Fraud Detection for MCC Manipulation

Hong-feng CHAI¹, Xin LIU², Yan-jun XU¹, Shuo HE¹, Xia-xia LIN², Zhen LIU²,3,* and Jin MA²

¹China Unionpay Limited Liability Company, Shanghai 201201, China
²Web Sciences Center, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
³Big Data Center, University of Electronic Science and Technology of China, Chengdu 611731, China
*Corresponding author

Keywords: MCC, Fraud detection, Hierarchical clustering, Logistic regression.

Abstract. MCC manipulation arising in recent years is a new kind of fraud happened in financial system. Aiming at this issue, this study proposes a computational framework to detect such fraud behavior. The framework utilizes hierarchical clustering to discover the inherent transaction behavior pattern existed in various businesses and predicts the fraud merchant by applying logistic regression model. The experimental results indicate that the model overall outperforms other classifiers in detecting merchant's MCC manipulation.

Introduction

Nowadays, with the rapid economic development, POS machines (Point Of Sale information management system) have become popular for various kinds of businesses and been widely used by merchants in China. At the same time, the behaviors of using forged Merchant Category Code (MCC) for fee evasion by merchants continue to mount up in recent years. Such behavior of MCC manipulation is determined as a common fraud behavior for China UnionPay Inc., a financial and payment card company which is similar to VISA and MasterCard, and has caused the issuer huge loss. Current means to handle this issue are still in the phase of manual inspection. Therefore, some automatic detection models need to be studied in this field. According to the literature, there are three mainstream categories of methods for credit card fraud detection including neural network[1,2], Bayesian Network(BN) [3,4], and Decision Tree(DT) [5,6] . Some other methods for card fraud detection are also studied by researchers, such as support vector machine (SVM) algorithm[7], hidden Markov model (HMM) [8], etc.

Based on the analysis of UnionPay's transaction dataset together with previous studies on the issue of credit card fraud detection, we propose a computational framework for Merchant Category Code fraud detection. This framework originally combines the hierarchical clustering and logistic regression together and develops a payment behavior calculation model on merchant’s transaction dataset. Through this framework, we can discover the operating pattern of various businesses. Finally, experimental results show that logistic regression algorithm overall outperforms other classifiers in detecting merchant's MCC manipulation.

Data Features and Description

In this paper, the real dataset provided by the UnionPay contains very detailed transaction logs of merchants. Note that, few of the raw data has inaccurate information which needs to be cleaned. The example of the original transaction records is shown in table 1. Each data record includes 10 attributes which describe a merchant’s basic properties, transaction frequency and transaction amount, etc. The last column indicates whether a merchant is fraud marked by the business department of the
UnionPay. The merchant who is committed to be fraud is marked 1, 0 otherwise. Meanwhile, we selected transaction records related to eight typical MCC codes from the UnionPay's database, which have half amount of normal merchants and half amount of fraud merchants.

Table 1. The example of the raw transaction records.

<table>
<thead>
<tr>
<th>Merchant's ID</th>
<th>MCC</th>
<th>Pay rate</th>
<th>month</th>
<th>week</th>
<th>hour</th>
<th>Transaction frequency</th>
<th>Transaction amount</th>
<th>tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5411</td>
<td>0.0038</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>80</td>
<td>702430</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>5411</td>
<td>0.0038</td>
<td>12</td>
<td>7</td>
<td>23</td>
<td>1</td>
<td>31200</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>5722</td>
<td>0.0125</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

The Model of MCC Fraud Detection

We model the problem of MCC fraud detection as a quantitative judgment of fraud probability. From the application aspect, because there are only two identities of all merchants: fraud merchants and normal merchants. Therefore, we can map this problem to a typical binary classification problem. This fraud detection framework shown in Figure 1 includes three steps. Step 1 is a hierarchical clustering process to obtain the standard behavior pattern. Step 2 is to generate a classifier by using logistic regression model. Step 3 is to apply the classifier to detect normal users and fraud users. The more details are discussed as follows.

![Figure 1. The flowchart of MCC fraud detection model.](image)

By investigating the raw data, we found that different business categories have different business time intervals, operating disciplines and transaction peak, etc., which we call it “pattern of behavior”. We selected over 100 representative normal merchant data in each business category to investigate its corresponding pattern of behavior. Aiming to a given business category which is corresponding to a MCC, we calculates its transaction amount in accordance with hourly, weekly and monthly scales. To reflect the transaction difference between the neighboring time periods, we also consider to calculate
the first order derivative of the transaction amount in different time scales. By doing so, we will get 86
dimensional features in different time granularity.

