A Machine-Learned Proactive Moderation System for Auction Fraud Detection

Liang Zhang
Yahoo! Labs
Sunnyvale, CA, USA
liangzha@yahoo-inc.com

Jie Yang
Yahoo! Labs
Sunnyvale, CA, USA
jielabs@yahoo-inc.com

Wei Chu
Microsoft
Bellevue, WA, USA
chu.wei@microsoft.com

Belle Tseng
Yahoo! Labs
Sunnyvale, CA, USA
belle@yahoo-inc.com

ABSTRACT

Online auction and shopping are gaining popularity with the growth of web-based eCommerce. Criminals are also taking advantage of these opportunities to conduct fraudulent activities against honest parties with the purpose of deception and illegal profit. In practice, proactive moderation systems are deployed to detect suspicious events for further inspection by human experts. Motivated by real-world applications in commercial auction sites in Asia, we develop various advanced machine learning techniques in the proactive moderation system. Our proposed system is formulated as optimizing bounded generalized linear models in multi-instance learning problems, with intrinsic bias in selective labeling and massive unlabeled samples. In both offline evaluations and online bucket tests, the proposed system significantly outperforms the rule-based system on various metrics, including area under ROC (AUC), loss rate of labeled frauds and customer complaints. We also show that the metrics of loss rates are more effective than AUC in our cases.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*parameter learning*; H.2.8 [**Database Management**]: Database Applications—*data mining*

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Auction, Fraud Detection, Multi-Instance Learning

1. INTRODUCTION

Since the commercialization of the world wide web in the mid-1990's, online marketplaces have been widely explored by commercial organizations for brand awareness and revenue sources. Individuals are able to buy and sell a broad variety of goods and services worldwide on online auction and shopping websites, e.g. eBay and Amazon. However, criminals have also attempted to conduct fraudulent activities against honest parties for the purpose of illegitimate profit. On Internet auction sites, auction fraud mainly involves fraud attributable to the misrepresentation of a product advertised for sale or the non-delivery of products purchased through an Internet auction site. Malicious sellers may post an (even non-existing) item for bidding with false description to deceive the buyer concerning its true value, and request payments to be wired directly to them. By using wire transfer services, the money is virtually unrecoverable for the victim. Similarly, malicious buyers may make a purchase via a fraudulent credit card where the address of the card holder does not match the shipping address. Both consumers and merchants can be victims of online auction fraud, as well as the commercial auction websites.

Normally, it is the commercial auction websites' obligation to provide insurance to cover the users' loss up to a certain amount. To reduce this compensation as well as maintain their reputation, the auction websites usually employ human experts to investigate their sellers and buyers as often as possible. Clearly the workload is prohibitive to look into every single event. In practice, moderation systems are widely deployed to detect suspicious events proactively for further inspection by human experts, and most moderation systems operate with rule-based decision-making processors that only funnel the suspicious cases to human experts and leave the remaining without inspection. The rules are created by expert knowledge to represent sellers' suspiciousness on fraudulent activities. The outputs of these rules are usually boolean, either positively or negatively suspicious. The final moderation decision is based on a weighted combination of the rule outputs, where the weights are also determined by expert knowledge. After deployment, such a moderation system is capable of automatically selecting a subset of suspicious events for investigation to keep experts' workload at a reasonable level. However model update in such rule-based moderation systems is still not automatic, since the weights on rule outputs have to be set manually by human experts.

Patterns of auction fraud are changing dynamically and rapidly. To maintain the selection/filter accuracy, moderation systems have to be updated periodically to catch the drifting patterns. It is desirable to design a learning system that is capable of automatically optimizing weights for the rules based on recent observations. Motivated by applications in a commercial online auction website, we develop various advanced machine learning techniques in the proactive moderation system. The existing rule-based moderation

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

system in the online auction website selects 20%-40% of the overall events for experts to examine and label. Our initial system is simply built upon the labeled subset by optimizing a generalized linear model. By noting the imbalanced labels (fraud rate is usually under 1%) and the limitation of rule-based features designed for the fraud detection, we improve the system by constraining the weights to be positive and introducing imbalanced/weighted loss functions. To overcome the selection bias in labeling, we also include the remaining unlabeled cases into training for unbiasedness. Being aware of specific noise patterns in the expert labels, we further enhance the optimization as done in multi-instance learning problems. The final model is formulated as optimizing unbounded generalized linear models in multi-instance learning problems, with intrinsic bias in selective labeling and massive unlabeled samples.

