Identifying Fraudulent Transactions in Mobile Payments

Kaushal A. Alate
Computer Science
Stanford University
Stanford, CA
kalate@stanford.edu

Kadar D. Qian
Symbolic Systems
Stanford University
Stanford, CA
kadarq@stanford.edu

Nicholas T. Lai
Computer Science
Stanford University
Stanford, CA
nicklai@stanford.edu

***Abstract*—Bayesian networks are a popular and effective means of modelling financial fraud. But do the same methods which have worked previously for credit card fraud carry over to mobile payments? We constructed a Bayesian network using a dataset of mobile payments created by the PaySim simulator and computed its predictive accuracy. This yielded a consistent 87% accuracy when classifying the transactions.**

Keywords—mobile payments, fraud, prediction, Bayesian networks

I. Introduction

It is estimated that there are 690 million people in the world with mobile payment accounts, and the number continues to rise every day [1]. Seen as a convenient means of payment for the portion of the population without bank accounts, mobile payments are increasingly dominating over other means of payment, particularly in China. As mobile payments through apps such as Venmo and WeChat become more popular, the number of fraudulent transactions has risen as well. Hence, there is a significant need for a model which can accurately predict if a financial transaction is fraudulent or not.

A Bayesian network is a directed acyclic graph where nodes correspond to random variables and edges correspond to conditional dependence relationships. Due to their containing nodes and edges in such a manner, Bayesian networks are commonly used to answer questions about how likely an event would be to have occurred given a specific set of data points. In our case, we're interested in the probability of fraud happening given a certain set of conditions being held: perhaps it's time of transaction, or the amount of money that was spent. In the context of our financial fraud data, the variable names we'll be investigating are described below in the "Dataset" section.

There are three types of inference which are typically done with Bayesian networks: inferring unobserved variables, parameter learning, and structure learning. In our model, we'll primarily be dealing with structure learning, since we want to know how the variables relate to one another.

Given a Bayesian network, how does one evaluate its accuracy in modeling financial data? The problem is essentially finding the probability P(G|D), where G is the graph structure and D is the data that we know. Fortunately, we can estimate P(G|D) using a metric called the Bayesian score, which increases as our graph structure and data are more "fit" together. Since one often does not know the conditional relationships beforehand, one must learn the structure of the Bayesian network given some data using algorithms designed for this purpose. Several such algorithms exist: greedy local search, K2, chow-liu, etc. For the purposes of this assignment, we have decided to use greedy local search as it is quick to implement and quick to run.

II. Literature Review

Previous papers have discussed using Bayesian networks to model financial data, but none so far have investigated in particular how the use of Bayesian networks plays into modeling specifically mobile payments and fraud. We wish to investigate as to whether the mobile payment space can be modeled in the same manner, or if there are quirks to mobile payments which we can exploit to improve on traditional credit card models.

Lev Mukhanov [2] compares using a Bayesian network with using a Naive Bayes classifier to predict credit card fraud. Of particular note is his emphasis of the MDL principle, which he uses to enhance the Bayesian score. He concludes that Bayesian networks are more accurate than Naive Bayes classifiers, but that Bayesian networks require greater complexity to train the data.

Sam Maes et al. [3] compare the tradeoffs of using a Bayesian network versus an artificial neural network to model financial data. There's a good discussion here about what exactly constitutes credit card fraud and issues with past attempts to detect fraud. They conclude that Bayesian networks are more accurate and have a shorter training period on financial data, but the actual fraud detection is considerably faster with artificial neural networks.

Abbasi et al. [4] have designed a meta-framework to use for financial fraud detection. Although not exactly a Bayesian network, there are good insights into which features are valuable to have and which are not. There is also some insight into ways we can extend our current model in the future: by layering models together.

III. Data Processing

A. The Dataset

The original dataset of mobile payment transactions was created by the simulator "PaySim" and is based on one month's worth of real financial data from an African mobile payments company. It contains 6,362,620 mobile payment transactions. The variables in the original dataset are:

1) Step: The simulator was run for 744 time "steps", where each step is one hour (so a total of approximately 30 days), and each transaction is labelled with its step.

2) Type: One of Cash-In, Cash-Out, Debit, Payment and Transfer.

3) Amount: The amount in the transaction between 0 (surprisingly) and 92 million.

