Fraud Detection for Online Retail using Random Forests

Eric Altendorf, Peter Brende, Josh Daniel, Laurent Lessard

Abstract

As online commerce becomes more common, fraud is an increasingly important concern. Current anti-fraud techniques include blacklists, hand-crafted rules, and review of individual transactions by human experts. These methods have achieved good results, but a significant number of costly fraudulent transactions still occur, and the fraud detection process is expensive due to its heavy reliance on human experts. Our goal is to improve the quality of fraud detection and decrease the need for costly human analysis of individual transactions. We propose two modifications to the approval process for an online retailer based on a Random Forests classifier.

1. Introduction

We have obtained data from two sources: an online retailer of computer equipment and peripherals, and Quova, a provider of geolocation data. The retail data consists of about 250,000 records of orders placed online over a 3 month span. Currently, when the retailer receives an order request, they decide whether to approve the order and ship the goods using the decision process illustrated in Figure 1. The flow consists of a series of decision nodes:

1. Pre-filter: rejects "blacklist" of bad customers, bank rejections, AVS (card-address mismatch): rejects 35,000.

2. Send to human: Uses simple rules like "contains iPod" to decide whether to send order to human analyst for review: 30,000 to *Analyst*, 185,000 to *Accept*.

3. Analyst: Expert reviewer manually analyzes orders and rejects based on likelihood of committing fraud, defaulting on credit, or returnin-product: reject 4,500.

All orders that are not rejected by the pre-filter or human expert are approved. Of these, 444 were later reported as fraudulent by the credit card holder.

The current process contains two sources of expense for the retailer that could be decreased with the use of intelligent classifiers. First, at the *Analyst* node, human labor is used to review individual orders. This labor is both slow and expensive relative to automated systems.

Second, of the orders that are passed to the *Accept* node, a certain fraction are fraudulent. Potentially, an intelligent classifier at this stage could significantly reduce the number of fraudulent orders approved while only slightly increasing the number of valid orders that are rejected. In Section 3, we discuss the tradeoffs between a high rate of fraud detection and mislabeling non-fraud instances.

Therefore, we propose a new order review process for the retailer as illustrated in Figure 2. Essentially, we augment the existing system with two additional decision nodes powered by intelligent classifiers:

1. Analyst simulator: either rejects, or sends to a real human for review. As explained in Section 5, the analyst simulator does not accept orders outright. To build a prototype for this classifier, we train on the 28,729 items that were reviewed by a human.

2. Fraud classifier: as a final step before approval, each order passes through our custom fraud classifier. We build our model by training on a set of approximately 210,000 orders (containing 444 fraudulent transactions) that were approved by the current system. (In Section 4, we deal with the issue of balancing FP and TP.)

