Combating Fraud By Leveraging Machine Learning

A CSCI Senior Thesis by Casey Astiz '19
Advised by Professor Michael Linderman
Fall 2018

Abstract

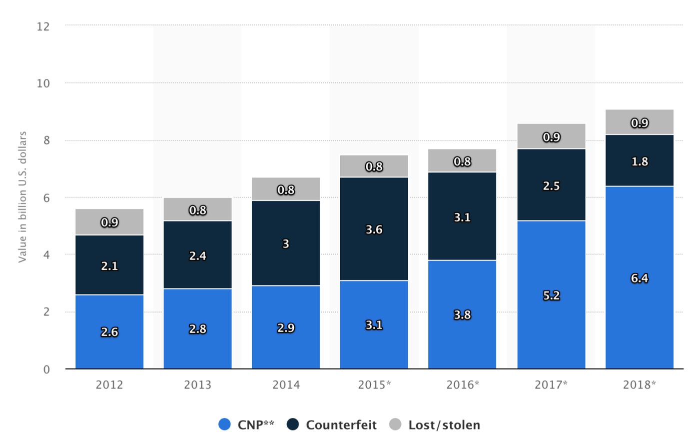
Credit card fraud in the United States has been on the rise, reaching over $8 billion in 2018 [1]. The majority of these are card not present transactions, which have become more popular with the increasing options for online transactions. While fraud transactions are typically larger than the average transaction size, they are not frequent, as they only make up around 1-3% of transactions. Therefore, it is difficult to create unbiased classifiers for fraud detection systems because there are few examples to train on. In this thesis, I summarize related works on the topic of anomaly detection, feature engineering for fraud, and game theory. I interview practitioners in the fraud detection space to compare their knowledge and day-to-day methods with current research methods. I implement my own fraud detection system using a synthetic mobile payment transaction dataset. From this dataset, I find decision trees to be the most effective classifier, which is consistent to the practitioners' use of rule based systems.

Introduction

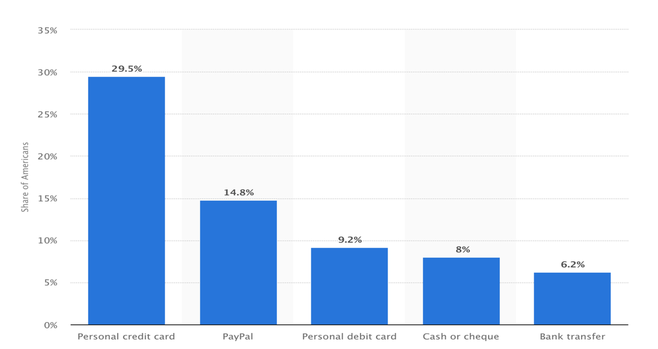
Fraud rates in United States are rising as new payment methods are introduced [1, 2].

Instances of fraud are relatively rare, making up only 1-3% of transactions. Therefore, fraud is an anomaly.

Difficult to build unbiased classifiers when there is a large class imbalance.



Leading payment methods in the United States in 2017



Related Work

Skewed Classes

- Artificially "unskew" your training data so a classifier sees more fraudulent examples.
- 50/50 training data distribution is best [3]
- Measure success of model based on Precision, Recall, and F1

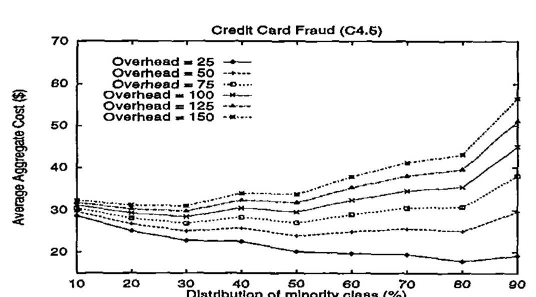


Figure 1: Training distribution vs. the credit card fraud cost model

Features Engineering

- Deep Feature Synthesis to automatically create based on historical behavior [6]
- Parenclitic Networks to measure relationships between features [7]

Game Theory

- Incorporate human intuition and experience with fraud into rules
- Predict next move of fraudster based on known behavior [4]

