A Case Study in Credit Fraud Detection With SMOTE and XGBoost

Cuizhu Meng¹, Li Zhou¹ and Bisong Liu¹

¹China national institute of standardization, Beijing, China

Abstract. Credit fraud observations are minority in the sample set, variables tend to be seriously unbalanced, and the prediction results tend to be biased towards more observed classes. Common resolution usually constructs 1:1 data, either cutting off part of more classes (undersampling) or reducing classes for bootstrap sampling (oversampling). XGBoost is an efficient system implementation of Gradient Boosting, and also GB algorithm based on CART. Based on the real online transaction data of an Internet financial institution, this paper studies the performance of XGBoost algorithm on the original data set, the undersampling and SMOTE data sets respectively.

1. Introduction

With the development of big data and e-commerce, Internet finance has grown rapidly, with transactions approaching $7 trillion in 2019. Take Yuebao as an example, the Yuebao online 18 days, the cumulative number of users reached more than 2.5 million; the cumulative transfer funds reached 6.6 billion Yuan.

Table 1. The Transaction Scale of China Internet Finance at 2011-2019

In the business scenario of Internet Finance, the risk of external credit fraud is more serious than that of offline business due to the following characteristics of the Internet.

- Concealment. In Internet finance, communication no longer face-to-face, instead automate the vast majority of business by exchanging data with servers over the network. These services no longer require service providers to interact face-to-face with customers, and the Internet communication protocol itself is anonymous. This gives fraudsters the convenience of disguising their identities, ‘On the Internet, no one knows you’re a dog.’ Fraudsters can use simple technical means to complete age, gender, identity and other disguises, through the network to carry out a variety of fraudulent activities.
- Customer selection. Such as online micro-loans and other mutual gold products, its customer base itself is a lack of adequate credit history, cannot be credited through banks and other channels. Such as migrant workers, students and other groups. Many of them are not high credit awareness, there is a small cheap psychology, and motivation to use various means to cheat credit.
• Defects in the product itself. Under the pressure of rapid market grabbing, each company has urged the development department to launch online products as soon as possible. Under the pressure of project progress, product design implementation process is prone to all kinds of omissions, or practitioners lack of experience wind control consideration, some products even relax the requirements of wind control under business pressure, these phenomena have increased operational risk. In addition, the lack of basic IT operations and maintenance capabilities is also easy to expose sensitive data and business processes to hackers.

2. Credit Fraud Detection

2.1. Existing Credit Fraud Detection Methods
This subsection concentrates on the analysis of some reliable data mining methods applied specifically to the data-rich areas of insurance, credit card, and telecommunications credit fraud detection, in order to integrate some of them. A brief description of each method and its applications is given.

The Bayesian Belief Network (BBN) and Artificial Neural Network (ANN) comparison study uses the STAGE algorithm for BBNs and BP algorithm for ANNs in credit fraud detection. Comparative results show that BBNs were more accurate and much faster to train, but BBNs are slower when applied to new instances, as in equation [1]. Real world credit card data was used but the number of instances is unknown. The distributed data mining model is a scalable, supervised black box approach that uses a realistic cost model to evaluate C4.5, CART, Ripper and NB classification models. The results demonstrated that partitioning a large data set into smaller subsets to generate classifiers using different algorithms, experimenting with credit fraud, legal distributions within training data and using stacking to combine multiple models significantly improves cost savings. This method was applied to one million credit card transactions from two major US banks, Chase Bank and First Union Bank. FairIsaac, formerly known as HNC, produces software for detecting credit card fraud. It favors a three-layer BP neural network for processing transactional, cardholder, and merchant data to detect fraudulent activity.

2.2. The New Credit Fraud Detection Method
First, the Internet platform can use its own user data, environmental data, and behavior data, combined with bioprobe technology collected by the user behavior data, to establish users, environment, and behavior portraits and based on the user, environment, and behavior of the relationship network, as in equation [2]. The recognition of cheating behavior is done by using machine learning algorithm to identify abnormal points in the relationship network in real time.

Secondly, according to the different types of cheating and different risk levels, through password, up and down text messages, voice, graphics verification code and other means to verify user behavior with real-time risk decision-making engine. At the same time, Internet companies can also use their own data and technical advantages, the development of biometrics, questionnaires and other breakthrough difficult identification means, to further ensure timely suppression of user cheating.

