Proceedings of the 4th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'18)
Madrid, Spain – August 21 – 23, 2018
Paper No. CIST 105
DOI: 10.11159/cist18.105

Data Anomaly Correction for Massive Customer Data

Wang Yuan, Vincent Ng
Department of Computing, the Hong Kong Polytechnic University
Hong Kong, China
cstyng@comp.polyu.edu.hk, 16054091@polyu.edu.hk,12900110r@connect.polyu.hk

Abstract **-** Data anomaly problems exist in all aspects of social life, and have become an obstacle to data mining. It would reduce the accuracy of decision making, or even result with wrong decisions. Solution of this problem is urgently needed in commercial banks where huge economic losses are caused by data anomalies, such as credit fraud problems. In this paper, we describe a novel method for data anomaly correction based on our previous work of anomaly detection. Through the use of artificial neural networks, key attributes that have great influence with abnormal values will be adjusted automatically. Preliminary experiments have been done with some customer data from a commercial bank and they have interesting results.

Keywords**:** Data Anomaly, Correction, Neural Network, Big Data.

1. Introduction

Many data mining methods regard anomalies as noise and remove them, but in some specific situations, such as fraud detection or illegal intrusion detection, rarely happening events are often more valuable than the frequent occurrence events. There are many important attributes in the original data, if we only simply abandon the ones with anomalies, this may change the validity of the data mining results. Therefore, it is better to good methods to correct the anomalies for later processing. In our work, we focus on customer data in a commercial bank. We observed there are many anomalies, which occur in different forms as follow:

- Data error: wrong format (for example: age is 12-3 or 12,3), random code, out of range, and have great influence on the analysis results, etc.;
- Missing values: This is one of the most common forms;
- Duplicate values: The same record appears twice or more times. When conducting clustering analysis, the distance between the duplicate values is 0, and one record is to be retained and removed the others; and
- Data conflict: Two records at the same time should be retained, and we cannot judge which data is correct.

In our work, we come up with a novel data anomaly correction method based on neural network, which uses the clustering result (normal and abnormal label for training), and find the key attributes which have great influence on the normality and abnormality of the data, then adjust the value of each anomaly until it becomes normal.

2. Background Review

Many research work focus on identification or detection of anomalies [1], and a few on the study of anomaly correction. The most common and simplest way to deal with missing data is to use deletions, which is also the default method to handle missing values in some common statistical software such as SPSS and SAS [2]. In this method, if any of the variables contain missing values, the corresponding record is removed from the analysis. If the proportion of missing values is relatively small, this method would be effective. The views of experts on the specific number of missing the proportion are different. Some scholars believe that it should be below 5%, and some think that around 20% is fine. However, this method has a lot of limitations. It would reduce the number of data samples and may result with a large number of waste of resources. In the case of a small sample size, the deletion of a few data objects is sufficiently affecting the correctness of the results. Thus, when the proportion of missing values is large, especially when the missing values are not randomly distributed, this conventional method may lead to data deviation which would lead to wrong conclusions.

In order to solve the problem aforementioned, imputation techniques have been developed. Common methods for multiple imputations contain propensity score (PS) and predictive mean matching (PMM), etc [3]. The advantage of these

methods is that the relationship between variables can be better maintained by simulating the distribution of missing data. In the field of database research, the authors in [4] propose a method for measuring the similarities between normal records and records with missing values and construct a logarithm linear model for data with multi-dimensions to fill missing data [5].

For data with conflicting form (), it occurs during the integration of customer data from different data sources. The same attribute of a customer record may have two different values. For example, the application amount exists in the customer information relational table and the loan application relational table. The way to obtain an intelligent estimated value of conflicting data is a problem often faced with in data cleaning [6]. Currently, common methods are sorting, fusion and approaches based on rules [7]. The DIRECT system divides data confliction into context independent conflicts and context dependent conflicts [8]. Context dependent conflicts are caused by the inherent data design and expression factors in data from different systems and applications. These data conflicts must be solved with data conversion rules. Context independent conflicts refer to inconsistency often caused by some external and random factors. To solve these data conflicts, we often need some human intervention and specific methods. DIRECT proposes that context dependent conflicts can be solved by machine learning technologies and makes feature evaluations, which means to use the linear combination of each feature, and to determine which value is the only correct according to the evaluation value [9]. Trio system from Stanford University develops a complete relational model to process lineage and inconsistent data [10], which provides a new idea of processing conflict data from the perspective of data model.

The approaches studied previous few related massive anomaly data correct automatically. In our study, we conduct a new method of data anomaly correction.

