Credit Card Fraud Detection

Course project for EE 401: Pattern Recognition and Machine Learning, Autumn Semester 2019-20

K. Sai Anuroop, Mandeep Bawa, Sushma Biradar, Aniruddha Joshi *

Computer Science and Engineering, IIT Dharwad

Faculty Supervisor: Prof. S.R.M. Prasanna

Report

December 7, 2019

Abstract

Credit card fraud can be defined as, *'Unauthorized account activity involving a payment card, by a person for which the account is not intended'*.
These frauds cost consumers and banks millions of dollars worldwide, as a response to which several modern fraud-detection techniques are in place today.

1 Introduction

We based our work on the dataset provided in Kaggle [1] under Open Database license. This dataset is already normalized but is highly biased towards Non-Fraud transactions, as is expected of a legal nature of transactions across the world. Feeding this raw data to our algorithm will highly affect its results. So, data pre-processing is a must to ensure correctness and validity of results. For addressing this issue, we have implemented the bagging phase. Since we are concerned with classification of feature vectors, we implemented SVM with Gaussian and Polynomial kernel functions. Also, we have explored two different strategies for training our classsifer.

2 Literature Reviewed

We reviewed the following articles:

*Email IDs of team members in order: [170030035, 170030038, 170010032, 170020004] @iitdh.ac.in

- **Parallel and incremental credit card fraud detection model to handle concept drift and data imbalance** [2]
  This paper proposes a transaction window bagging (TWB) model, a parallel and incremental learning ensemble, as a solution to handle the issues in credit card transaction data. TWB model uses a parallelized bagging approach, incorporated with an incremental learning model, cost-sensitive base learner and a weighted voting-based combiner to effectively handle concept drift and data imbalance.

- **Fraud Detection using Machine Learning** [3]
  An effective fraud detection system should be able to detect fraudulent transactions with high accuracy and efficiency. While it is necessary to prevent bad actors from executing fraudulent transactions, it is also very critical to ensure genuine users are not prevented from accessing the payments system. A large number of false positives may translate into bad customer experience and may lead customers to take their business elsewhere.
  Designing an accurate and efficient fraud detection system that is low on false positives but detects fraudulent activity effectively is a significant challenge for researchers.

3 Algorithm

We decompose the proposed algorithm into three parts:

- Bagging of feature vectors
- Assigning weights to the classifier associated with every bag
- Obtaining the model for classification
- Testing the dataset with obtained classifier

We explain each of the above parts in detail in the following sub-sections.

3.1 Bagging of feature vectors

Random selections tend to retain data proportions, hence the constituent imbalance levels in training data are carried forward to the base learners. This leads to data imbalance affecting the training process to a large extent. The proposed model enables balanced data selection such that the effects of data imbalance are considerably reduced during model training.
Let, T_{min} to be the set of *Fraudulent Transactions* and T_{major} to be *Legitimate Transactions*.
TB_b be the b^{th} bag containing feature vectors on which base learner is trained.
Then define,

$TB_b = T'_{major} \cup T_{min}$
where T'_{major} defines sampled instances of T_{major}, and is created by performing n overlapping divisions of T_{major}.
Instances for T'_{major} are obtained by sampling the data from T_{major} within the interval $[(b-1)NT + 1, (b)NT]$
where $1 \leq b \leq n$ is the bag identifier,
$NT = |T'_{major}| = \frac{|T_{major}|}{n} + \theta|T_{major}|$,
$0 \leq \theta \leq 1$ is a hyper-parameter that defines the degree of accepted overlap among majority classes.
Consecutive bags contain certain levels of overlaps to make sure that the temporal distribution change is gradual and hence does not exhibit sudden changes in predictions between consecutive bags.

3.2 Assigning weights to false positives and false negatives in the cost function

Let the *Hypothesis Function* be $H(x)$ where x is the input vector and $H(x)$ gives us the probability of the input vector belonging to the positive class. Then the *Cost Function (i.e. cost for an input vector)* $c(x^{(i)})$ may be written as:
$c(x^{(i)}) := y^{(i)}(\alpha_1 k_1(H(x^{(i)}))) + (1 - y^{(i)})(\alpha_2 k_2(H(x^{(i)})))$
where k_1 and k_2 are functions such that:

$$k_1(p) = \begin{cases} 1, & \text{if } p < T \\ 0, & \text{otherwise} \end{cases}$$

$$k_2(p) = \begin{cases} 1, & \text{if } p > T \\ 0, & \text{otherwise} \end{cases}$$