Methods

Data Paysim: synthetic mobile payments with 6 million non-fraud, 8000 fraud [5]

Summary Statistics

Transaction Measure	Mean	Min	Max
Legitimate Transaction			
Amount	178,197	0.01	92,445,516
Count = 6,354,407			
Fraudulent Transaction			
Amount	1,467,967	0	10,000,000
Count = 8,213			
Cash In (Deposit)			
Amount	168,920	0.04	1,915,267
Count = 1,399,284			
Cash Out (Withdrawal)			
Amount	176,273	0	10,000,000
Count = 2,237,500			
Debit			
Amount	5,483	0.55	569,077
Count = 41,432	17.54		
Payment			
Amount	13,057	0.02	238,637
Count = 2,151,495			
Transfer			
Amount	910,647	2.60	92,445,516
Count = 532,909			

Experiments

Training Data Distributions	Models
Original Dataset	SVM
50/50	Neural Network
66/33	Decision Tree
75/25	Logistic Regression
80/20	Gaussian

Data Process

Example for a 50/50 data distribution. All data combined with random sampling.

6 M Non-Fraud

8 K Fraud

50% Non-Fraud

50% Fraud

6/7 unskewed data: Training Data

1/7 unskewed data + rest of Non-Fraud: Testing Data

Results

Non-normalized Data Results

Method	Accuracy	Precision	Recall	F1 Score
Original data distribution				
Logistic Regression	0.999	0.3735	0.493	0.425
Neural Network	0.999	0	0	0
Decision Tree	0.9997	0.8974	0.903	0.899
SVM	0.004	1	.0013	0.003
Gaussian	0.952	0.532	0.015	0.029
50/50 data distribution				
Logistic Regression	0.910	0.002	0.902	0.004
Neural Network	0.0002	0.0002	1	0.0004
Decision Tree	0.989	0.017	0.995	0.033
SVM	0.002	0.0002	1	0.0004
Gaussian	0.491	0.0003	0.856	0.0006
66.6/33.3 data distribution				
Logistic Regression	0.910	0.854	0.885	0.869
Neural Network	0.999	0	0	0
Decision Tree	0.993	0.029	0.994	0.055
SVM	0.002	0.0002	1	0.0004
Gaussian	0.509	0.0003	0.857	0.0007
75/25 data distribution				
Logistic Regression	0.956	0.004	0.841	0.007
Neural Network	0.999	0	0	0
Decision Tree	0.995	0.036	0.997	0.070
SVM	0.002	0.0002	1	0.0004
Gaussian	0.505	0.0003	0.882	0.0007
80/20 data distribution				
Logistic Regression	0.971	0.005	0.809	0.011
Neural Network	0.999	0	0	0
Decision Tree	0.996	0.048	0.981	0.091
SVM	0.002	0.0002	1	0.0004
Gaussian	0.738	0.0005	0.617	0.001

Normalized Data Results

Method	Accuracy	Precision	Recall	F1 Score
Original data distribution				
Logistic Regression	0.999	0	0	0
Neural Network	0.999	0	0	0
Decision Tree	0.999	0	0	0
SVM	0.999	0	0	0
Gaussian	0.999	0	0	0
50/50 data distribution				
Logistic Regression	0.501	0	0	0
Neural Network	0.501	0	0	0
Decision Tree	0.499	0.499	1	0.666
SVM	0.534	0.536	0.499	0.516
Gaussian	0.499	0.499	1	0.666
66.6/33.3 data distribution				
Logistic Regression	0.670	0	0	0
Neural Network	0.670	0	0	0
Decision Tree	0.330	0.330	1	0.497
SVM	0.330	0.330	1	0.497
Gaussian	0.330	0.330	1	0.497
75/25				
Logistic Regression	0.750	0	0	0
Neural Network	0.750	0	0	0
Decision Tree	0.250	0.250	1	0.400
SVM	0.250	0.250	1	0.400
Gaussian	0.250	0.250	1	0.400
80/20				
Logistic Regression	0.801	0	0	0
Neural Network	0.801	0	0	0
Decision Tree	0.199	0.199	1	0.332
SVM	0.801	0	0	0
Gaussian	0.167	0.120	0.500	0.192

