

Online Modeling of Proactive Moderation System for Auction Fraud Detection

Joel Srideep Bandi ,M.Tech Student QCET,Nellore,,joelsrideep@gmail.com

Abdul Haq Associate Professor, H.O.D Of CSE QCET,Nellore,Syedabdulhaq2203@yahoo.com

We consider the problem of building online machine-learned models for detecting auction frauds in e-commerce web sites. Since the emergence of the world wide web, online shopping and online auction have gained more and more popularity. While people are enjoying the benefits from online trading, criminals are also taking advantages to conduct fraudulent activities against honest parties to obtain illegal profit. Hence proactive fraud-detection moderation systems are commonly applied in practice to detect and prevent such illegal and fraud activities. Machine-learned models, especially those that are learned online, are able to catch frauds more efficiently and quickly than human-tuned rule-based systems. In this paper, we propose an online probit model framework which takes online feature selection, coefficient bounds from human knowledge and multiple instance learning into account simultaneously. By empirical experiments on a real-world online auction fraud detection data we show that this model can potentially detect more frauds and significantly reduce customer complaints compared to several baseline models and the human-tuned rule-based system.

Keywords

Online Auction, Fraud Detection, Online Modeling, Online Feature Selection, Multiple Instance Learning

1. INTRODUCTION

Since the emergence of the World Wide Web (WWW), electronic commerce, commonly known as e-commerce, has become more and more popular. Websites such as eBay and Amazon allow Internet users to buy and sell products and services online, which benefits everyone in terms of convenience and profitability. The traditional online shopping business model allows sellers to sell a product or service at a preset price, where buyers can choose to purchase if they find it to be a good deal. Online auction however is a different business model by which items are sold through price bidding. There is often a starting price and expiration time specified by the sellers. Once the auction starts, potential buyers bid against each other, and the winner gets the item with their highest winning bid.

Similar to any platform supporting financial transactions, online auction attracts criminals to commit fraud. The varying types of auction fraud are as follows. Products purchased by the buyer are not delivered by the seller. The delivered products do not match the descriptions that were posted by sellers. Malicious sellers may even post non-existing items with false description to deceive buyers, and request payments to be wired directly to them via bank-to-bank wire transfer. Furthermore, some criminals apply phishing techniques to steal high-rated seller's accounts so that potential buyers can be easily deceived due to their good rating. Victims of fraud transactions usually lose their money and in most cases are not recoverable. As a result, the reputation of the online auction services is hurt significantly due to fraud crimes.

To provide some assurance against fraud, E-commerce sites often provide insurance to fraud victims to cover their loss up to a certain amount. To reduce the amount of such compensations and improve their online reputation, e-commerce providers often adopt the following approaches to control and prevent fraud. The identifies of registered users are validated through email, SMS, or phone verifications. A rating system where buyers provide feedbacks is commonly used in e-commerce sites so that fraudulent sellers can be caught immediately after the first wave of buyer complaints. In addition, proactive moderation systems are built to allow human experts to manually investigate suspicious sellers or buyers. Even though e-commerce sites spend a large budget to fight frauds with a moderation system, there are still many outstanding and challenging cases. Criminals and fraudulent sellers frequently change their accounts and IP addresses to avoid being caught. Also, it is usually infeasible for human experts to investigate every buyer and seller to determine if they are committing fraud, especially when the e-commerce site attracts a lot of traffic. The patterns of fraudulent sellers often change constantly to take advantage of temporal trends. For instance, fraudulent sellers tend to sell the "hottest" products at the time to attract more potential victims. Also, whenever they find a loophole in the fraud detection system, they will immediately leverage the weakness.