2. OUR METHODOLOGY

Our application is one of the major online auction sites in Asia. Lots of items are posted for bidding every day. Each item is sent to the proactive anti-fraud system to assess the risk of being a fraud. The existing system is featured by:

- **Rule-based features**: Human experts with many years of experience created more than 30 rules for fraud detection. Each rule can be regarded as a binary feature that indicates the fraud likeliness. For example, a possible rule could be whether the seller's rating or reputation score is higher than a certain threshold. However, we cannot list the rules and their importance here due to confidentiality.
- **Linear scoring function**: The existing system only supports linear models. Given a set of weights on features, the fraud score is computed as the weighted sum.
- **Selective labeling**: If the fraud score is above a certain threshold, the item will be investigated by human experts, otherwise it will be passed by the system. The final result is labeled as fraud or clean. Items of higher scores have higher priority to be reviewed by experts.
- **Fraud churn**: Once one item is judged as fraud, it is very likely that the seller is not trustable and may be also selling other frauds, therefore all the items submitted by the same seller are labeled as fraud too, and the seller's account will be suspended by the website immediately.
- **User feedback**: Buyers can file a claim if they become victims of fraud. Similarly sellers may also complain if his/her items have been judged as fraud mistakenly.

Motivated by these specific attributes in the existing moderation system, we propose several statistical machine learning models incrementally to improve the fraud detection performance.

2.1 Weighted Logistic Regression

Let us denote the binary response variable from the expert labeled data as y, i.e. fraud if $y = 1$; otherwise $y = 0$. For each observation i, denote the corresponding feature set as $\boldsymbol{x}_i$. The logistic regression, one of the most natural probabilistic model is defined as $P[y_i = 1] = \frac{1}{1+\exp(-\boldsymbol{x}_i'\boldsymbol{\beta})}$, where $\boldsymbol{\beta}$ is the unknown coefficient vector. Suppose each observation i is further associated with a weight w_i. The corresponding loss function with L_k penalty on $\boldsymbol{\beta}$ [3, 6] becomes

$$\begin{aligned}\mathscr{L} &= \sum_i w_i[y_i \log(1+\exp(-\boldsymbol{x}_i'\boldsymbol{\beta})) + \\ &\quad (1-y_i)\log(1+\exp(\boldsymbol{x}_i'\boldsymbol{\beta})) + \rho\|\boldsymbol{\beta}\|_k], \qquad (1)\end{aligned}$$

where ρ is the trade-off parameter to control the shrinkage of $\boldsymbol{\beta}$ and can be estimated by cross-validation. In this paper we mainly consider $k = 2$, which is equivalent to assigning a Gaussian prior $N(\mathbf{0}, \delta\boldsymbol{I})$ on $\boldsymbol{\beta}$, where $\delta = 1/(2\rho\sum_i w_i)$.

In the case where we have no domain knowledge on w_i, it is common to assume all $w_i = 1$. We call this model **LR** (i.e. standard logistic regression). However, in scenarios when the positive rate of Y in the training data is too low (e.g. less than 1%), it is also common to up-weight the positive samples or down-sample the negatives. In this paper we assume all the positive samples have the same weight w_1, and all the negative samples have weight 1. The optimal w_1 can be determined by cross-validation in criterion of predictive performance. We call the weighted logistic regression as **WLR**. We optimize the models by minimizing the loss functions through the standard L-BFGS algorithm [4, 7].

2.2 Coefficient Bounds for Fraud Detection

It is always important to incorporate domain knowledge into the model, which can sometimes boost the model performance. In our fraud detection system, the feature set $\boldsymbol{x}$ was proposed by experts with years of experience. Currently all the features are in fact binary "rules", i.e. any violation of one rule should somehow increase the probability of fraud. However, simply fitting the model by equation (1) might generate negative coefficients on some of the features, because given limited training data, the sample size might be too small for those coefficients to converge to the right values, or when some features are highly correlated. Hence we bound the coefficients of those binary "rules" to force them to be equal or greater than 0. Specifically, we consider the following optimization problem

$$\begin{aligned}\min_{\boldsymbol{\beta}} \quad & \sum_i w_i[y_i \log(1+\exp(-\boldsymbol{x}_i'\boldsymbol{\beta})) + \\ & (1-y_i)\log(1+\exp(\boldsymbol{x}_i'\boldsymbol{\beta})) + \rho\|\boldsymbol{\beta}\|_k], \qquad (2)\end{aligned}$$

such that $\boldsymbol{\beta} \geq \boldsymbol{T}$. where for feature j, T_j is 0 in our problem. Note that T_j can also be set as other values if domain knowledge is available. When $w_i = 1$ for all i, we call the model **BLR** (bounded logistic regression). For pre-defined w_i values, we call the model **WBLR** (weighted **BLR**).