T is the threshold of our prediction based on the model, i.e., we predict positive class if $H(x) > T$ and predict negative class otherwise
α_1 is the weight given to False Negative
α_2 is the weight given to False Positive.
Final cost function i.e the *Cost* can be written as:
$Cost := \sum_{i=1}^{M} c(x^{(i)})D + \frac{(w^{(i)})^2}{2}$
In our proposed approach, we may penalize mistakes made in classifying positive classes more than the mistakes made in classifying negative class or vice-versa. Threshold T may be defined on the basis of ROC or PRC curves. We may choose to plot PRC over ROC as PRC is more sensitive to mis-classifications when dealing with highly imbalanced dataset like ours.

3.3 Defining the model for classification

We propose to use SVM with Gaussian kernel function for classification of transactions. The transformed cost function $Cost$ may be written as:

$Cost = \sum_{i=1}^{M} c(f^{(i)})D + \frac{(w^{(i)})^2}{2}$

where $f^{(i)}$ is the transformed feature vector obtained by applying the Gaussian kernel function to $x^{(i)}$.

4 Modelling Scheme

4.1 Bagging Phase

Bagging phase is carried-out as explained in 3.1.

4.2 Training Phase

4.2.1 Strategy #1

We vary the values of weights given to FPs and FNs to find the optimal values of these at which a bag gives maximum sum of accuracy, precision and recall. We do this for all the bags and obtain the classifier of the bag at the optimal value. Next, we assign weights to the classifiers obtained, on the basis of the sum of accuracy, precision and recall at their optimal FP and FN weights. We then test the pooled and entire datasets with all these classifiers, by weighing the classification output according to the weight of the classifier and summing up weighted outputs to find the final output of classification. If it is less than 0.5, we call it non-fraud and fraud otherwise.

4.2.2 Strategy #2

We vary the values of weights given to FPs and FNs to find the optimal values of these at which the sum of accuracy, precision and recall of all the bags is maximum. Next, at the optimal value of weights for FPs and FNs obtained, we train the classifiers of every bag and find that classifier which gives the maximum sum of accuracy, precision and recall. We then use this classifier to test on the entire dataset. Polynomial kernel performed better than Gaussian kernel.

4.3 Testing Phase

We merged data used for testing different bags to create one big pooled testing dataset. This was achieved by pooling 20% of the testing dataset of each bag. Also, we tested on the entire dataset.

5 Results

5.1 Strategy #1

Pooled Dataset
Accuracy: 97.32%
Precision: 97.93%
Recall: 96.19%
Entire Dataset
Accuracy: 97.33%
Precision: 97.93%
Recall: 96.14%

5.2 Strategy #2

Pooled Dataset
Accuracy: 87.18%
Precision: 81.20%
Recall: 93.36%
Entire Dataset
Accuracy: 99.92%
Precision: 87.34%
Recall: 69.69%

6 Conclusion

We reviewed different algorithms for credit card fraud detection starting with logistic regression, SVM and then came up with our own algorithm, fusing different techniques employed in the literature reviewed.
In our tests, Strategy #1 performed better on the pooled dataset when compared to Strategy #2. Also, it was noticed that Strategy #2 gave better performance when tested over entire dataset.
It is observed that adding weights based on false positives and false negatives to SVM classifier improves performance.
While training, it is observed that the method of bagging enabled balanced data selection such that the effects of data imbalance are considerably reduced during model training.
We thus proposed and implemented an algorithm for credit card fraud detection which yields good classification results.

7 Future Work

For real-time fraud detection, bags containing recent transactions could be given more weight when compared to bags containing older transactions.

8 Acknowledgements

We would like to thank Prof. SRM Prasanna for his guidance throughout the project.

References

[1] https://www.kaggle.com/mlg-ulb/creditcardfraud

[2] **Parallel and incremental credit card fraud detection model to handle concept drift and data imbalance**
Somasundaram, A. & Reddy, S.
Neural Computing and Applications, Springer, (2019) 31(Suppl 1): 3
https://doi.org/10.1007/s00521-018-3633-8

[3] **Fraud Detection using Machine Learning**
Aditya Oza, Stanford University
http://cs229.stanford.edu/proj2018/report/261.pdf