Credit Fraud Detection System based on Neural Network

Rakendra Thapa – rakendra.thapa@infineon.com

Introduction

The expansion of electronic commerce, together with increasing confidence of customers in electronic payment, effective Fraud Detection Systems (FDS) are increasingly becoming a critical factor.
The Fraud Detection we discuss in this paper is the process of identifying if an authorized credit card transaction belongs to the class of fraudulent or genuine transaction.
To model a FDS, one of the major challenge faced is the Class Imbalance present in the transaction dataset, i.e. Genuine Transactions far outnumber frauds. We evaluate different learning strategies and compare their performance based on evaluation metrics F1-Score, ROC and AUC.

Data Preprocessing

Data Characteristics

We use the dataset (http://www.ulb.ac.be/di/map/adalpozz/data/creditcard.Rdata). shared in public domain containing information about credit card transaction with examples of fraudulent samples .

1) Contained 32 numerical input variable and 284801 transaction samples.
2) Contained binary, and numeric data.
3) Dataset is highly unbalanced with fraud representing 0.172% of all transaction (492 frauds out of 284807 transactions).
4) Data was highly inseparable/overlapping

Data Preprocessing

1) Z-score Normalizing of the data.
2) Undersampling based balancing.
3) Correct the artificial bias introduced by undersampling by calibrating the Posteriors Probability of the model learned on balanced subset.
4) Using Bayes Minimum Risk theory to correct classification threshold and adjust after undersampling.

Algorithm Selection

Learning Strategy based on 3 layer Neural net.
We performing a Cross Validation for different frequencies of undersampling calibrating the Posterior Probability and the Threshold adjustment.
Comparing the performance with the Posterior Probability Calibration.

Let β = Probability of selecting a negative instance with undersampling.
P = Bias-corrected probablity obtained from P_s
$P = \beta P_s / (\beta P_s - P_s + 1)$.

Diving the Samples into:
- Training Set (60%, $N^- = 170590$, $N^+ = 294$)
- Testing Set (40%, $N_t^- = 113725$, $N_t^+ = 198$)

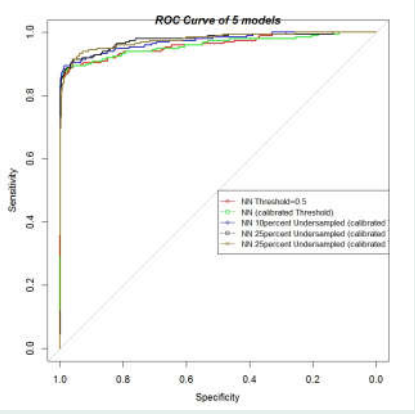
Setting β = 1(not sampled), 0.015, 0.00516, 0.00172(completely balanced)

Adjusting Threshold, T – cost of missing a positive instance N_t^+(false negative) >> cost of missing a negative N_t^-(false positive))

Final Result

Based on of AUC, we achieved significant improvement with Probablility Calibration after undersampling and Threshold adjustment.

Performance Evaluation



β	1	0.015	0.00516	0.00172	NN with no calibrated Probability and Threshold Adjustment
False Negative (FN)	32	21	17	12	41
False Positive(FP)	134	3113	5220	9415	18
True Negative (TN)	113591	110612	108505	104310	113711
True Positive (TP)	166	177	181	186	153
Accuracy	0.9985428755	0.9724901907	0.9540303538	0.9172511258	0.99948210
Classification Error Rate	0.001457124549	0.02750980926	0.04596964616	0.08274887424	0.000517893665
Precision	0.5533333333	0.0537993921	0.03351231253	0.01937298198	0.8947368421
Sensitivity (TPR)	0.8383838384	0.8939393939	0.9141414141	0.9393939394	0.7886597938
Specificity (TNR)	0.9988217191	0.972626951	0.9540998022	0.9172125742	0.999641729
AUC	0.9494208	0.9726553	0.9764814	0.976139	0.9494208

Future Work

1. Run Esemble methods, like RUSBoost, which improves classification performance when training data is skewed.
2. Incorporate following features:

- *Concept Drift* - Transactions might change their statistical properties over time.
- *Verification Latency* - The way and timing with which supervised information is provided by the expert investigators.