3. Neural Network Design

Two years ago, a survey has been done by our research team about anomalous data from a commercial bank with 500 customers. As shown in Fig. 1 of the survey results, there are about 72 anomalous data records (mainly occurred in one or two attributes) with different types of data anomalies. We propose the PCDP algorithm on detecting data anomalies by parallelizing the CDP algorithm with the support of multiple machines. Through preliminary experiments, the PCDP algorithm has demonstrated its fast computation as well as retaining the anomaly detection accuracy for customer data of a commercial bank. (NOT UNDERSTAND AS INDICATED IN THE PREVIOUS COMMENT)

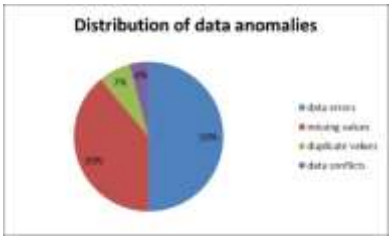

Fig. 1: A survey conducted for customer records in a commercial bank.

The methods of anomaly detection based on matrix decomposition [11,12] and Replicator Neural Networks [12] all mention the concept of error reconstruction. If the reconstruction error is out of certain range, we can regard the respective data as an anomaly. That is to say, if we adjust the anomalous value reversely until the reconstruction error within the range of criterion for anomaly detection, we would consider the data is having a normal value. Here, in our work, we propose to use the neural network for implementing this idea.

Neural networks [13] are computing systems inspired by the biological neural networks that constitute animal brains. A neural network is based on a collection of connected units called neurons, (analogous to axons in a biological brain). Each connection between neurons can transmit a signal to another neuron. The receiving neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have state, generally represented by real numbers. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream. Further, they may have a threshold such that only if the aggregate signal is below (or above) that level is the downstream signal sent. Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Over time, attention focused on matching specific mental abilities, leading to deviations from biology such as backpropagation, or passing information in the reverse direction and adjusting the network to reflect that information [11]. Neural networks have been used on a variety of tasks, including computer vision, speech

recognition, machine translation, social network filtering, playing board and video games, medical diagnosis and in many other domains.

The key idea of anomaly correction here is to train a powerful classifier based on multi-layer perception and then find the key attributes/characteristics which determine the normality of the data by the trained network, finally adjust these attributes/characteristics to turn the output to be normal. Every input value is classified to be normal or abnormal. Once the classifier gave an abnormal output, we replace the original value with a new one by adding a perturbation such that the new value is within the values of the respective normal range. In our design, the number of nodes in the output layer of the neural network is L_2. Its loss between output of classifier and the adjusted value is calculated. Then we can work out the gradient of this L_2 loss towards the input. A large value of the gradient of an attribute reflects that it is a key attribute to determine the data records as anomalies. Hence, after adjusting these types of attribute values in data records, if the data records turn out to be normal, then these attribute values can be regarded as the corrected values. The process flow of anomaly correction based on Neural Network is depicted as in Figure 2.

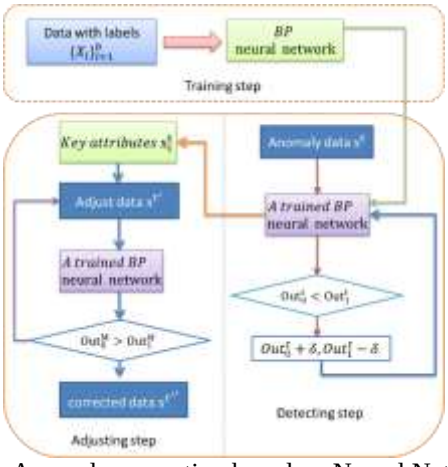

Fig. 2: Anomaly correction based on Neural Network.

With the above idea, we first train a classifier based on Multi-layer perception via back-propagation [12,13] as shown in Figure 3. The basic steps of the classifier training is provided in Algorithm CT.

Classifier Training Algorithm (CT)

