Comparing Graph Features for Automated Credit Card Fraud Detection

Michael Holtz

December 1, 2023

1 Abstract

Credit card fraud detection is a binary classification problem that seeks to identify transactions as either fraudulent or legitimate. Transactions consist of a customer/buyer, a merchant/seller, an amount, and metadata such as the zipcode of the merchant, the age, gender and zipcode of the customer, and the category of the purchase (transportation, food, health, etc.). To separate the fraudulent transactions from the legitimate, we attempt to extract useful features from a set of labeled transactions in hope that these features can be applied to transactions whose true labels are unknown. In order to create those features, we can aggregate all of our training data and define new variables to encode fraud indicators. Examples of this could be the percent of training transactions known to be fraudulent given they are with a given merchant. In addition, we can treat each transaction as an edge in a graph, where the nodes are the customers and merchants, and the weight of the edge is the amount of the transaction. There are many measures defined on graphs to quantify the importance of a node or edge. While calculating all of these measures would improve performance of machine learning models, many graph measures are computationally expensive. With this in mind, we can find a subset of graph features that provide the most correlation with the fraud value, while minimizing the redundancy of the features. To achieve this we use minimum redundancy, maximum relevance feature selection. This method will give each feature a score, we can then train models based on the n-best features, where n ranges from 1 to the total number of features available. With each successive feature we would expect greater performance from the classifiers, however there will be a point of diminishing returns. Based on these results, we will find the best features to balance compute time and accuracy.

2 Introduction

Utilizing machine learning methods to classify credit card fraud transactions is not a novel idea. As far back as 1994 there were major publications on the

use of multi-layer perceptron models to root out fraudsters. Since then, there is a large selection of papers on applying both supervised and unsupervised learning methods to the problem. While the supervised methods take advantage of the labels on the training data, the unsupervised methods look for outliers in the problem space, and consider those to be contenders for fraud. We will focus on supervised methods in this paper. Decision trees, random forests, k-nearest neighbors classifiers, multi-layer perceptron classifiers and support vector machines, will be trained on a subset of the labeled data, and validated against the complement of that subset. While the performance of each of these classifiers is not important, we will look at the effectiveness of all classifiers when they are trained on different features.

3 Importance of Problem

Every year billions of dollars are lost to fraudulent credit card transactions. In an increasingly cashless world, the opportunity for fraudsters grows year over year. Stopping these attempts benefits both credit card companies, who incur a financial loss each time fraud occurs, as well as customers, who must cancel cards and convince credit card companies to reimburse them for unauthorized purchases. STATS HERE

4 Difficulties inherent to problem

There are two large difficulties inherent to the fraud detection problem. Firstly, the number of transactions that are fraudulent is much smaller than the number of legitimate transactions. STAT. Secondly, credit card data is some of the most legally protected data in the world. Companies are rightly reluctant to disclose transaction information, even for purely academic reasons. We can address both of these problems in a single step by using synthetic data. The goal of synthetic data is to create an unlimited amount of data that is both statistically similar to real data, and does not contain any personal information or other compromising content. This addresses the data protection problem by being completely fictional, and the class imbalance problem by artificially inflating the proportion of fraudulent transactions while maintaining the same indicators found in the real world data.

5 Turning transactions into a network

A graph is a mathematical object that consists of a set of nodes and a set of edges. In order to create a graph out of transaction data, we simply need to define what our nodes and edges will be. We define the nodes to be merchants and customers, and edges between them to be the transactions themselves. This is enough to build our graph, however there are some interesting characteristics of this graph that are worth mentioning. Firstly, every transaction consists of

exactly one merchant and exactly one customer. In our dataset there cannot be a transaction between two merchants or two customers. This means that our graph is bi-partite, with the partite sets being the merchants and the customers. This is significant as some algorithms can only be applied to bipartite graphs, and others can be significantly sped up on bipartite graphs. Secondly, a customer may make multiple purchases from the same merchant. In our graph, this would mean that there are multiple edges between the same two nodes, creating a multi-graph. Finally, there is a natural way to introduce a weighting on the edges of this multi-graph. By defining the weight of each edge to be the amount of the transaction, we can apply a whole new set of algorithms to the graph, and also introduce features for both their weighted and unweighted variants, nearly doubling the number of features to consider.

6 New Features + MRMR

The novel work we seek is in considering a large number of graph features and doing feature selection to find the most effective subset of them. Graph libraries such as Networkx allow simple computation of many centrality measures for graphs which can fit in memory. Once all of these features have been calculated there are several python implementations of the original MRMR feature selection which can be used.

7 Related Work

7.1 Prusti

Prusti, Das, and Rath found that including degree centrality, pagerank, and label propagation communities made a significant improvement on 5 supervised learning methods and 2 unsupervised methods[1]. They utilized the Banksim dataset and Neo4j to store the data and to calculate the graph features.

7.2 Banksim

Lopez-Rojas developed Banksim as a tool specifically to combat the data scarcity problem in credit card fraud detection. They used transaction data from a Spanish bank to tune a simulation of customers, merchants, and fraudsters that contained both a relatively high fraud rate of

8 Methodology

9 Graph Model

10 Dataset and Experimental Setup

11 Banksim EDA

12 Features from transactional Data

12.1 Amount

The amount of a transaction, normalized by a minmax scaler.

12.2 Training fraud rate

For each of the nominal categories merchant, customer, and category, the fraud rate is defined by the number of fraudulent transactions with that feature divided by the total number of transactions with that feature.

13 Graph Feature Definitions

13.1 Degree Centrality

The degree centrality of a node is defined as the number of edges adjacent to the node divided by the number of nodes in the graph.

References

[1] D. Prusti, D. Das, and S. K. Rath, "Credit card fraud detection technique by applying graph database model." *Arabian Journal for Science and Engi-*

neering, vol. 46, no. 1-20, 2021.