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 2

October 14, 2019

Abstract

In this report, we brief about the algorithm we propose to implement for detecting fraudulent transactions.

1 Literature Reviewed

We reviewed the following articles:

- **Parallel and incremental credit card fraud detection model to handle concept drift and data imbalance**
  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. [1]

- **Fraud Detection using Machine Learning**
  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.

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

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.
[2]

2 Proposed Algorithm

We decompose the proposed algorithm into three parts:

- Bagging of feature vectors
- Assigning weights to false positives and false negatives in the cost function
- Defining the model for classification

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

2.1 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.

2.2 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.

2.3 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 $x^{(i)}$.

References

[1] **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`

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