1. Acquire mini-batch data set $S' = \{X_i\}_{i=1}^{b}$, $X_i = [x_1^t, x_2^t, \dots x_m^t]$ and their corresponding labels $\{y_i\}_{i=1}^{b}$, $y_i \in \{0,1\}$. Each data record is represented as a m-dimension vector;
2. Initialize the network
   a. The number of nodes in the input layer is I;The number of nodes in the first hidden layer is J, and the second is K;The number of nodes in the output layer is layer 2; The weight of the input layer to the first hidden layer is ω_{ij} and the bias is a_j; The weight of the first hidden layer to the second hidden layer is ω_{jk} and the bias is b_k, while the second hidden layer to the output layer is ω_{kl} and the bias is c_l, Learning rate is η, the activation function is the Sigmoid function $f(x) = \frac{1}{1+e^{-x}}$
   b. Here, we construct a two-hidden layers network, as Fig.3 shows, $i = 1,2,\cdots,I;\ j = 1,2,\cdots,J;\ k = 1,2,\cdots,K;\ l = 1,2;$
3. Calculate the output of each layer:
   a. The output of the first hidden layer $H_j' = f\left(\sum_{i=1}^{I} \omega_{ij} x_i + a_j\right)$;
   b. The output of the second hidden layer $H_k'' = f\left(\sum_{j=1}^{J} \omega_{jk} H_j' + b_k\right)$;
   c. The output of the output layer $O_l = f(\sum_{k=1}^{K} \omega_{kl} H_k'' + c_l)$;
4. Specifically, we denote the output of this classifier by $Out = [Out_0, Out_1] \in R^2$, where $Out_0 > Out_1$ indicates that the input data record is normal and vice versa. By the softmax, $Out_0 = \frac{e^{O_1}}{e^{O_2}+e^{O_1}}$, and $Out_1 = \frac{e^{O_2}}{e^{O_2}+e^{O_1}}$, we calculate the softmax loss between label y_i and output Out^i by $L = -\frac{1}{b}\sum_{i=1}^{b}[y_i log(Out_0^i) + (1-y_i) log(Out_1^i)]$;
5. Calculate the gradients of weights of each layer:

a. The second hidden layer: $\frac{\partial L}{\partial \omega_{kl}} = -\frac{1}{b}\sum_{i=1}^{b} R \times H_k''$.

The first hidden layer: $\frac{\partial L}{\partial \omega_{jk}} = -\frac{1}{b}\sum_{i=1}^{b} R \times H_k'' \,(1 - H_k'')H_j' \sum_{k=1}^{K} \omega_{kl}$

b. The input layer:

$\frac{\partial L}{\partial \omega_{ij}} = -\frac{1}{b}\sum_{i=1}^{b}(R \times \sum_{k=1}^{K} \omega_{kl})\, H_k'' \,(1 - H_k'')(\sum_{j=1}^{J} \omega_{jk})\, H_j'(1 - H_j')x_i$

6. Calculate the gradients of biases of each layer

a. The second hidden layer: $\Delta c_l = \frac{\partial L}{\partial c_l} = -\frac{1}{b}\sum_{i=1}^{b} R$

b. The first hidden layer: $\Delta b_k = \frac{\partial L}{\partial b_k} = \frac{\partial L}{\partial H_k''} \times \frac{\partial H_k''}{\partial b_k} = -\frac{1}{b}\sum_{i=1}^{b} R(\sum_{k=1}^{K} \omega_{kl})\, H_k''(1 - H_k'')$

c. The input layer: $\Delta a_j = \frac{\partial L}{\partial a_j} = \frac{\partial L}{\partial H_k''} \times \frac{\partial H_k''}{\partial H_j'} \times \frac{\partial H_j'}{\partial a_j} = -\frac{1}{b}\sum_{i=1}^{b}(R \times \sum_{k=1}^{K} \omega_{kl})\, H_k'' \,(1 - H_k'')(\sum_{j=1}^{J} \omega_{jk})\, H_j'(1 - H_j')$

7. Update the weights and biases of all layers via Stochastic Gradient Descent [13]:

a. $\omega_{kl} = \omega_{kl} - \eta\Delta\omega_{kl},\ c_l = c_l - \eta\Delta c_l$;

b. $\omega_{jk} = \omega_{jk} - \eta\Delta\omega_{jk},\ b_k = b_k - \eta\Delta b_k$;

c. $\omega_{ij} = \omega_{ij} - \eta\Delta\omega_{ij},\ a_j = a_j - \eta\Delta a_j$;

8. Calculating the second hidden layer

a. $\frac{\partial L}{\partial \omega_{kl}} = \frac{\partial\left(-\frac{1}{b}\sum_{i=1}^{b}\left[y_i \log\left(Out_0^i\right) + (1-y_i)\log\left(Out_1^i\right)\right]\right)}{\partial \omega_{kl}} = \frac{\partial\left(-\frac{1}{b}\sum_{i=1}^{b}\left[y_i \log\left(\frac{e^{O_2}}{e^{O_2}+e^{O_1}}\right) + (1-y_i)\log\left(\frac{e^{O_2}}{e^{O_2}+e^{O_1}}\right)\right]\right)}{\partial \omega_{kl}} = \frac{\left(-\frac{1}{b}\sum_{i=1}^{b}\left[y_i(O_1 - O_2) + O_2 - \log(e^{O_2}+e^{O_1})\right]\right)}{\partial \omega_{kl}}$.