2.3 Removing the Selection Bias

As introduced in Section 1, the existing rule-based moderation system only sends the cases with high risky score for experts to review and label; hence the labeled training data is not generated through a random selection. Note that in the real world, many such online auction fraud detection systems would not allow an entirely random selection scheme to generate the training data: it is simply too expensive and ineffective. Instead usually a set of hand-tuned rules are used in the initial system to provide the basic fraud detection function. Therefore, it is important to correct the selection bias and leverage the data without labels to improve the model.

The simplest idea to remove the selection bias is to assume all events that are not labeled by the current system are defined as not fraud with a low confidence. Mathematically, denoting $\boldsymbol{z}_j$ to be the feature set for an unlabeled event j, we want to solve

$$\begin{aligned}\min_{\boldsymbol{\beta}} \quad & \sum_i w_i[y_i \log(1+\exp(-\boldsymbol{x}_i'\boldsymbol{\beta})) \\ & +(1-y_i)\log(1+\exp(\boldsymbol{x}_i'\boldsymbol{\beta})) + \rho\|\boldsymbol{\beta}\|_k] \\ & + \sum_j \tilde{w}_j[\log(1+\exp(\boldsymbol{z}_j'\boldsymbol{\beta})) + \rho\|\boldsymbol{\beta}\|_k], \qquad (3)\end{aligned}$$

such that $\boldsymbol{\beta} \geq \boldsymbol{T}$. For simplicity we could assume all $\tilde{w}_j = \tilde{w}$, i.e. all the unselected events having the same confidence to be non-

fraud. When $\tilde{w} = 0$, it implies the training data is selected randomly so no need to adjust for the selection bias. When $\tilde{w} = 1$ ($w_i = 1$ for non-fraud), it implies we hold high confidence as in expert labels that all the events that are not selected by the current system are non-fraud. Usually when $w_i = 1$, the optimal $\tilde{w}$ should be within the range $(0, 1)$, and it can be calculated through cross-validation. We call this model **WBLRRSB** (**WBLR** after removing selection bias (**RSB**)). An alternative approach of using the data without labels is to treat the responses as missing values and apply the EM algorithm [1] to fit the model. However, this approach does not take the domain knowledge that the selection of training is in fact biased; instead it purely relies on the features to fill in the missing responses iteratively. Hence in practice, it is proven to perform worse than **WBLRRSB**.

2.4 Multiple Instance Learning

When we looked into the actual expert reviewing and labeling process, we noted that the experts actually assign labels in a "bagged" fashion, i.e. for each seller id, one expert looks through all of his/her posted items, and if the expert finds any item as fraud, all of this seller id's posted items are labeled as fraud. In literature the models for this scenario are called "Multiple Instance Learning" [2, 5]. Suppose for each labeled seller i, there are K_i number of cases. For these cases, the labels should be identical, thus can be denoted as y_i. The multiple instance learning model with logistic function becomes $P[y_i = 1] = 1 - \prod_{j=1}^{K_i} \frac{1}{1+\exp(\boldsymbol{x}_j'\boldsymbol{\beta})}$, which is essentially a noisy-or likelihood function. The noisy-or likelihood function only requires a subset of the events in the bag are fraud rather than all are fraud events. The optimization problem can thus be written as

$$\begin{aligned}\min_{\boldsymbol{\beta}} \quad & \sum_i w_i[-y_i \log(1 - \prod_{j=1}^{K_i} \frac{1}{1+\exp(\boldsymbol{x}_j'\boldsymbol{\beta})}) \\ & +(1-y_i)\sum_{j=1}^{K_i} \log(1+\exp(\boldsymbol{x}_i'\boldsymbol{\beta})) + \rho K_i \|\boldsymbol{\beta}\|_k] \\ & + \sum_j \tilde{w}_j[\log(1+\exp(\boldsymbol{z}_j'\boldsymbol{\beta})) + \rho\|\boldsymbol{\beta}\|_k], \end{aligned} \tag{4}$$

such that $\boldsymbol{\beta} \geq \boldsymbol{T}$. We call this model **WBMILRSB** (weighted and bounded (**WB**) multiple instance learning (**MIL**) after removing selection bias (**RSB**)).