As stated, due to partial raw data being inaccurate, we should check all the behavior vectors to get
rid of any noise data. We use hierarchical clustering to do this task. We found that the noise data are
not independently scattered in the data set. Instead, the noise data are commonly clustered together in
small clusters. Hierarchical clustering algorithm is an efficient way to find the noise points which are
different from the normal clusters as shown in Figure 2.

![Figure 2. Checking noise data via hierarchical clustering.](image)

The key steps for hierarchical clustering are described as follows:

1. Each point (an vector) is initially regarded as a cluster. If there are N points, we have N
   clusters; Pairwise Euclidean distances among the clusters are calculated and recorded in a
distance matrix.

2. By checking the distance matrix, we pick out two clusters who have the shortest distance and
   merged them as a new cluster. The number of the clusters is updated to N-1.

3. To updating the distance matrix, we calculate the distances between the new cluster and the
   remaining clusters through a formula as follows:

   \[
   D(A_{new}, C) = \frac{D(A, C) \times \#(A) + D(B, C) \times \#(B))}{\#(A) + \#(B)},
   \]

   where \(D(A, C)\) denotes the distance between cluster A and C, \(\#(A)\) denotes the number of points in
   the cluster A. \(A_{new}\) represents a the new cluster merged by cluster A and B.

   Repeat step 2) and step 3) until we obtain K clusters (K is the expected number of clusters we set in
   advance).

After the hierarchical clustering on all merchants’ behaviors data, we take the behavior data points
as noise when the points belong to clusters whose member size is less than 10% of the total number of
the data. We treat these data as noise because they do not frequently happen among merchants and
thereby can not reflect the unique pattern of a given business.

By hierarchical clustering and denoising, we can get a set of standard behavior patterns library for
each merchant category. Interestingly, through the test, we find that the number of behavior patterns
for each MCC will not exceed 4. The monthly, weekly and hourly behavior pattern library for eight
MCC are shown in Figure 3.
Figure 3. Monthly, weekly and hourly behavior pattern library.

Next, we are about to build the classification model. After some carefully comparisons, we prefer to choose the Logistic Regression model. The procedure of the model is briefly introduced as follows:

Firstly, we set a hypothesis classification function:

\[ h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}. \] (2)

Given the unknown parameter vector \( \theta(x_1, x_2, \ldots, x_{95}) \) (setting each component of the vector as 1 when we initialize it), we need to determine its values by utilizing some labeled merchant data to calculate. Such process is known as model training. To this end, we further set a cost function:

\[ J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)})]. \] (3)

\( h_\theta(x) \) represents the predicted result, \( y \) presents real result. The formula is to accumulate all samples’ difference between the predicted result and the real result \( y \). Through iterative computing, we can minimize the value of the cost function, which is to make sure its convergence.

By using so-called gradient descent method to adjust the parameter vector \( \theta \), the value of cost function \( J(\theta) \) will be set to converge. After setting an appropriate leaning rate \( \alpha \) (here the initial value set as 0.001), we update \( j \) from 1 to \( n \) (\( n=95 \)) synchronously:

\[ \theta_j = \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}; \] (4)

until the value of cost function is converged. We set the obtained parameter to the model and then the training of the classifier is done. For a given merchant which is denoted by a 95 dimensional vector, we use the model to calculate a predicted value. If the value exceeds a threshold, which is simply set as 0.5 in our study, we determine it as a fraud merchant, normal otherwise.

Experiment and Analysis

We applied some well-known algorithms for comparison including Naive Bayes, Multilayer Perceptron, RBF neural network, Sequential minimal optimization, J48 and Random Tree, which are commonly used for fraud detection. All the experiments are performed by an open source software program WEKA. Figure 4 shows accuracy and running time comparisons among different algorithms. According to the comprehensive comparisons, we can find that the model of logistic regression and
the model of SMO have almost the identical best accuracy results in F-Measure value yet logistic regression performs better than SMO in running-time. Therefore, we determine that logistic regression is the most suitable model for the fraud detection of MCC manipulation.

Figure 4. The accuracy and running time comparisons among seven algorithms through WEKA 10 fold cross-validation tests.

Conclusion
The merchant’s MCC manipulation is a new kind of financial fraud behavior compared with other traditional fraud issues such as credit card fraud. In this paper, we proposed a complete framework to solve this problem, including standard behavior patterns extraction via hierarchical clustering, extended features calculating, Logistic Regression based classifier modeling. The experimental results indicate that the fraud detection model based on the Logistic Regression has overall best performance on the dataset with 8 MCCs.

References