b. $\frac{\partial f(x)}{\partial x} = \frac{\partial\left(\frac{1}{1+e^{-x}}\right)}{\partial x} = f(x) \times (1 - f(x))$,

c $\frac{\partial O_1}{\partial\, \omega_{kl}} = \frac{\partial f(\sum_{k=1}^{K} \omega_{kl} H_k'' + c_l)}{\partial \omega_{kl}} = \frac{\partial\left(\frac{1}{1+e^{-\sum_{k=1}^{K} \omega_{kl} H_k'' + c_l}}\right)}{\partial \omega_{kl}} = O_1(1 - O_1)H_k''$ and similarly, $\frac{\partial O_2}{\partial \omega_{kl}} = (1 - O_2)H_k''$.

Substituting $\frac{\partial O_1}{\partial \omega_{kl}}$ and $\frac{\partial O_2}{\partial \omega_{kl}}$ into $\frac{\partial L}{\partial \omega_{kl}}$, we get the gradient of weight of the second hidden layer $\Delta\omega_{kl} = \frac{\partial L}{\partial \omega_{kl}} = -\frac{1}{b}\sum_{i=1}^{b}[(y_i - 1)O_1(1 - O_1) - y_i O_2(1 - O_2)]H_k''$. In order to simplify the formula and reuse the repetitive operator, we let $R = (y_i - 1)O_1(1 - O_1) - y_i O_2(1 - O_2)$ at the end.

9. Calculating the first hidden layer

For the first hidden layer, we have $\frac{\partial L}{\partial \omega_{jk}} = \frac{\partial L}{\partial H_k''} \times \frac{\partial H_k''}{\partial \omega_{jk}}$, $\frac{\partial L}{\partial H_k''} = \frac{\partial\left(-\frac{1}{b}\sum_{i=1}^{b}\left[y_i(O_1 - O_2) + O_2 - \log(e^{O_2}+e^{O_1})\right]\right)}{\partial H_k''}$ and $\frac{\partial O_1}{\partial H_k''} = O_1(1 - O_1)\frac{\partial f(\sum_{k=1}^{K} \omega_{kl} H_k'' + c_l)}{\partial H_k''} = O_1(1 - O_1)\sum_{k=1}^{K} \omega_{kl}$. Similarly, $\frac{\partial O_2}{\partial H_k''} = O_2(1 - O_2)\sum_{k=1}^{K} \omega_{kl}$, $\frac{\partial H_k''}{\partial \omega_{jk}} = \frac{\partial f\left(\sum_{j=1}^{J} \omega_{jk} H_j' + b_k\right)}{\partial \omega_{jk}} = H_k''(1 - H_k'')H_j'$. By substituting $\frac{\partial O_1}{\partial H_k''}$ and $\frac{\partial O_2}{\partial H_k''}$ into $\frac{\partial L}{\partial \omega_{jk}}$, we get the gradient of weight of the first hidden layer $\Delta\omega_{jk} = \frac{\partial L}{\partial \omega_{jk}} = -\frac{1}{b}\sum_{i=1}^{b} R \times H_k''\,(1 - H_k'')H_j' \sum_{k=1}^{K} \omega_{kl}$.

10. Calculating the input layer, we have $\frac{\partial L}{\partial \omega_{ij}} = \frac{\partial L}{\partial H_k''} \times \frac{\partial H_k''}{\partial H_j'} \times \frac{\partial H_j'}{\partial \omega_{ij}}$, and $\frac{\partial H_k''}{\partial H_j'} = \frac{\partial f\left(\sum_{j=1}^{J} \omega_{jk} H_j' + b_k\right)}{\partial H_j'} = H_k''(1 - H_k'')\sum_{j=1}^{J} \omega_{jk})$, $\frac{\partial H_j'}{\partial \omega_{ij}} = H_j'(1 - H_j')x_i$, By substituting $\frac{\partial H_k''}{\partial H_j'}$, $\frac{\partial L}{\partial H_k''}$ and $\frac{\partial H_j'}{\partial \omega_{ij}}$ into $\frac{\partial L}{\partial \omega_{ij}}$, we get the gradient of weight $\Delta\omega_{ij} = \frac{\partial L}{\partial \omega_{ij}} = -\frac{1}{b}\sum_{i=1}^{b}(R \times \sum_{k=1}^{K} \omega_{kl})\, H_k'' \,(1 - H_k'')(\sum_{j=1}^{J} \omega_{jk})\, H_j'(1 - H_j')x_i$.