3. EXPERIMENTS

This section starts with experiment settings and metrics in evaluation, and then report extensive experimental results of both offline evaluations and online bucket tests to demonstrate the performance of the various proposed techniques.

3.1 Experiment Settings

We evaluated our proposed models from Section 2 using data from a major Asian online auction website which attracts a big volume of items posted for bidding every day. Our experiments consist of offline evaluations and online A/B tests. For offline evaluation, we were able to test all the models from Section 2 on two-months of log data. For online A/B test, we compared our best model **WBMILRSB** based on the offline experiments with the **Expert** model, which was used by the system with expert-crafted rules and weights.

For offline evaluations, we created our training and test data set via a bias sampling scheme from the real data set to avoid releasing the company confidential information. Our training data contains around 1M labeled cases with around 12K fraud cases from Sep 2010. Our test data also contains around 1M labeled cases with 7.6K fraud cases from Oct 2010. The number of unlabeled cases in each data is about 3M to 5M. For testing models we also obtained a sample of 446 customer fraud complaint cases in Oct 2010. The offline experiments and results are shown in Section 3.3. Note that due to the company policy concerns, we are unable to reveal what features are used in our experiments.

Model	AUC	Loss Rate of Frauds	Loss Rate of Complaints
Expert	0.724	0.00%	34.66%
LR	0.903	16.07%	55.61%
BLR	0.924	7.40%	40.56%
WBLR	0.923	4.21%	26.17%
WBLRRSB	0.922	2.83%	20.75%
WBMILRSB	0.926	2.07%	20.73%

Table 1: Offline evaluation of the model performance. The loss rate of labeled frauds and customer complaints are obtained given 100% workload rate.

3.2 Metrics for Offline Evaluation

We considered three metrics for offline model evaluation: Area under ROC curve (AUC), the loss rate of labeled frauds and the loss rate of customer complaints. The AUC is a traditional metric that people use a lot for measuring model performance. However for our scenario with labeling bias, we will show in Section 3.3 that simply using AUC might not be optimal.

Note that the training and test data were generated as follows: for each case the existing rule-based moderation system uses a human-tuned linear scoring function to determine whether to send it for expert labeling. If so, experts review it and make a fraud or non-fraud judgment; otherwise it would be considered as clean and not reviewed by anyone. Therefore, two concerns arise: a). For those cases that are not reviewed by experts, we would never know whether they are fraud or not. b). If we want to propose a new machine-learned scoring model to replace the existing one, we have to make sure it is able to catch more frauds with the same or lower expert labeling workload.

For test data, we regard the number of labeled cases as the expected 100% workload of labeling N items. For any model we could re-rank all the cases (labeled and unlabeled) by the model scores in the test set and select the first M cases with the highest scores. We call the fraction M/N the "workload rate" in the following discussion. Assume the total number of labeled frauds in test is F and the total number of reported complaints is C. For a specific workload rate such as 100%, we could count the number of labeled fraud cases F_m and the number of reported complaints C_m in the M cases. We define the loss rate of labeled frauds as $1 - F_m/F$ and the loss rate of customer complaints as $1 - C_m/C$ given the workload rate M/N. We argue model A is better than model B if given the same workload rate, both the loss rates of labeled frauds and customer complaints of model A are lower than those of B. For the **Expert** model which is exactly how the training and test data are generated 100% workload rate, the loss rate of labeled frauds becomes 0 while the loss rate of customer complaints is usually greater than 0. Therefore, these two metrics might not be appropriate when used to compare the machine-learned models with the **Expert**, but these are very useful measures to compare two machine-learned models.

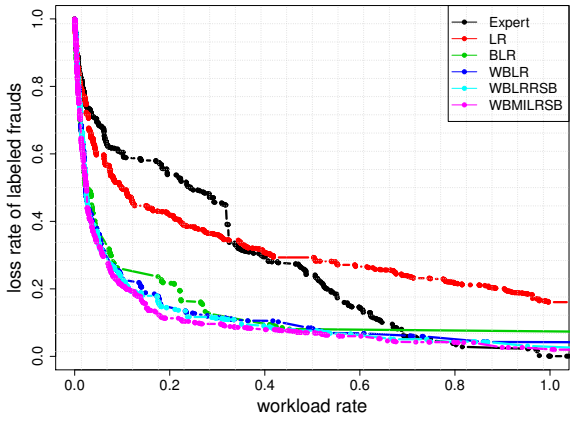

Figure 1: Offline evaluation: Workload rate versus the loss rates of labeled frauds for all the models.