- Decision Trees had highest F1 score
- SVMs had lowest F1 score
- Normalizing the data led to worse performance than non-normalized data.
- Unskewing the data increases Total Positive Rate at cost of False Positive Rate

Table 4.5: Results: Non-normalized Data False Positive Rates by Model

Data	Logistic Regression	Gaussian	SVM	Neural Network	Decision Tree
50/50	0.090	0.509	0.995	1	0.011
60/33	0.057	0.491	0.998	0	0.007
75/25	0.044	0.495	0.998	0	0.005
80/20	0.029	0.262	0.998	0	0.004

Interviews

Interviewed 4 practitioners in fraud analytics. Example insights:

1) I hypothesized that fraud detection was more advanced in business than in research.

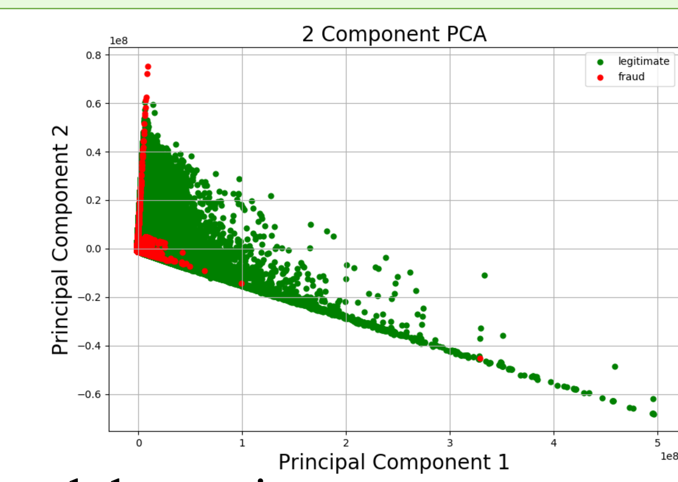
However, most businesses take simpler approaches than most recent research, and use manually created rule based systems or outsource their fraud detection.

2) Fraudsters use free trials, e.g. from Netflix or Spotify, to test out illicitly bought credit card information, then spending money on those cards that pass.

Discussion

Available Data Was Very Limited

- Few repeated users
- No location information
- No address information
- Very small proportion of fraud



Results

- Overlap between fraud and non-fraud data points.
- Larger fraud training proportions increases sensitivity, leading to higher false positive rates (FPR). Tradeoffs depend on cost of fraud versus cost of investigation and lost business costs.
- Small FPR increase of 0.03 leads to 180,000 more false positives in 6M transactions

Conclusion

Decision Tree Classifiers have the best performance on this dataset. This is consistent with interviews, as they are similar to rule based systems.

Training data distributions can alter the sensitivity of a classifier to detecting the minority class.

As future work, would like to try combining a rule based system with a classifier to filter out highly likely licit transactions in first phase.

References

[1] Most popular payment methods in the U.S. 2017 — Statistic.
[2] U.S. payment card fraud losses by type 2018 — Statistic.
[3] Philip K Chan and Salvatore J Stolfo. Toward Scalable Learning with Non-uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. Technical report.
[4] Suvasini Panigrahi, Amlan Kundu, Shamik Sural, and A.K. Majumdar. Credit card fraud detection: A fusion approach using Dempster Shafer theory and Bayesian learning. Information Fusion, 10(4):354–363, Oct 2009.
[5] Paysim. Synthetic Financial Datasets For Fraud Detection — Kaggle.
[6] Roy Wedge, James Max Kanter, Kalyan Veeramachaneni, Santiago Moral Rubio, Sergio Iglesias Perez, Banco Bilbao, Vizcaya Argentaria, and Spain Madrid. Solving the "false positives" problem in fraud prediction Automated Data Science at an Industrial Scale. Technical report.
[7] Massimiliano Zanin, Miguel Romance, Santiago Moral, and Regino Criado. Credit Card Fraud Detection through Parenclitic Network Analysis. Complexity, 2018:1– 9, may 2018.