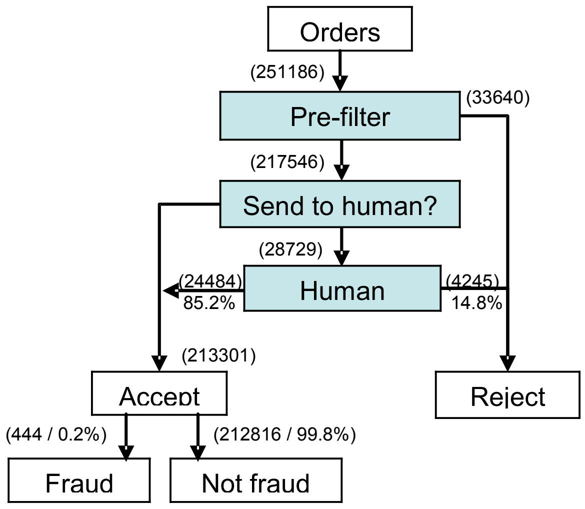

Figure 1. Current Infrastructure

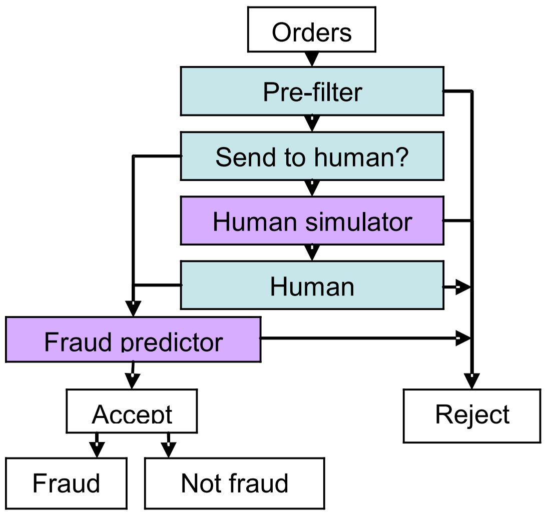

Figure 2. Proposed Infrastructure

2. Data sources and feature engineering

We obtained data from two primary sources. First, a major online retailer of computer equipment and peripherals supplied us with 251,186 records of orders placed over a 3 month span in 2004, which identify the customer, time/date of the order, items ordered, and credit card and shipping information. These records also include the IP address from which the order was placed, allowing us to supplement the orders data with geolocation and network information data obtained from QUOVA, inc. This allows us to determine, for each order, the continent, country, state/province, city, postal code of the computer from which the order was placed. It also provides information about the local network connection (modem, cable, ISDN, DSL, mobile wireless, etc.) and the IP routing method (anonymizers, proxies, mobile gateways, satellites). Finally, it lets us determine information about the network to which the user is attached (the carrier's name, such as Stanford or Comcast, whether or not the user is on the AOL network, which geolocates all users to specific AOL office locations, and the domain name of the user's IP).

Much of our time investment was on data preparation and intelligent feature engineering, a process guided by our collaborators in the finance department at the retail company. They helped us identify characteristics typical of users placing fraudulent orders, who:

- Tend to be transient geographically
- Use non-standard Internet connections
- Exhibit unique patterns in purchase timing
- Purchase items that are easily resellable and small in size

Based on these ideas, we extracted a total of 72 features, including

- Billing/shipping address match
- Distances between location of network connection and billing and shipping addresses
- Indicator variables for specific products
- Indicator variables for both "hot" (e.g., iPod, iPaq, handheld computer) and "cold" (e.g, printers, cables) keywords occuring in item descriptions
- Item cost per weight
- Indicator variables for time-of-day and day-of-week
- Credit card history: number of recent orders, cumulative recent charges

In addition, we have some international orders, but probably too few to learn probabilities of fraud from each of over 200 countries. Therefore, we omit country identity and instead substitute demographic and socioeconomic statistics obtained from the CIA World Factbook, such as number of internet connections and telephones, average income, and unemployment rate.

3. Random Forests

Random Forests™ is an algorithm for classification and regression (Breiman 2001). Briefly, it is an ensemble of decision tree classifiers. The output of the Random Forest classifier is the majority vote amongst the set of tree classifiers. To train each tree, a subset of the full training set is sampled randomly. Then, a decision tree is built in the normal way, except that no pruning is done and each node splits on a feature selected from a random subset of the full feature set. Training is fast, even for large data sets with many features and data instances, because each tree is trained independently of the others. The Random Forest algorithm has been found to be resistant to overfitting and provides a good estimate of the generalization error (without having to do cross-validation) through the "out-of-bag" error rate that it returns.

Our data sets are quite imbalanced, which in general, can lead to problems during the learning process. Several approaches have been proposed to deal with imbalance in the context of Random Forests including resampling techniques, and cost-based optimization (Chen, et. al. 2004).

We take a different approach, using regression forests and classifying based on an adjustable threshold. By changing the threshold level, we create a set of classifiers, *H*, each of which has a different false positive (FP) and true positive (TP) rate. The tradeoff between the FP and TP rates is captured in the standard ROC curve (see, for example, Figures 5 and 6).

4. Experiments and results

For our experiments we worked with the OpenML (Machine Learning) library developed by Intel, Corp.

We specified the maximum number of features to be considered at each tree node to be 4 and the out-of-bag sampling rate as 0.6.

For both analyst and fraud problems, we trained the Random Forest classifier on the first 80% of our dataset and used the remaining 20% for validation.

For each validation sample, the regression model returns a response y between 0 (indicating the positive class) and 1 (negative class).

In Figure 3, we show a histogram of these responses for the analyst problem. Note that about half of the rejected samples are easy to classify.

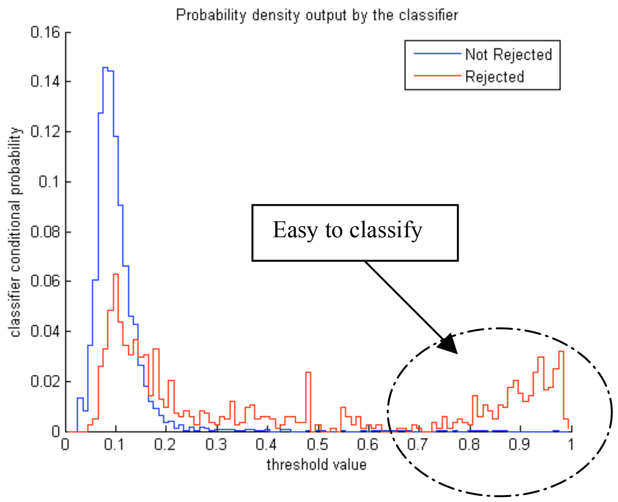

Figure 3: Response distribution for "Analyst"