In this paper, we consider the application of a proactive

moderation system for fraud detection in a major Asian online auction site, where hundreds of thousands of new auction cases are created everyday. Due to the limited expert resources, only 20%-40% of the cases can be reviewed and labeled. Therefore, it is necessary to develop an automatic pre-screening moderation system that only directs suspicious cases for expert inspection, and passes the rest as clean cases. The moderation system for this site extracts rule-based features to make decisions. The rules are created by experts to represent the suspiciousness of sellers on fraudulence, and the resulting features are often binary.¹ For instance, we can create a binary feature (rule) from the ratings of sellers, i.e. the feature value is 1 if the rating of a seller is lower than a threshold (i.e. a new account without many previous buyers); otherwise it is 0. The final moderation decision is based on the *fraud score* of each case, which is the *linear* weighted sum of those features, where the weights can be set by either human experts or machine-learned models. By deploying such a moderation system, we are capable of selecting a subset of highly suspicious cases for further expert investigation while keeping their workload at a reasonable level.

The moderation system using machine-learned models is proven to improve fraud detection significantly over the human-tuned weights [38]. In [38] the authors considered the scenario of building offline models by using the previous 30 days data to serve the next day. Since the response is binary (fraud or non-fraud) and the scoring function has to be linear, logistic regression is used. The authors have shown that applying expert knowledge, such as bounding the rule-based feature weights to be positive and multiple-instance learning, can significantly improve the performance in terms of detecting more frauds and reducing customer complaints given the same workload from human experts. However, offline models often meet the following challenges: (a) Since the auction fraud rate is generally very low ($< 1\%$), the data becomes quite imbalanced and it is well-known that in such scenario even fitting simple logistic regression becomes a difficult problem [27]. Therefore, unless we use a large amount of historical training data, offline models tend to be fairly unstable. For example, in [38], 30 days of training data with around 5 million samples are used for the daily update of the model. Hence it practically adds a lot of computation and memory load for each batch update, compared to online models. (b) Since the fraudulent sellers change their pattern very fast, it requires the model to also evolve dynamically. However, for offline models it is often non-trivial to address such needs.

Once a case is determined as fraudulent, all the cases from this seller will be suspended immediately. Therefore smart fraudulent sellers tend to change their patterns quickly to avoid being caught; hence some features that are effective today might turn out to be not important tomorrow, or vice versa. Also, since the training data is from human labeling, the high cost makes it almost impossible to obtain a very large sample. Therefore for such systems (i.e. relatively small sample size with many features with temporal pattern), online feature selection is often required to provide good performance. Human experts are also willing to see the results of online feature selection to monitor the ef-

fectiveness of the current set of features, so that they can understand the pattern of frauds and further add or remove some features.

Our contribution. In this paper we study the problem of building online models for the auction fraud detection moderation system, which essentially evolves dynamically over time. We propose a Bayesian probit online model framework for the binary response. We apply the stochastic search variable selection (SSVS) [16], a well-known technique in statistical literature, to handle the dynamic evolution of the feature importance in a principled way. Note that we are not aware of any previous work that tries to embed SSVS into online modeling. Similar to [38], we consider the expert knowledge to bound the rule-based coefficients to be positive. Finally, we consider to combine this online model with multiple instance learning [30] that gives even better empirical performance. We report the performance of all the above models through extensive experiments using fraud detection datasets from a major online auction website in Asia.

The paper is organized as follows. In Section 2 we first summarize several specific features of the application and describe our online modeling framework with fitting details. We review the related work in literature in Section 3. In Section 4 we show the experimental results that compare all the models proposed in this paper and several simple baselines. Finally, we conclude and discuss future work in Section 5.

2. OUR METHODOLOGY

Our application is to detect online auction frauds for a major Asian site where hundreds of thousands of new auction cases are posted every day. Every new case is sent to the proactive anti-fraud moderation system for pre-screening to assess the risk of being fraud. The current system is featured by:

- **Rule-based features:** Human experts with years of experience created many rules to detect whether a user is fraud or not. An example of such rules is “blacklist”, i.e. whether the user has been detected or complained as fraud before. Each rule can be regarded as a binary feature that indicates the fraud likeliness.
- **Linear scoring function:** The existing system only supports linear models. Given a set of coefficients (weights) on features, the fraud score is computed as the weighted sum of the feature values.
- **Selective labeling:** If the fraud score is above a certain threshold, the case will enter a queue for further investigation by human experts. Once it is reviewed, the final result will be labeled as boolean, i.e. fraud or clean. Cases with higher scores have higher priorities in the queue to be reviewed. The cases whose fraud score are below the threshold are determined as clean by the system without any human judgment.
- **Fraud churn:** Once one case is labeled as fraud by human experts, it is very likely that the seller is not trustable and may be also selling other frauds; hence all the items submitted by the same seller are labeled as fraud too. The fraudulent seller along with his/her cases will be removed from the website immediately once detected.