4) nameOrig: Name of the customer who initiated the transaction

5) oldBalanceOrig: Initiating customer's balance before the transaction

6) newBalanceOrig: Initiating customer's balance after the transaction

7) nameDest: Name of customer who is the destination of the transaction

8) oldBalanceDest: Destination account's balance before the transaction

9) newBalanceDest: Destination account's balance after the transaction

10) isFraud: Whether the transaction in the simulation is fraudulent

B. Data Cleaning

Some of the data seems inaccurate, for instance the transactions for which the transaction amount is 0. This may be a bug in the simulator (whose code we do not have access to) or a deliberate decision to reflect inaccuracies in real-world payment data. We removed all transactions with amount equal to 0.

After removing all such transactions, there were some transactions with positive "amount"s that had oldBalanceOrig and newBalanceOrig equal to 0. This would be possible if money were withdrawn through an overdraft. However, since there are no negative balances in the dataset, it is unclear whether these data points represent overdrafts or whether this is simply missing data. To err on the side of caution, we assumed the latter and deleted datapoints with oldBalanceOrig equal to 0.

We used type, amount, oldBalanceOrig and nameDest (along with isFraud) to construct our Bayesian network. We believed that these variables would be effective in predicting fraud and that the additional computation time of adding more variables would outweigh their potential benefit.

C. Data Discretization

Bayesian networks in the Pomegranate library can only work with discrete data, and we hence had to discretize the data. These are the discretized variables which were continuous in the original dataset along with their interpretation:

1) amountCat: If amountCat is *x*, the amount is between 1,000,000(*x*-1) and 1,000,000*x*.

2) oldBalOrigCat: If oldBalOrigCat is *x*, the oldBalanceOrig value is between 100,000(*x*-1) and 100,000*x*.

We also numerized the categorical variables before constructing the Bayesian network. Here is the mapping for each of the two numerized categorical variables:

1) type: Cash-In is mapped to 0, Cash-Out to 1, Debit to 2, Payment to 3 and Transfer to 4

2) nameDest to *nameDestNum*: "C"ustomer accounts to 1, "M"erchant accounts to 2 TABLE

The fifth and last variable in our processed dataset was the dependent indicator variable isFraud, which is 1 is the transaction is fraudulent and 0 otherwise.

IV. APPROACH

We constructed a Bayesian network on these five variables using the Pomegranate library. We used greedy search due to its speed, which would allow us to debug quickly and experiment with different amounts of data.

After constructing the Bayesian network, we predicted whether each transaction in the test set was fraudulent. We computed the probability that it is fraudulent and predicted fraud if this probability was greater than 0.5 (which would imply that the probability of the transaction being legal is less than 0.5).

V. Results & Discussion

A. Structure Interpretation

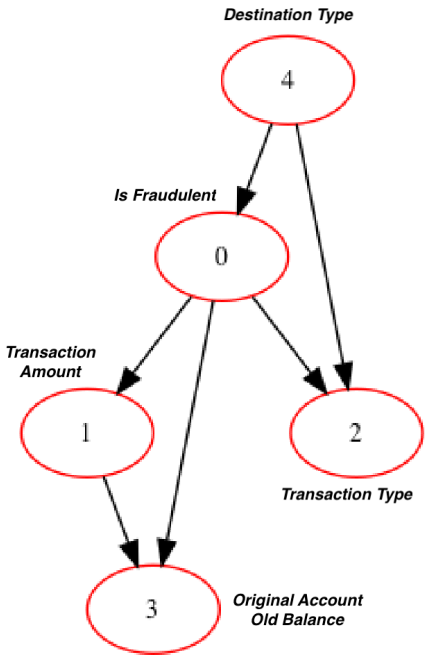

Figure 1: Learned Bayesian network structure

After learning the structure that maximized our Bayesian score, we applied this model to infer the missing values from our development and test sets. The model can be seen in the figure above.

Interpretation of the model can be done by associating each of the digits 0 through 4 as follows:

0: Is Fraud (variable to infer)

1: The amount in the transaction

2: The type of transaction that occurred

3: Original old balance

4: Destination type - customer or merchant

We can interpret this now as each of these factors bearing influence on others. In fact, the structure learned makes intuitive sense.

First, we see from the structure that the destination account type influences whether the transaction was fraudulent and the type of transaction that occurred. It is understandable how if the destination account were a merchant, then the transaction would be less likely to be fraudulent, as we can observe simply by looking at the data. However, if the destination were an individual customer, it would be more likely for the transaction to be fraudulent. Additionally, official businesses and merchants would take different types of transactions, which explains the flow of influence from destination type to the transaction type.