Finally, the Internet platform takes advantage of its own big data, can all the historical identified cheating classification storage, and the correlation spread out more abnormal user, environment, behavior data. Use this data through blacklists and comparisons with existing behaviors to increase the recognition of cheating and the difficulty of cheating.

2.3. Class Imbalance
Credit fraud monitoring is a typical class imbalance problem in machine learning because the credit fraud monitoring fraud sample percentage is small in the real world. Class-imbalance refers to the uneven distribution of the training sets used in the training classifier. For example, a two-classification problem, 1000 training samples, the more ideal situation is the number of positive and negative class samples is similar, and if the positive class samples have 995, negative class samples only 5, it means that there is class imbalance, as in equation [3].
From the training process of the model, from the point of view of the training model, if the sample size of a class is very small, then the "information" provided by this category is too little, the model did not learn how to distinguish a few classes. Consider extremes: of the 1000 training samples, 999 were positive and 1 was negative. At the end of a certain iteration during training, the model divided all the samples into positive classes, and although the negative category was misclassified, the damage was negligible, with accuracy already 99.9%.

3. The DATA
The aim of this experiment is to detect fraudulent activity in a real-world dataset of credit card transactions. The dataset contains 284,807 transactions, only 492 of which are labeled as fraudulent, that is 0.172%. Due to confidentiality considerations, this dataset has been transformed from its original form and only provides time and amount of each transaction along with 28 principal components of the original features obtained by PCA. Some information is inevitably lost during this transformation which limits how well any algorithm can do on this dataset compared to similar non-transformed datasets. It also renders feature engineering virtually irrelevant, as I discuss below in more detail.

Table 2. Original Data distribution

3.1. Check the Spread of Fraud vs non-fraud on Selected Variables
This is done identify variables selection which has good spread of frauds and non-frauds.

Table 3. The Spread of Fraud vs non-fraud on Selected Variables

3.2. Data pre-processing
To address the extreme class imbalance in the data, I will use two different preprocessing techniques. For the XGBoost algorithm I simply reweight the instances of the positive class (fraudulent transactions) by the class imbalance ratio. In the case of Logistic regression (one of the algorithms
to which I will compare XGBoost), I use the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE (Chawla et al. 2002) balances the class distribution by creating new synthetic instances of the minority class, as in equation [4].

Standardize the Amount feature, and here's the post-processing data:

| V9  | V10 | ... | V21 | V22 | V23 | V24 | V25 | V26 | V27 | V28 | Class | normAmount |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------|------------|
| 95378 | 0.909794 | ... | 0.191807 | 0.277638 | -0.110474 | 0.069978 | 0.126593 | -0.191119 | 0.133558 | -0.021753 | 0 | 0.244964 |
| 355245 | -0.190974 | ... | -0.225775 | -0.038672 | 0.101288 | -0.339646 | 0.161707 | 0.125895 | -0.008983 | 0.014714 | 0 | 0.342475 |
| 514654 | 0.207643 | ... | 0.247696 | 0.771679 | 0.906442 | -0.09281 | 0.220264 | -0.119697 | 0.055353 | -0.069752 | 0 | 1.106086 |
| 387024 | -0.054652 | ... | -0.108300 | 0.005274 | -0.190321 | -1.377575 | 0.647376 | -0.221929 | 0.962273 | 0.061456 | 0 | 0.140534 |
| 317739 | 0.750074 | ... | 0.009431 | 0.798278 | -0.137458 | 0.141207 | -0.200206 | 0.502282 | 0.218442 | 0.215153 | 0 | -0.073403 |

Table 4. Post-processing Data

However, I do not perform any further feature engineering (beyond rescaling for Logistic regression) for the following reasons:

- There are no missing values in this dataset and hence no need for imputing missing values. All variables are continuous numerical values.
- XGBoost is an ensemble learning algorithm whose individual learning units are decision trees and trees have two favourable features which, again, render feature engineering unnecessary. First, decision trees are invariant to monotonic transformations of features (e.g. scaling or polynomial transformations). Second, they can inherently capture and model interactions between features. So, we do not need to manually create feature interactions.