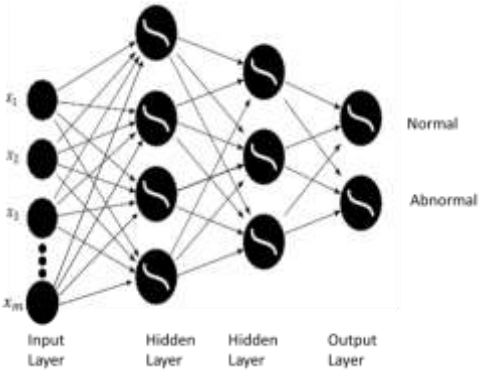

Fig. 3: The structure of anomaly correction based on Neural Network.

The trained classifier is denoted as F, i.e. $\mathrm{Out} = \mathrm{F(x)}$, and we can use F to detect abnormal attributes of a given abnormal record. Suppose that we are given a new record $\mathrm{x^t} = [x_1^t, x_2^t, \dots x_m^t]$, and m is also the number of layers of the input. The output of classifier is $\mathrm{Out^t} = \mathrm{F(x^t)} = [\mathrm{Out_0^t}, \mathrm{Out_1^t}]$. If $\mathrm{Out_0^t} < \mathrm{Out_1^t}$, the given record $\mathrm{x^t}$ is abnormal, which is what we want to deal with. We want to detect which attribute(s) in $\mathrm{x^t}$ is the key reason to the abnormal data record, and then adjust it to normal. Here, the basic steps of Detecting Stage are as follows:

Anomaly Detection Algorithm (AD)

1. For a new record $\mathrm{X^t} = [x_1^t, x_2^t, \dots x_m^t]$ and its output of the classifier $\mathrm{Out^t} = \mathrm{F(X^t)} = [\mathrm{Out_0^t}, \mathrm{Out_1^t}]$, if $\mathrm{Out_0^t} < \mathrm{Out_1^t}$, we construct an output with small perturbation $\delta > 0$, i.e. $\widehat{\mathrm{Out}} = [Out_0^t + \delta, Out_1^t - \delta]$. We want to modify the input $\mathrm{x^t}$ and let $\mathrm{Out^t} \to \widehat{\mathrm{Out}}$, i.e. push the output towards normal output;
2. Construct a loss $\mathrm{L_2} = \frac{1}{2}\left|\left|Out^t - \widehat{Out}\right|\right|_2^2$;
3. Calculate the gradient of $\mathrm{L_2}$ with respect to input $\mathrm{x^t}$ by $\frac{\partial \mathrm{L_2}}{\partial X^t} = \frac{\partial \left|\left|Out^t - \widehat{Out}\right|\right|_2^2}{\partial X^t} = \left(Out^t - \widehat{Out}\right)\frac{\partial Out^t}{\partial X^t}$.
   a. Here, when we construct the output $\mathrm{Out^t}$ with small perturbation δ, $\widehat{Out}$ is a constant, which can be approximated by $\mathrm{Out^t}$ with small perturbation δ, so $\frac{\partial \widehat{Out}}{\partial X^t} = \vec{0}$.

And $\frac{\partial Out_0^t}{\partial X^t} = \frac{\partial\left(\frac{e^{O_1}}{e^{O_2}+e^{O_1}}\right)}{\partial X^t} = \frac{1}{e^{O_1-O_2+2}+e^{O_2-O_1}} \times \frac{\partial(O_1-O_2)}{\partial X^t}$, $\frac{\partial Out_1^t}{\partial X^t} = \frac{\partial\left(\frac{e^{O_2}}{e^{O_2}+e^{O_1}}\right)}{\partial X^t} = \frac{1}{e^{O_1-O_2+2}+e^{O_2-O_1}} \times \frac{\partial(O_2-O_1)}{\partial X^t}$,

$\frac{\partial O_1}{\partial X^t} = O_1(1-O_1)(\sum_{k=1}^{K}\omega_{kl})H_k''(1-H_k'')(\sum_{j=1}^{J}\omega_{jk})H_j'(1-H_j')\sum_{i=1}^{I}\omega_{ij}$. Similarly, $\frac{\partial O_2}{\partial X^t} = O_2(1-O_2)(\sum_{k=1}^{K}\omega_{kl})H_k''(1-H_k'')(\sum_{j=1}^{J}\omega_{jk})H_j'(1-H_j')\sum_{i=1}^{I}\omega_{ij}$.