- **User feedback:** Buyers can file complaints to claim loss if they are recently deceived by fraudulent sellers.

Motivated by these specific attributes in the moderation system for fraud detection, in this section we describe our Bayesian online modeling framework with details of model fitting via Gibbs sampling. We start from introducing the online probit regression model in Section 2.1. In Section 2.2 we apply stochastic search variable selection (SSVS), a well-known technique in statistics literature, to the online probit regression framework so that the feature importance can dynamically evolve over time. Since it is important to use the expert knowledge, as in [38], we describe how to bound the coefficients to be positive in Section 2.3, and finally combine our model with multiple instance learning in Section 2.4.

2.1 Online Probit Regression

Consider splitting the continuous time into many equal-size intervals. For each time interval we may observe multiple expert-labeled cases indicating whether they are fraud or non-fraud. At time interval t suppose there are n_t observations. Let us denote the i -th binary observation as y_{it} . If $y_{it} = 1$, the case is fraud; otherwise it is non-fraud. Let the feature set of case i at time t be \mathbf{x}_{it} . The probit model [3] can be written as

$$P[y_{it} = 1 | \mathbf{x}_{it}, \beta_t] = \Phi(\mathbf{x}'_{it}\beta_t), \tag{1}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution $N(0, 1)$, and β_t is the unknown regression coefficient vector at time t .

Through data augmentation the probit model can be expressed in a hierarchical form as follows: For each observation i at time t assume a latent random variable z_{it} . The binary response y_{it} can be viewed as an indicator of whether $z_{it} > 0$, i.e. $y_{it} = 1$ if and only if $z_{it} > 0$. If $z_{it} \leq 0$, then $y_{it} = 0$. z_{it} can then be modeled by a linear regression

$$z_{it} \sim N(\mathbf{x}'_{it}\beta_t, 1). \tag{2}$$

In a Bayesian modeling framework it is common practice to put a Gaussian prior on β_t ,

$$\beta_t \sim N(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), \tag{3}$$

where $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ are prior mean and prior covariance matrix respectively.

Model fitting. Since the posterior $\pi(\beta_t | \mathbf{y}_t, \mathbf{x}_t, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ does not have a closed form, this model is fitted by using the latent vector \mathbf{z}_t through Gibbs sampling. For each iteration we first sample $(\mathbf{z}_t | \mathbf{y}_t, \mathbf{x}_t, \beta_t)$ and then sample $(\beta_t | \mathbf{z}_t, \mathbf{y}_t, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$. Specifically, for each observation i at time t , sample

$$\pi(z_{it} | y_{it} = 1, \mathbf{x}_{it}, \beta_t) \sim N(\mathbf{x}'_{it}\beta_t, 1), \tag{4}$$

truncated by 0 as lower bound. And

$$\pi(z_{it} | y_{it} = 0, \mathbf{x}_{it}, \beta_t) \sim N(\mathbf{x}'_{it}\beta_t, 1), \tag{5}$$

truncated by 0 as upper bound. Then sample

$$\pi(\beta_t | \mathbf{z}_t, \mathbf{y}_t, \mathbf{x}_t) = (\beta_t | \mathbf{z}_t, \mathbf{x}_t) \sim N(\hat{\mathbf{m}}_t, \hat{\mathbf{V}}_t), \tag{6}$$

where

$$\hat{\mathbf{V}}_t = (\boldsymbol{\Sigma}_t^{-1} + \mathbf{x}'_t \mathbf{x}_t)^{-1}, \hat{\mathbf{m}}_t = \hat{\mathbf{V}}_t (\boldsymbol{\Sigma}_t^{-1} \boldsymbol{\mu}_t + \mathbf{x}'_t \mathbf{z}_t). \tag{7}$$