As I mentioned above, the PCA transformation makes it impossible to use our background knowledge about the features to create new ones, as in equation [5]. Moreover, we do not need to worry about feature correlation as principal components are, by construction, orthogonal and therefore uncorrelated with one another.

4. Experiment

Because the positive and negative samples are extremely uneven, as a comparison, this paper first models XGBoost directly on the native dataset and evaluates the model results using recall and ROC curves. Secondly, this paper uses the methods of subsampling and oversampling, modeling and evaluating the results respectively.

4.1. Direct modelling assessment

In the native data set, the number of positive samples was 284315, and the number of negative samples was 492. Based on the original data set, modeling using the XGBoost algorithm, the recall of the model is 0.80625, AUC is 0.97954.

4.2. Undersampling

Undersampling is an undersampling of a large number of samples (most classes) in the training set, discarding some samples to alleviate class imbalances. This paper is modeled by undersampling and modeled by the XGBoost algorithm, with the recall of the model at 0.92258 and the AUC at 0.98142.
4.3. SMOTE Oversampling

SMOTE is an oversampling algorithm proposed by JAIR in his 2002 article "SMOTE: Synthetic Over-Sampling Technique". In summary, this algorithm synthesizes new samples for a few classes based on interpolation.

If the number of samples for a minority class with a training set is \( T \), the SMOTE algorithm will synthesize a new \( NT \) sample for that minority class. The requirement here is that \( N \) must be a positive integer, and if given \( N<1 \) then the algorithm will ‘think’ the number of samples of a few classes, \( T=NT \), and forcedly set \( N=1 \), as in equation [6].

Consider a sample \( i \) for this minority class with a characteristic vector of \( x_i, i \in \{1,...,T\} \).

- The \( k \) neighbours of sample \( x_i \) (e.g. Euclidean Distance) were first found in all \( t \) samples of this minority class, which are recorded as \( x_i(near) \), \( near \in \{1,...,k\} \).

- Then randomly select a sample \( x_{i(nn)} \) from this \( k \) neighbour, and then generate a random number between 0 and 1, resulting in a new sample \( x_{i1} \):

\[
x_{i1} = x_i + \zeta_1 \cdot (x_{i(nn)} - x_i)
\]

(1)

- Repeat Step 2 \( N \) times so that \( n \) new samples can be synthesized: \( x_{i\text{new}},\, \text{new} \in \{1,...,N\} \)

Over-sampling was sampled using the SMOTE algorithm and fraud monitoring was carried out by XGBoost algorithm, with a model recall of 0.9 and AUC is 0.98533.

| Table 6. Data Distribution And AUC curve of Undersampling |
|---|

5. Conclusion

This paper used the XGBoost algorithm to detect fraudulent credit card transactions in a highly imbalance dataset where credit fraud examples constitute only 0.172% of examples. Given the importance of detecting as many instances of credit fraud as possible in this application, we evaluated models based on their recall and ROC curve at a given level of precision. Based on the native data set, the undersampling data set, the oversampling data set, the XGBoost algorithm is modeled, and the results are compared to the AUC value by recall and ROC curve, and it is necessary to use SMOTE algorithm to achieve the balance of data. By balancing the data, a more stable and generalized model can be achieved.

References

[1] Alejandro Correa Bahnsen, Aleksandar Stojanovic, Djamila Aouada, and Bj"orn Ottersten. Cost
sensitive credit card fraud detection using bayes minimum risk. In 2013 12th international conference on machine learning and applications, volume 1, pages 333–338. IEEE, 2013.

[2] Adrian W Bowman and Adelchi Azzalini. *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*, volume 18. OUP Oxford, 1997.

[3] Wojtek J Krzanowski and David J Hand. *ROC curves for continuous data*. Chapman and Hall/CRC, 2009.

[4] Giovanna Menardi and Nicola Torelli. Rose: random over-sampling examples. *Data Mining and Knowledge Discovery*, 28(1):92–122, 2014.

[5] John A Swets. *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers*. Psychology Press, 2014.

[6] Marina Valdora and V´ictor J Yohai. Robust estimators for generalized linear models. *Journal of Statistical Planning and Inference*, 146:31–48, 2014.