Second, we see that being a fraudulent transaction influences the transaction amount, the original account balance, and the transaction type. Fraudulent transactions may take certain amounts out of accounts that have enough balance using more untraceable payment types like cash.

Finally, the transaction amount influences the original account old balance. Of course, knowing the transaction amount gives us some insight to the original account balance, since one is unable to take out more money than the quantity of the original balance.

B. Structure Accuracy

Once we were satisfied with our development set's accuracy, we ran inference on our test set. Both the development and test sets contained 30% of the total transactions and were missing fraud classifications to be inferred. For the development set, we were able to correctly identify 2182 transactions of 2501 as fraudulent or valid. This yielded an accuracy of 87.24%.

After working with this development set, we applied our model to make predictions with the test set, which was done once to get an accurate sense of how good our model was.

The model correctly identified 2184 transactions of 2502 total, yielding an accuracy of 87.29%. Since the accuracy on the development and test set were near completely similar, we were able to deem our model consistently accurate when identifying fraudulent and non-fraudulent mobile transactions.

C. Misclassified Transactions

With 87% accuracy, we are able to confidently classify mobile transactions. However, we wanted to take a deeper look into which transactions were misclassified and any patterns that developed.

Of the 318 total misclassified in the test set, 170 of them were not identified as fraudulent despite actually being so. Nearly all 170 of them had '1's assigned to each of the variables. This means that when the amount of the transaction was small, the transaction type was cash-out, the balance was small, and the recipient was a customer, our model tended to identify them as non-fraudulent despite them being fraudulent. This makes some intuitive sense because transactions with small amounts or balances are less likely to be fraudulent. Still, this is remarkable because our training data contain examples of both fraudulent and non-fraudulent transactions that had all '1's assigned. We could address this further in our future work.

VI. CONCLUSION & FUTURE WORK

Ultimately, we see that Bayesian networks can provide a good foundation for modeling mobile financial transactions and classifying fraudulent ones. With 87% accuracy using just a handful of discretized variables, one can be optimistic about the potential of Bayesian networks in predicting fraud in mobile payments.

	Actually Fraud	*Actually Not Fraud*
Identified Fraud	1058	148
Identified Not Fraud	170	1126

Figure 2: Predicted and actual classifications

Going forward, it is most important that we reduce misclassification errors – both Type I and Type II. As seen in the confusion matrix from Figure **()**, we had about 6% of our predictions as false positives, and about 6.7% of our predictions as false negatives. Until this number is closer to 0, we cannot yet deploy our model. We do not want to identify valid transactions as fraudulent or miss fraudulent transactions.

That being said, the Bayesian network offers promise not only in its accuracy but in its treatment of false positives and false negatives. While incorrectly tagging transactions as fraudulent can be bothersome to the customer and the

merchant, the case can be made that it is better than missing a fraudulent transaction. The Bayesian network approach yields fewer false positives than false negatives, which aligns with this preference.

In order to increase the accuracy, we may want to consider more factors, such as the balances of the recipient of the money or the type of currency that is used. Additionally, we are currently discretizing our data, which means that our level of granularity is restricted to ranges. Adding support for continuous values, particularly for the quantities of currency may help increase accuracy as well. However, using continuous variables would add significantly to the time complexity, and thus would require some approximations to run as quickly as our implementation now.

ACKNOWLEDGMENT

The authors would like to thank Professor Mykel Kochenderfer for technical guidance and Harper Carroll for assisting with some of the conceptual foundations of this project.

REFERENCES

[1] GSMA. “State of the Industry Report on Mobile Money”. In: (2017).

[2] Lev Mukhanov. “Using Bayesian Belief Networks for credit card fraud detection”. In: (Feb. 2008), pp. 221– 225.

[3] Sam Maes et al. “Credit Card Fraud Detection. Applying Bayesian and Neural networks”. In: (Dec. 2018).

[4] Ahmed Abbasi et al. “Metafraud: A Meta-learning Framework for Detecting Financial Fraud”. In: MIS Q. 36.4 (Dec. 2012), pp. 1293–1327. ISSN: 0276-7783. URL: http://dl.acm.org/citation.cfm?id=2481674.2481688.