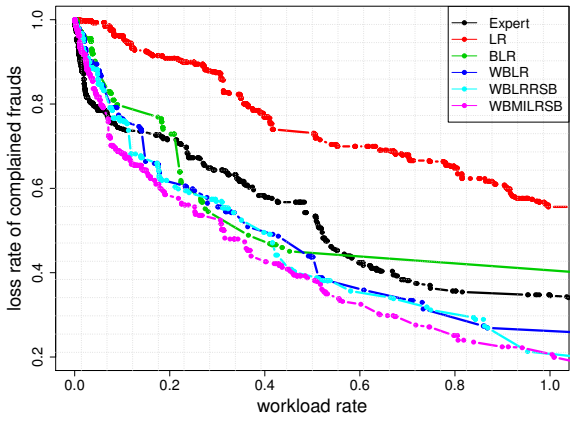

Figure 2: Offline evaluation: Workload rate versus the loss rates of customer complaints for all the models.

3.3 Offline Experiments

In this section we show the performance of the test data for 6 models: **Expert**, **LR**, **BLR**, **WBLR**, **WBLRRSB** and **WBMILRSB**. Table 1 summarizes the three metrics discussed in Section 3.2: AUC, the loss rate of labeled frauds, and the loss rate of customer complaints. The latter two are obtained given 100% workload rate. Note the loss rate of labeled frauds for the Expert model is always 0 since the test data is generated by it. Figure 1 and Figure 2 show the loss rates of the labeled frauds as well as customer complaints given different workload rates for all the models.

According to the AUC numbers, the machine-learned **LR** is much better than the human-tuned model **Expert**. It also seems obviously that the four models **BLR**, **WBLR**, **WBLRRSB** and **WBMILRSB** are better than **LR**, although among them there is very little difference when comparing AUC. However, when looking at the loss rates of labeled frauds and customer complaints, we saw a significant performance difference between **BLR** and **WBMILRSB**, and we started to doubt that **LR** and **BLR** can work well in the real auction fraud detection system since their loss rates of customer complaints look much worse than the **Expert** model! On the other hand, the numbers in the table suggest **WBMILRSB** to be the winning model that would improve the existing moderation system significantly.

Our intuition behind why AUC is not working well in our scenario is as follows: it only considers the labeled test data while ignoring the unlabeled part. In scenarios when the training and test set are biased, the AUC metric can be really misleading. On the other hand, our proposed two metrics utilize the unlabeled data and extra unbiased information such as customer complaints. This is very important to note when doing offline experiments and evaluation.

3.4 Online A/B Test

We performed online A/B test during April-May, 2011, which compared two models: our best model **WBMILRSB** and the expert-crafted **Expert** model. We used the last 30 days data for the daily training and tuned the threshold of **WBMILRSB** scores so that the workload generated by this model is roughly the same everyday. Running 4 weeks of A/B test, we observed that **WBMILRSB** significantly outperformed **Expert** by catching 25.5% more frauds and reducing the customer complaints by 15.7%, while using only 74.2% of the expert workload.

4. CONCLUSION

In this paper, we introduced various advanced machine learning techniques for real world auction fraud detection systems. By extensive offline experiments and online bucket test, we have shown our proposed model significantly outperforms the existing human-tuned rule-based system. Compared with baselines, we show that the multiple instance learning model with bounded coefficients and properly weighted observations after removing the selection bias performs the best. Hence we have pushed this model to production. For model evaluation, besides using traditional metrics such as area under ROC (AUC), we introduced two extra metrics under this fraud detection framework. We show that these metrics are more effective than AUC or ROC for distinguishing the best models.

5. REFERENCES

[1] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[2] T. Dietterich, R. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[3] A. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[4] D. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[5] V. Raykar, B. Krishnapuram, J. Bi, M. Dundar, and R. Rao. Bayesian multiple instance learning: automatic feature selection and inductive transfer. In *Proceedings of the 25th international conference on Machine learning*, pages 808–815. ACM, 2008.

[6] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[7] C. Zhu, R. Byrd, P. Lu, and J. Nocedal. L-bfgs-b: Fortran subroutines for large-scale bound constrained optimization. *ACM Transactions on Mathetmatical Software*, 23(4):550–560, 1997.