By iterative sampling the conditional posterior of \mathbf{z}_t and β_t for N iterations plus B number of burn-in samples (in our experiments we let $N = 10000$ and $B = 1000$), we can obtain

N posterior samples of β_t . We can thus obtain the posterior sample mean $\hat{\boldsymbol{\mu}}_t$ and sample covariance $\hat{\boldsymbol{\Sigma}}_t$, to serve as the posterior mean and covariance for β_t respectively.

Online modeling. At time t , given the prior of β_t as $N(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ and the observed data, by Gibbs sampling we obtain the posterior of $\pi(\beta_t | \mathbf{y}_t, \mathbf{x}_t, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \sim N(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\Sigma}}_t)$. At time $t+1$, the parameters of the prior of β_{t+1} can be written as

$$\boldsymbol{\mu}_{t+1} = \hat{\boldsymbol{\mu}}_t, \boldsymbol{\Sigma}_{t+1} = \hat{\boldsymbol{\Sigma}}_t / \delta, \tag{8}$$

where $\delta \in (0, 1]$ is a tuning parameter that allows the model to evolve dynamically. When $\delta = 1$, the model updates by treating all historical observations equally (i.e. no “forgetting”). When $\delta < 1$, the influence of the data observed k batches ago decays in the order of $O(\delta^k)$, i.e. the smaller δ is, the more dynamic the model becomes. δ can be learned via cross-validation. This online modeling technique has been commonly used in literature (see [1] and [37] for example). In practice, for simplicity we let $\boldsymbol{\Sigma}_0 = \sigma_0^2 \mathbf{I}$ and $\boldsymbol{\Sigma}_{t+1} = \text{diag}(\hat{\boldsymbol{\Sigma}}_t) / \delta$ to ignore the covariance among the coefficients of β_t .

Besides the probit link function used in this paper, another common link function for the binary response is logistic [26]. Although logistic regression seems more often used in practice, there does not exist a conjugate prior for the coefficient β_t hence the posterior of β_t always does not have a closed form; therefore approximation is commonly applied (e.g. [20]). Probit model through data augmentation, on the other hand, allows us to sample the posterior of β_t through Gibbs sampling without any approximation. It also allows us to plug-in more complicated techniques such as SSVS conveniently.

2.2 Online Feature Selection through SSVS

For regression problems with many features, proper shrinkage on the regression coefficients is usually required to avoid over-fitting. For instance, two common shrinkage methods are L2 penalty (ridge regression) and L1 penalty (Lasso) [33]. Also, experts often want to monitor the importance of the rules so that they can make appropriate adjustments (e.g. change rules or add new rules). However, the fraudulent sellers change their behavioral pattern quickly: Some rule-based feature that does not help today might help a lot tomorrow. Therefore it is necessary to build an online feature selection framework that evolves dynamically to provide both optimal performance and intuition. In this paper we embed the stochastic search variable selection (SSVS) [16] into the online probit regression framework described in Section 2.1.

At time t , let β_{jt} be the j -th element of the coefficient vector β_t . Instead of putting a Gaussian prior on β_{jt} , the prior of β_{jt} now is

$$\beta_{jt} \sim p_{0jt} 1(\beta_{jt} = 0) + (1 - p_{0jt}) N(\mu_{jt}, \sigma_{jt}^2), \tag{9}$$

where p_{0jt} is the prior probability of β_{jt} being exactly 0, and with prior probability $1 - p_{0jt}$, β_{jt} is drawn from a Gaussian distribution with mean μ_{jt} and variance σ_{jt}^2 . Such prior is called the “spike and slab” prior in the literature [19] but how to embed it to online modeling has never been explored before.