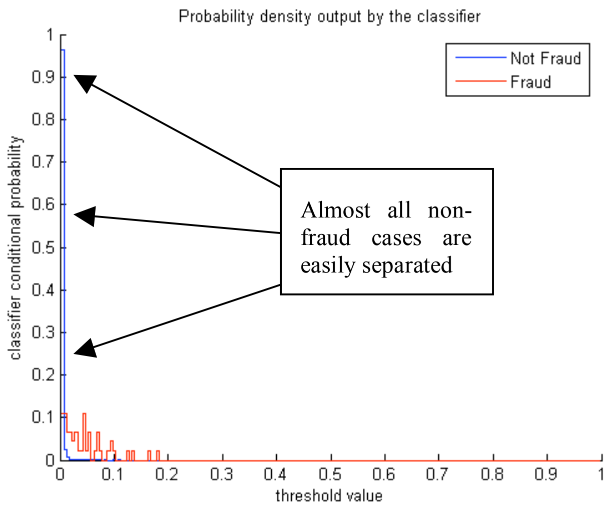

Figure 4: Response distribution for "Fraud"

For the fraud problem, the response histogram (Figure 4) looks quite different. Non-fraud cases appear to be highly distinguishable.

Since fraud cases are very rare (only 444 cases out of 213,301), we have to be careful about picking operating points on the ROC curve (Figure 6). If we zoom in near FP = 0, we see that the probability isn't actually zero, and this has consequences.

5. Optimizing the decision threshold

To select an appropriate threshold value and, in turn, a classifier, we must specify some objective in the problem. In this problem, our success can be measured by the net change in expected profit.

In general, if we can assign some cost to each false positive and to each true positive, it is possible to choose amongst the set, *H*, of classifiers (described above). Letting p_0 and p_1 be the prior probabilities for classes 0 and 1, and c_0 and c_1 the respective misclassification costs, our objective is defined as:

$$\begin{aligned} f &= p_0 FP c_0 + p_1(1 - TP)c_1 \\ &= p_0 FP c_0 + p_1(1 - g(FP))c_1, \\ &\quad \text{where } g() \text{ specifies the ROC curve} \end{aligned}$$

$$\frac{\partial f}{\partial FP} = p_0 c_0 - p_1 c_1 g'(FP)$$

Setting this to zero, we get:

$$g'(FP) = \frac{c_0 p_0}{c_1 p_1}$$

The optimal classifier corresponds to the point on the ROC curve where the slope is equal to a ratio involving the prior probabilities for the two classes and the two costs.

The "analyst simulator" problem

For the "analyst simulator" problem, the difficulty lies in assigning costs to the two types of classification errors. Since this (classification) problem occurs far upstream in the retailer's approval process, the costs and benefits of each type of error can only be accurately determined by doing a controlled experiment; that is, by approving certain orders that would normally be rejected, and observing the outcome.

If we knew the cost of human analysis (per order reviewed) and the cost of a false positives (the profit we would make from the transaction), it would be straightforward to tradeoff the threshold parameter and find the policy that maximizes profit as described in Section 4.

Notice, however (see Figure 5), that a moderate TP rate can be achieved while maintaining a FP close to zero. This means that we can easily choose a decision boundary which will reliably pre-reject a sizeable portion of the samples without needing the intervention of a human analyst.

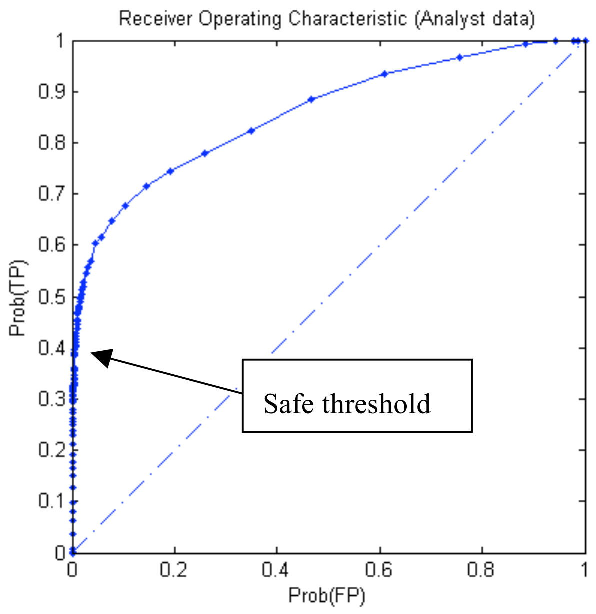

Figure 5: ROC for "analyst" problem

We propose, as a conservative policy, to only pre-reject cases for which we are virtually certain there will be no false positives. From the ROC plot, this corresponds to 0.4 on the TP axis. If we take the prior probability of rejection into account, we can expect to pre-filter 0.4 × 0.148 = 6% of the samples sent to the analyst.

