Capital One Fraud Detection Algorithm

Maggie Huiying Mao
Thomas Metzger
Sazzadur Rahaman
Yizhi Sun

Friday, April 1, 2016

1 Data Preparation

The data contain approximately 89 million observations of 69 attributes useful in predicting whether a transaction is fraudulent or legitimate. These attributes relate to the cardholder's account, the plastic credit card itself, the location of the transaction, and the specific transaction properties.

Before fitting the model to classify each transaction based on its covariates, several adjustments were made to the data set and individual covariates to facilitate the statistical analysis. Firstly, many of the covariates contained missing values. If a categorical variable was missing, the mode of the variable was imputed into the missing value; if a numeric variable was missing, the median of the variable was imputed. Similarly, some variables contained the same value for every observation; for example, attribute PIN_BLK_CD was at level "*2" for every observation considered, so as it did not provide any unique information regarding a transaction it was excluded from analysis. Finally, the class of several variables was changed based on context. For example, while a value of "840" for attribute SRC_CRCY_CD might look like an integer, it must be classified as a category in order to ensure the model fits it appropriately.

Attributes represented as dates and times provided an interesting challenge. The date string was parsed into individual components of day, month, and year, and so the individual attribute was split into three distinct attributes. The date attributes were then able to detect differences in season, for example. The time was similarly parsed into three values: second, minute, and hour, and thus able to detect differences such as morning and evening.

Of the large data set provided, it was first randomly split into a subset including every fraudulent transaction along with an equal number of randomly selected legitimate transactions. 80% of these observations were used to train or create the model, and 20% of the observations were used to cross-validate the model as well as determine the probability thresholds for classifying transactions. Thus the model is implicitly biased toward fraud detection. Cross-validation is done for two reasons: first, it prevents over-fitting the model, which would classify transactions too specifically to this particular data set and consequently perform poorly for new data. Second, the cross-validation serves to assess how well the model predicts known observations, without using observations already used in the model training process. Our probability thresholds were chosen differently to account for the unbalanced loss function.

2 Model Fitting Technique

A gradient boosted decision tree is created to label each transaction as either fraudulent or legitimate, based on its attribute values. Gradient boosting begins with a constant prediction on every transaction, and subsequently adds more specific classification rules as individual trees based on a minimization of some objective function. This objective function balances the misclassification error with the complexity of the model. Ultimately, the gradient boosted tree will yield some score, akin to a logistic regression, which will yield the final transaction classification of fraud or legitimate.

The gradient boosted classification tree method creates a single probability based on weighted probabilities from smaller sub-trees. Each probability is then assigned a weight based on the gradient of the Taylor expansion of the objective function, in an attempt to minimize this objective function. For example, suppose sub-tree 1 yields a probability of $\hat{p}_1$; sub-tree 2 predicts $\hat{p}_2$; and so on. A final $\hat{p} = w_1 \cdot \hat{p}_1 + w_2 \cdot \hat{p}_2 + \cdots + w_{100} \cdot \hat{p}_{100}$ is then created; if $\hat{p}$ is greater than some threshold, the transaction is predicted to be fraudulent, and legitimate otherwise. This threshold will be determined from the loss function.

2.1 Loss Function

Actual	Predicted	Absolute Loss
0	0	0
1	1	0
1	0	$l_1 = \begin{cases} \text{L:} & .02 \times 75 + .0075 \times \text{AUTHZN\_AMT} \\ \text{H:} & .05 \times 275 + .005 \times 4 \times \text{AUTHZN\_AMT} \end{cases}$
0	1	$l_2 = \text{AUTHZN\_AMT}$

Table 1: Loss Function for misclassified transactions, for Low and High credit limits.

The loss function describes the expected loss of an individual transaction. Recall that AUTHZN_AMT describes the amount of money requested by the cardholder; that is, the value of the transaction in question. A fraudulent transaction that is misclassified (not detected) results in a loss equivalent to this authorization amount - the crook has "gotten away with" the fraudulent purchase on the credit company's dime. The loss from a legitimate transaction that is misclassified as fraudulent is more subtle: the customer loses confidence in the credit company and runs a small risk of canceling their account, and, the credit company loses out on its share of the authorization amount.