Thereby, $\frac{\partial Out_0^t}{\partial X^t} = \frac{(O_1-O_2)(1-O_1-O_2)(\sum_{k=1}^{K}\omega_{kl})H_k''(1-H_k'')(\sum_{j=1}^{J}\omega_{jk})H_j'(1-H_j')\sum_{i=1}^{I}\omega_{ij}}{e^{O_1-O_2+2}+e^{O_2-O_1}}$,

$\frac{\partial Out_1^t}{\partial X^t} = \frac{(O_2-O_1)(1-O_1-O_2)(\sum_{k=1}^{K}\omega_{kl})H_k''(1-H_k'')(\sum_{j=1}^{J}\omega_{jk})H_j'(1-H_j')\sum_{i=1}^{I}\omega_{ij}}{e^{O_1-O_2+2}+e^{O_2-O_1}}$. So,

$$\frac{\partial \mathrm{L_2}}{\partial X^t} = \left(Out^t - \widehat{Out}\right)\frac{\partial Out^t}{\partial X^t} = (-\delta,\delta)\frac{\partial(Out_0^t, Out_1^t)}{\partial X^t} = (-\delta,\delta) \times \begin{bmatrix} \frac{\partial Out_0^t}{\partial x_1^t} & \cdots & \frac{\partial Out_0^t}{\partial x_m^t} \\ \frac{\partial Out_1^t}{\partial x_1^t} & \cdots & \frac{\partial Out_1^t}{\partial x_m^t} \end{bmatrix} =$$

$$\left[\delta\left(\frac{\partial Out_1^t}{\partial x_1^t} - \frac{\partial Out_0^t}{\partial x_1^t}\right) \quad \cdots \quad \delta\left(\frac{\partial Out_1^t}{\partial x_m^t} - \frac{\partial Out_0^t}{\partial x_m^t}\right)\right] = [\Delta x_1, \Delta x_2, \dots, \Delta x_m]$$

Therefore, $\Delta x_i = \frac{\partial \mathrm{L_2}}{x_i^t} = \delta\left(\frac{\partial Out_1^t}{\partial x_1^t} - \frac{\partial Out_0^t}{\partial x_1^t}\right) = \frac{2\delta(O_2-O_1)(1-O_1-O_2)(\sum_{k=1}^{K}\omega_{kl})H_k''(1-H_k'')(\sum_{j=1}^{J}\omega_{jk})H_j'(1-H_j')\omega_{ij}}{e^{O_1-O_2+2}+e^{O_2-O_1}}$

4. $\frac{\partial \mathrm{L_2}}{\partial X^t} = [\Delta x_1, \Delta x_2, \dots, \Delta x_m]$ is also a m-dimensional vector. If $|\Delta x_i|$ is the largest one in $\{\Delta x_1, \Delta x_2, \dots, \Delta x_m\}$, then x_i^t is abnormal attribute, which we can modify x_i^t and let $\mathrm{Out^t}$ approximate $\widehat{\mathrm{Out}}$ infinitely, i.e. push abnormal output to normal one.

For the adjusting stage, after detecting key anomalous attribute/characteristic $x_{q_i}^t$ of a record X^t (Here, we use $\{q_i\}_{i=1}^m$ to denote the descending order of subscript of $\{\Delta x_i\}_{i=1}^m$, which subject to $\Delta x_{q_1} \geq \Delta x_{q_2} \geq \cdots \geq \Delta x_{q_N}$), we can make some modification to $x_{q_i}^t$ to verify that the output of modified record $Out^M = F(x_{q_i}^t + \eta' \frac{\partial L_2}{x_{q_i}^t})$ until it becomes normal. The basic steps of this stage are as provided in the Data Correction (DC) Algorithm.

Anomaly Detection Algorithm (AD)

1. For a new anomalous record $X^t = [x_1^t, x_2^t, \ldots x_m^t]$, whose key abnormal attribute/characteristic $x_{q_i}^t$ has been detected in the Detecting step, we initialize the learning rate $\eta' = 0.001$ and $i = 1$;
2. Make a modification to $x_{q_i}^t$, i.e. $x_{q_1}^t = x_{q_1}^t + \eta' \Delta x_{q_1}^t$;
3. Calculate the output of the modified record $Out^M = F(x_{q_1}^t + \eta' \Delta x_{q_1}^t)$, if $Out_0^M > Out_1^M$, we acquire the corrected result i.e. the whole process is terminated; Otherwise, go to step 2 and continue to modify $x_{q_1}^t$ until $Out_0^M > Out_1^M$, and record the number of iteration N and the convergence value of Out^M by modifying $x_{q_1}^t$, if $N > L$, then turn to step 4 (L is a limit to the maximum number of iterations);
4. Let $i = i + 1$, i.e. make some modification to $x_{q_i}^t$, i.e. $x_{q_2}^t = x_{q_2}^t + \eta' \Delta x_{q_2}^t$, and then turn to step 3.