The fraud prediction problem

Suppose we wanted to detect 60% of the fraudulent cases. This would cause a 0.2% FP rate. Taking prior probability into account, we can expect to reject 267 fraudulent transactions at the expense of also rejecting 426 legitimate ones (over the entire data set).

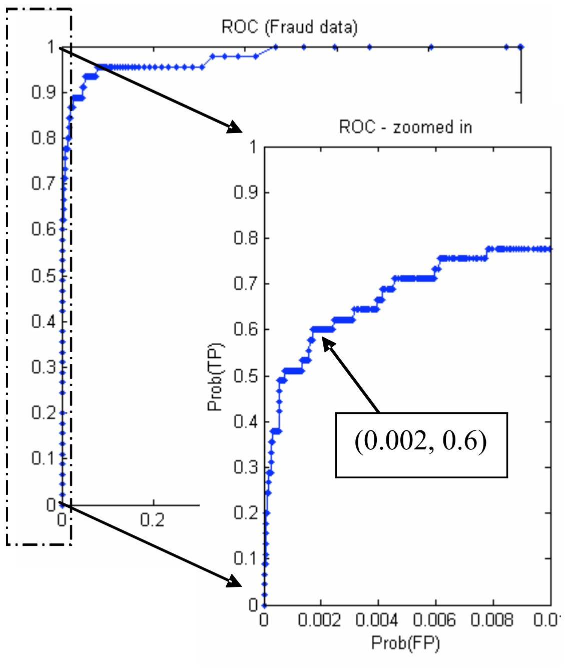

Figure 6: ROC for "fraud"

Generally, allowing a fraudulent purchase costs the retailer the full price of the item, while denying a legitimate purchase only results in a loss equal to the profit margin. So in the example above, our classifier will save the retailer money under the reasonable assumption that the profit margin on items is less than 50%.

As in the Analyst case, knowing our exact profit margin will allow us to design a more aggressive policy. The retailer simply needs to pinpoint the average cost of a fraudulent order and the average profit margin on items sold. Then, using the procedure described above, we pick the point on the ROC curve that optimizes expected misclassification cost.

6. Discussion

Cascades of classifiers are a popular method of dealing with rare event detection (see, for instance, Viola and Jones, 2001, Wu, et al., 2005).

The main assumption is that we can train *n* independent classifiers all with a very high (say, 0.999) TP rate and medium (say, 0.5) FP rate. Then, by requiring all *n* classifiers to agree on a positive detection, we can reduce the FP rate to 0.5^n, while retaining a TP rate of 0.999^n. For $n =$ 10, say, we would obtain very good TP and FP rates.

One popular application of cascades is the face detection image processing problem. In face detection, non-face images are quite diverse. This makes it easy to enforce independence of the cascade nodes. All we have to do is design a long chain of classifiers where each one specializes in rejecting a particular type of non-face object with a near-perfect TP rate. The question is: "can we always enforce the independence of the classifiers?".

For the fraud problem, we suspect not. Although it is a rare event detection problem, it is fundamentally different from the face detection problem. As we have seen from the ROC, most non-fraud cases look similar and are easily identified. Therefore, training on subsets of the non-fraud data would not generate independent classifiers and we would not reap the benefit of exponentially decreasing errors.

A standard cascade will not work with the analyst problem either; the ROC plot shows it is not possible to obtain the needed high TP and moderate FP. We do, however, have points on the ROC with very low FP rates. We believe it should be possible to build an "inverted cascade" which exploits this ROC shape by requiring all classifiers in the cascade to agree on a negative detection.

We would like to thank Dr. Gary Bradski at Intel Research for his valuable help and advice.

7. References

Breiman, Leo (2001). "Random Forests". Machine Learning 45 (1), 5-32.

Chao Chen, Andy Liaw and Leo Breiman (2004). "Using Random Forest to Learn Imbalanced Data."

Paul A Viola, Michael J. Jones (2001) "Rapid Object Detection using a Boosted Cascade of Simple Features." CVPR, 1:511-518.

Jianxin Wu, Matthew D. Mullin, James M. Rehg. (2005) "Linear Asymmetric Classifier for Cascade Detectors." ICML pages 993-1000.