Note that due to the uneven loss function, a gradient boosted probability of greater than .50 will not necessarily lead to a classification as fraud. The classification threshold, based on the multiple gradient-boosted trees and loss function, is 0.184. This allows much greater customization of the model and greater capacity for the model to learn as more transactions are made. The optimization process of this cutoff, as a function of the losses, is shown as follows:

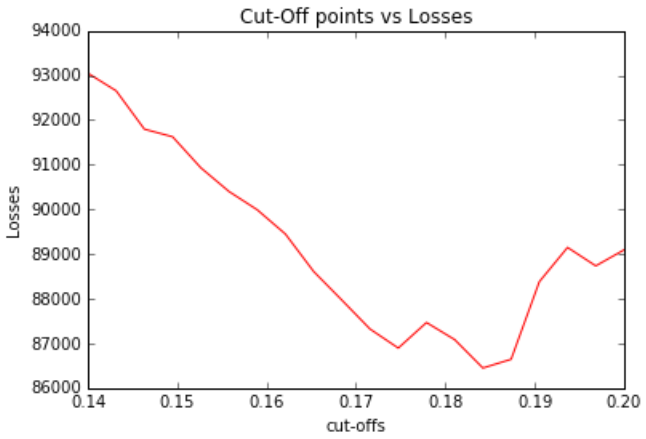

Figure 1: The optimal threshold probability is determined to be about 0.184.

Note that this curve is not a function of the proportion of fraudulent transactions, but rather a function of the different potential cutoff values examined.

3 Model Performance

It is important to note that while a model can be fitted based on a huge data set containing vast amounts of information, ultimately, only the person making the transaction can know for certain whether a purchase was fraudulent or legitimate. While this model has a specific misclassification rate, it is based on the data provided which might contain misclassified transactions themselves.

This "unknowability" of the truth is one of the most important concepts in statistics, and prevents any model from achieving perfection under any real-world circumstances.

3.1 Recall and Precision

	Predicted Fraud	Predicted Negative
True Fraud	A	B
True Legitimate	C	D

Table 2: Definitions of recall and precision, based on counts.

Performance is judged using two industry standard measurements: recall, defined as the ratio of correctly predicted frauds to all true frauds, or $\frac{A}{A+B}$; and precision, defined as the ratio of correctly predicted frauds to all fraud predictions, or $\frac{A}{A+C}$. It is important to note that in the case of a 0-1 loss, these numbers would ideally be near 1.0 in both cases. However, since the loss function used here unevenly penalizes false positives and false negatives, the recall and precision values must be considered in the context of minimizing the cost to the credit company. The recall and precision of our model based on the cross-validation data set are presented as follows:

	Recall	Precision
Observed Value	0.974	0.691

Table 3: Observed recall and precision.

The classification threshold of 0.184 results in high recall but low precision. That is, our model will produce a relatively large number of false positives, but in the context of reducing cost, the false positives are considered acceptable.

3.2 Receiver Operating Characteristic (ROC)

The receiver operating characteristic (ROC) curve represents a comparison of false positive rate versus true positive rate. It can intuitively be thought of as representing the trade-off between perfectly conservative models, which would classify all transactions as fraudulent and all transactions as legitimate, respectively. The ROC curve generated by our model is shown as follows:

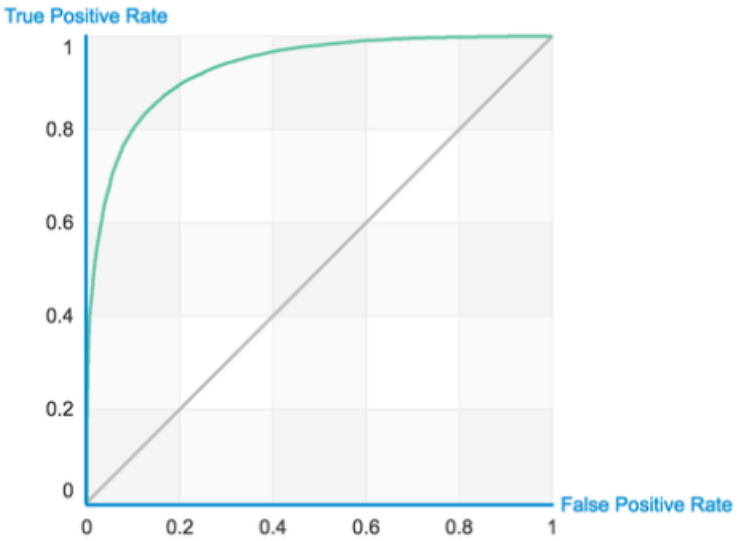

Figure 2: ROC curve.

4 Discussion

An obvious drawback to our model is that the velocity features were unable to be accommodated. Our model can classify existing fraud, but cannot detect impending fraud. With more time and resources to build the model, we considered examining individual accounts with a fraud associated with them to determine how certain attributes changed over time in a way that could significantly predict a future fraudulent transaction. This would lead to an even better classification rate by considering not just individual transactions, but how they compared to previous transactions associated with the same account.

Another clear mistake was in the original data partition. This partition of equal proportions of fraud and legitimate transaction in the training data, along with the unbalanced loss function, will make the model more prone to predicting fraud. In hindsight, we should have trained the model on a more representative data set.

We collectively learned a great deal in our work on this project. We were previously unfamiliar with such enormous data sets, but these big data sets are becoming increasingly more common and important so methods of analysis will continue to become more powerful and sophisticated.