When we make some modification to $x_{q_i}^t$, it may not meet the termination condition $Out_0^M > Out_1^M$ after performing many times, so we set a limit to the maximum number of iterations L. Because there are more than one attribute are anomalous, it is not sufficient to adjust the largest one ,i.e. $\max\{|\Delta x_i|\} = \Delta x_{q_1}^t$. So when we modify the $x_{q_i}^t$ until converging at a steady value but still $Out_0^M < Out_1^M$, we start to modify the next attribute in the descending order of $\{\Delta x_i\}_{i=1}^m$ i.e. $\Delta x_{q_{(i+1)}}^t$ is only less than $\Delta x_{q_i}^t$.

4. Experimental Results

First, we train the neural network on the data set $\{X_i\}_{i=1}^{b=1,000,000}$ which has been labelled by "normal" and "abnormal" from customer data of a commercial bank. As the anomaly correction of the research methods is few, for the anomaly correction, we can take two ways as follows:

- Traditional method: we randomly selected 1000 anomalous data points (records) from a real data set collected from a Chinese commercial bank for the precision rate test, and then calculate the corresponding P:

$$p = \frac{number\ of\ anomalies\ corrected\ correctly}{number\ of\ \ anomalies\ corrected}$$

- Indirect method: Through the abnormal correction, then we use these revised data to do some other work, such as risk assessment in commercial bank.

Therefore, we randomly selected 1000 customer records data point from a real data set collected from a Chinese commercial bank for the recall rate test, and get a P value 78.3% . But if we have no baseline, we cannot evaluate our method. So we use the indirect method to do a reasonable evaluation. Here, we select a data set used for risk evaluation in commercial bank, and perform some traditional methods to do risk evaluation by using tools such as SVM [12], discriminant analysis [13], regression analysis [12,13], decision tree analysis [12]. Then, we perform our anomaly correction method based on neural network. After that, we use the revised data to perform the same methods for risk evaluation, finally we compare the P/R of these methods, as shown in Table 1. We can see that our method has a satisfactory improvement to the risk evaluation methods.

Table 1: Performance of SVM and other methods.

Data source	**Performance (Precision / Recall)**			
	SVM	**Discriminant analysis**	**Regression analysis**	**Decision tree analysis**
Revised data by our method	86% / 75%	81% /67%	82% / 70%	85% / 74%

Original data	80% / 65%	73% / 62%	78% / 69%	79% / 66%

Especially, we use a new way to test our data anomaly correction method. First we choose the data labelled by "normal" in the data set $\{X_i\}_{i=1}^{b=1,000,000}$ which has been labeled by "normal" and "abnormal". A new data set $\{X_i'\}_{i=1}^{b=500,000}$ is created.With the new dataset, we randomly choose 10,000 (i.e. 2% data points as data set $\{X_i''\}_{i=1}^{b=10,000}$) and modify the values of some attributes randomly and label it with "abnormal" to form another new data set $\{X_i'''\}_{i=1}^{b=10,000}$. Therefore, we get 2% data points which have been modified, and then mix them with the data set $S = \{X_i'\}_{i=1}^{b=500,000} - \{X_i''\}_{i=1}^{b=10,000}$ as dataset S'. Therefore, we have our training set S' and test set $\{X_i''\}_{i=1}^{b=10,000}$. After we have performed the DC algorithm, we compare $\{X_i''\}_{i=1}^{b=10,000}$ with the corrected results, and calculate the precision of our method.

As we know, the number of neurons of the input layer is 40 i.e. $m = 40$ (the dimension of the data point is 40) and the number of neurons of output layer is 2 (represent normal and abnormal). We have trained the neural network by setting different number of hidden layers, different number of neurons of hidden layers and learning rate η to get the best results. In this way, we get a P value 78.1% eventually when the number of hidden layers is 2 (the number of neurons of these two hidden layers is 300 and 1000), and get the fastest convergence when the learning rates is 0.005. The training accuracy increases with the number of neurons increasing, and after reaching a certain degree, the effect of increasing the number of neurons or the number of hidden layers is not obvious.