Model fitting. Let $\beta_{-j,t}$ be the vector β_t excluding β_{jt} . The model fitting procedure for this model is again through Gibbs sampling since the conditional posterior $\pi(\mathbf{z}_t | \mathbf{y}_t, \mathbf{x}_t, \beta_t)$

and $\pi(\beta_{jt}|\beta_{-j,t}, \mathbf{z}_t, \mathbf{y}_t, p_{0jt}, \mu_{jt}, \sigma_{jt})$ have closed form. Specifically, sampling z_{it} for observation i at time t has the same formula as in Section 2.1.

$$\pi(z_{it}|y_{it} = 1, \mathbf{x}_{it}, \beta_t) \sim N(\mathbf{x}'_{it}\beta_t, 1), \tag{10}$$

truncated by 0 as lower bound. And

$$\pi(z_{it}|y_{it} = 0, \mathbf{x}_{it}, \beta_t) \sim N(\mathbf{x}'_{it}\beta_t, 1), \tag{11}$$

truncated by 0 as upper bound.

Denote

$$\tilde{z}_{it} = z_{it} - \sum_{k \neq j} x_{ikt}\beta_{kt}. \tag{12}$$

We also let function $\text{dnorm}(x; m, V)$ be the density function of Gaussian distribution $N(m, V)$, i.e.

$$\text{dnorm}(x; m, V) = \frac{1}{\sqrt{2\pi V}} \exp\left(-\frac{(x-m)^2}{2V}\right). \tag{13}$$

To sample $\pi(\beta_{jt}|\beta_{-j,t}, \mathbf{z}_t, \mathbf{y}_t, p_{0jt}, \mu_{jt}, \sigma_{jt})$,

$$\begin{aligned} & \pi(\beta_{jt}|\beta_{-j,t}, \mathbf{z}_t, \mathbf{y}_t, p_{0jt}, \mu_t, \Sigma_t) \\ &= \pi(\beta_{jt}|\beta_{-j,t}, \mathbf{z}_t, p_{0jt}, \mu_t, \Sigma_t) \\ &\propto \left[\prod_{i=1}^{n_t} \exp\left(-\frac{(\tilde{z}_{it} - x_{ijt}\beta_{jt})^2}{2}\right) \right] \\ & \quad [p_{0jt} \mathbf{1}(\beta_{jt} = 0) + \frac{1 - p_{0jt}}{\sqrt{2\pi\sigma_{jt}^2}} \exp\left(-\frac{(\beta_{jt} - \mu_{jt})^2}{2\sigma_{jt}^2}\right)] \\ &\propto \hat{\gamma}_{jt} \mathbf{1}(\beta_{jt} = 0) + (1 - \hat{\gamma}_{jt})N(\hat{m}_{jt}, \hat{V}_{jt}), \end{aligned} \tag{14}$$

where

$$\hat{V}_{jt} = (\sigma_{jt}^{-2} + \mathbf{x}'_{jt}\mathbf{x}_{jt})^{-1}, \tag{15}$$

$$\hat{m}_{jt} = \hat{V}_{jt}(\mathbf{x}'_{jt}\tilde{\mathbf{z}}_t + \frac{\mu_{jt}}{\sigma_{jt}^2}), \tag{16}$$

$$\hat{\gamma}_{jt} = \frac{p_{0jt}}{p_{0jt} + (1 - p_{0jt}) \frac{\text{dnorm}(0; \mu_{jt}, \sigma_{jt}^2)}{\text{dnorm}(0; \hat{m}_{jt}, \hat{V}_{jt})}}. \tag{17}$$

Since the conditional posterior $\pi(\beta_{jt}|\beta_{-j,t}, \mathbf{z}_t, \mathbf{y}_t, p_{0jt}, \mu_{jt}, \sigma_{jt}) \sim \hat{\gamma}_{jt} \mathbf{1}(\beta_{jt} = 0) + (1 - \hat{\gamma}_{jt})N(\hat{m}_{jt}, \hat{V}_{jt})$, it implies that to sample β_{jt} we first flip a coin with probability of head equal to $\hat{\gamma}_{jt}$. If it is head, we let $\beta_{jt} = 0$; otherwise we sample β_{jt} from $N(\hat{m}_{jt}, \hat{V}_{jt})$.

