp. 1-12.
[16] M. E. Otey, A. Ghoting and S. Parthasarathy, "Fast distributed outlier detection in mixed-attribute data sets", Data Mining and Knowledge Discovery, vol. 12, no. 2-3, **(2006)**, pp. 203-228.
[17] A. Koufakou, E. G. Ortiz, M. Georgiopoulos, G. C. Anagnostopoulos, and K. M. Reynolds, "A scalable and efficient outlier detection strategy for categorical data", Proceedings of 2007 19th IEEE International Conference on Tools with Artificial Intelligence, **(2008)**, pp. 210-17.
[18] L. Deyi, "Uncertainty in knowledge representation Engineering Science", (in chinese), vol. 2, no. 10, **(2000)**, pp. 73-79.
[19] L. Deyi, L. Changyu, D. Yi and H. Xu, "Uncertainty of artificial intelligence", Journal Of Software, (in chinese), vol. 15, no. 9, **(2004)**, pp. 1583-1592.
[20] L. Deyi and L. Changyu, "Discuss the universality of normal cloud model Engineering Science", (in chinese), vol. 6, no. 8, **(2004)**, pp. 28-33
[21] L. Deyi and D. Yi, "Uncertainty of artificial intelligence", Beijing: National Defence Industry Press, (in chinese), **(2005)**.
[22] Z. Y. He, X. F. Xu and S. C. Deng, "Discovering cluster-based local outliers", Pattern Recognition Letters, vol. 24, no. 9-10, **(2003)** June, pp. 1641-1650.
[23] C. Stanfill and D. Waltz, "Toward memory-based reasoning", Communications Of The Acm, vol. 29, no. 12, **(1986)**, pp. 1213-1228.
[24] S. Boriah, V. Chandola and V. Kumar, "Similarity measures for categorical data: A comparative evaluation", Proceedings of the 8th SIAM International Conference on Data Mining, **(2003)**, pp. 243-254.
[25] S. Hawkins, H. He, G. J. Williams and R. A. Baxter, "Outlier Detection Using Replicator Neural Networks", Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery, **(2002)**, pp. 170-180.
[26] D. Cristofor and D. A. Simovici, "Finding median partitions using information-theoretical-based genetic algorithms", Journal of Universal Computer Science, vol. 8, no. 2, **(2002)**, pp. 153-172.

Authors

Dajiang Lei received his M.Sc. in computer application (2006) from the School of Computer Science, Wuhan University of Science & Technology and PhD in computer science (2009) from Chongqing University, China. Now he is lecturer of informatics at School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, China. His current research interests include different aspects of data mining and intelligent computing.

Liping Zhang received his M.Sc. in computer application (2009) from the School of Computer Science, Wuhan University of Science & Technology. Now she is lecturer of informatics at College of Mobile Telecommunications, Chongqing University of Posts and Telecommunications, China. Her current research interests include different aspects of Data Mining and Information Security.

Lisheng Zhang received his M.Sc. in Mathematics (1993) from Southwest Teachers University. Now he is associate professor of informatics at School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, China. His current research interests include different aspects of Data Mining and Intelligence Information System.