For the record ID=001158423463, we put the respective record to a trained network, and find the weight of the attribute value "2000" of the attribute "Income" is bigger, so we adjust the attribute value "2000" to a changed number with a setting learning rate 0.005 and put it into the trained network until the label of this record turn to normal when the income has reached "20000". Similarly, for the record ID=001158441549, we first adjust the attribute value "Manager" to a changed data because it has the largest gradient $\Delta x_{q_1}^t$ and put it into the trained network until the label of this record turn to normal, but it converges with "Assistant" and the label has not turned to normal. So then we adjust the attribute value "16" of the attribute "Age" to a changed number because it has the second largest gradient $\Delta x_{q_2}^t$ and put it into the trained network until the label of this record turn to normal when the age has reached "22". From here, we can see that if we have low-dimensional data which only has some attributes like "Age, Income, Position", then the information which can be used to help us to determine how to adjust the anomalous data is nothing much. If we adjust multiple attributes at the same time and then the anomalous records also can turn to be normal. But it will change the nature of the original data so that this kind of adjustment is not meaningful. So the way that appending the anomalous data record with customer portrait characteristics may provide more useful information for the training model.

Table 2: Example of the data anomaly correction results.

Customer ID	**Age**	**Position**	**Income**	**Label**
ID=001150059734	23	Assistant	4000	normal
ID=001158423454	35	Manager	9000	normal
Δx_i of X^t	1.132	1.457	1.291	
ID=001158441549	**16**	**Manager**	3000	abnormal
Δx_i of $X^t : x_{q_1}^t$		**-4.998**	1.445	
Δx_i of X^t: $x_{q_2}^t$	**3.774**			
Adjust and correct	***22***	***Assistant***	*3000*	***normal***
ID=001158423463	45	CEO	**2000**	abnormal
Δx_i of X^t	1.774	1.862	**-5.343**	
Adjust and correct	*45*	*CEO*	***20000***	***normal***

4. Conclusion

In our previous work, we have designed a fast clustering method for data anomaly detection. Based on the clustering results, we have described a novel method for data anomaly correction based on neural network. It has the detecting stage and adjustin g stage. The first stage finds the key attributes that have a great influence on the normality and abnormality of

the data, and then adjust the value of each anomaly until it becomes normal automatically with the second stage. We have the following observations after conducting the experiments:

- When there are more than one attributes be anomalous, it is not enough to adjust the largest one to meet the termination condition $\mathrm{Out}_0^{\mathrm{M}} > Out_1^{\mathrm{M}}$, multi-attributes adjustment may be a dynamic programming problem for error minimization, which may be better than adjusting in turn.
- In our later work, we need to consider adjusting multiple attributes and ensuring the reliability of the adjusted data records.

References

[1] L. Zhao, S. S. Yuan, Q. X. Yang, S. Peng, "Dynamic similarity for fields with null values," *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, pp. 161-169, 200.

[2] J. Grzymala-Busse, M. Hu, "A comparison of several approaches to missing attribute values in data mining," *In proceedings of the second international conference on rough sets and current trends in computing*, 2000

[3] T. Imieliński, W. Lipski Jr, "Incomplete information in relational databases," *Journal of the ACM (JACM*), vol. 31, no. 4, pp. 761-791, 1984.

[4] E. A. Gustavo, P. A. Batisa, M. C. Monard, "A study of k-nearest neighbour as an imputation method," 2003.

[5] R. Neal, "Probabilistic inference using Markov chain Monte Carlo methods," CRGTR -93-1, Department of Computer Science, University of Toronto, 1993.

[6] W. Fan, H. Lu, S. E. Madnick, D. Cheung, "Discovering and reconciling value conflicts for numerical data integration," Information systems, vol. 26, pp. 635-656b, 2001.

[7] A. Motro, P. Anokhin, A. C. Acar, "Utility-based resolution of data inconsistencies," *IQIS*, pp. 35-43, 2004.

[8] O. Benjielloun, A. Das Sarma, A. Halevy, "ULDBs:Databases with uncertainty and lingeage," VLDB, 2006.

[9] L. Yu, X. Yao, S. Wang, et al, "Credit risk evaluation using a weighted least squares SVM classifier with design of experiment for parameter selection," *J. Expert Systems with Applications*, vol. 38, no. 12, pp. 15392-15399, 2011.

[10] Z. Yan, "Analysis of the commercial bank non- performing loan problem based on statistical analysis methods," *Business*, vol. 12, p. 177, 2013. (in Chinese).

[11] C. Karen, K. Greenidge, W. Moore, D. Worrell, "Quantitative assessment of a financial system: Barbados," International Monetary Fund, Monetary and Financial Systems Department, pp. 5-76, 2005.

[12] M. Yu, Y. Yang, G. Wang, "Analysis of risk characteristics of bank personal non-performing loans*," Oriental Enterprise Culture*, vol. 10, pp. 45-46, 2010. (in Chinese).

[13] Q. Chi, G. Chen, "An empirical study of the decision tree technique in the analysis of non-performing bank personal loans," *Journal of Tonghua Teachers College*, 2010, vol. 4, pp. 52-53, 73, 2004. (in Chinese).