After B burn-in samples for convergence purpose, denote the collected k -th posterior sample of β_{jt} as $\beta_{jt}^{(k)}$, $k = 1, \dots, N$. We estimate the posterior distribution of $\pi(\beta_{jt}|\mathbf{y}_t, p_{0jt}, \mu_{jt}, \sigma_{jt})$ by

$$\pi(\beta_{jt}|\mathbf{y}_t, p_{0jt}, \mu_{jt}, \sigma_{jt}) \sim \hat{p}_{jt} \mathbf{1}(\beta_{jt} = 0) + (1 - \hat{p}_{jt})N(\hat{\mu}_{jt}, \hat{\sigma}_{jt}^2), \tag{18}$$

where

$$\hat{p}_{jt} = \sum_{k=1}^N \mathbf{1}(\beta_{jt}^{(k)} = 0) / N, \tag{19}$$

$$\hat{\mu}_{jt} = \sum_{k=1}^N \beta_{jt}^{(k)} / \sum_{k=1}^N \mathbf{1}(\beta_{jt}^{(k)} \neq 0), \tag{20}$$

$$\hat{\sigma}_{jt}^2 = \frac{\sum_{k=1}^N \mathbf{1}(\beta_{jt}^{(k)} \neq 0) (\beta_{jt}^{(k)} - \hat{\mu}_{jt})^2}{\sum_{k=1}^N \mathbf{1}(\beta_{jt}^{(k)} \neq 0) - 1}, \tag{21}$$

Online modeling. When $t = 0$, we could set $p_{0j0} = 0.5$ for all j , i.e. before observing any data we consider the probability of the j -th feature coefficient being zero or non-zero is equal. At time $t + 1$, we let

$$p_{0j(t+1)} = \omega \hat{p}_{jt} + 0.5(1 - \omega), \tag{22}$$

$$\mu_{j(t+1)} = \hat{\mu}_{jt}, \sigma_{j(t+1)}^2 = \hat{\sigma}_{jt}^2 / \delta, \tag{23}$$

where $\omega \in (0, 1)$ and $\delta \in (0, 1]$ are both tuning parameters. Although it seems more natural to let $p_{0j(t+1)} = \hat{p}_{jt}$, note that in practice we often see \hat{p}_{jt} becomes 1 or 0 even though N is large (say 10000), which implies that there are some features which are very important (i.e. $\hat{p}_{jt} = 1$) or can be excluded from the model to avoid over-fitting (i.e. $\hat{p}_{jt} = 0$). In such scenario, simply letting $p_{0j(t+1)} = \hat{p}_{jt}$ will make the posterior $p_{j(\hat{t}+1)}$ be 1 or 0 again regardless of what data is observed at time $t + 1$, and so for all the latter batches. Therefore, to allow the feature importance indicator $p_{j(\hat{t}+1)}$ to evolve by using both the observed data at time $t + 1$ and the prior knowledge learned before time $t + 1$, it is important to let $p_{0j(t+1)}$, the prior probability for time $t + 1$, to drift slightly away from \hat{p}_{jt} towards the initial prior belief (i.e. $p_{0j0} = 0.5$). Intuitively, the value of ω controls how much we “forget” the prior knowledge: the smaller ω is, the more dynamic the model becomes. In practice we can tune both ω and δ via cross-validation.

2.3 Coefficient Bounds

Incorporating expert domain knowledge into the model is often important and has been proved to boost the model performance (see [38] for instance). In our moderation system, the feature set \mathbf{x} is proposed by experts with years of experience in detecting auction frauds. Most of these features are in fact “rules”, i.e., any violation of one rule should ideally increase the probability of the seller being fraud to some extent. A simple example of such rules is the “blacklist”, i.e. whether the seller has ever been detected or complained as fraud before. However, for some of such rules simply applying probit regression as described in Section 2.1 or logistic regression as in [38] might give negative coefficients, because given limited training data the sample size might be too small for those coefficients to converge to right values, or it can be because of the high correlation among the features. Hence we bound the coefficients of the features that are in fact binary rules, to force them to be either positive or equal to 0. Note that this approach couples very well with the SSVS described in Section 2.2: all the coefficients which were negative are now pushed towards zero.

Suppose feature j is a binary rule and we wish to bound its coefficients to be greater than or equal to 0. At time t , the prior of β_{jt} now becomes

$$\beta_{jt} \sim p_{0jt} \mathbf{1}(\beta_{jt} = 0) + (1 - p_{0jt})N(\mu_{jt}, \sigma_{jt}^2) \mathbf{1}(\beta_{jt} > 0), \tag{24}$$

where $N(\mu_{jt}, \sigma_{jt}^2) \mathbf{1}(\beta_{jt} > 0)$ means β_{jt} is sampled from $N(\mu_{jt}, \sigma_{jt}^2)$, truncated by 0 as lower bound.

Model fitting. For observation i at time t , the sampling step for z_{it} is the same as Section 2.1 and 2.2. To sample

$$\begin{aligned} &\pi(\beta_{jt}|\beta_{-j,t}, \mathbf{z}_t, \mathbf{y}_t, p_{0jt}, \mu_{jt}, \sigma_{jt}), \\ &\pi(\beta_{jt}|\beta_{-j,t}, \mathbf{z}_t, \mathbf{y}_t, p_{0jt}, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \\ &\propto \left[\prod_{i=1}^{n_t} \exp\left(-\frac{(\tilde{z}_{it} - x_{ijt}\beta_{jt})^2}{2}\right) \right] \\ &\left[p_{0jt}1(\beta_{jt} = 0) + \frac{1 - p_{0jt}}{\sqrt{2\pi\sigma_{jt}^2}} \exp\left(-\frac{(\beta_{jt} - \mu_{jt})^2}{2\sigma_{jt}^2}\right)1(\beta_{jt} > 0) \right] \\ &\propto \hat{\gamma}_{jt}1(\beta_{jt} = 0) + (1 - \hat{\gamma}_{jt})N(\hat{m}_{jt}, \hat{V}_{jt})1(\beta_{jt} > 0), \end{aligned} \tag{25}$$

where

$$\hat{V}_{jt} = (\sigma_{jt}^{-2} + \mathbf{x}'_{jt}\mathbf{x}_{jt})^{-1}, \tag{26}$$

$$\hat{m}_{jt} = \hat{V}_{jt}(\mathbf{x}'_{jt}\tilde{\mathbf{z}}_t + \frac{\mu_{jt}}{\sigma_{jt}^2}), \tag{27}$$

$$\hat{\gamma}_{jt} = \frac{p_{0jt}}{p_{0jt} + (1 - p_{0jt}) \frac{\Phi(\hat{m}_{jt}/\sqrt{\hat{V}_{jt}}) \text{dnorm}(0; \mu_{jt}, \sigma_{jt}^2)}{\Phi(\mu_{jt}/\sigma_{jt}) \text{dnorm}(0; \hat{m}_{jt}, \hat{V}_{jt})}}. \tag{28}$$

After B number of burn-in samples we collect N posterior samples of β_{jt} . Denote $\beta_{jt}^{(k)}$ as the k -th sample. Similar to Section 2.2,

$$\begin{aligned} &\pi(\beta_{jt}|\mathbf{y}_t, p_{0jt}, \mu_{jt}, \sigma_{jt}) \\ &\sim \hat{p}_{jt}1(\beta_{jt} = 0) + (1 - \hat{p}_{jt})N(\hat{\mu}_{jt}, \hat{\sigma}_{jt}^2)1(\beta_{jt} > 0), \end{aligned} \tag{29}$$

where

$$\hat{p}_{jt} = \sum_{k=1}^N 1(\beta_{jt}^{(k)} = 0)/N. \tag{30}$$

The estimated values of $\hat{\mu}_{jt}$ and $\hat{\sigma}_{jt}^2$ actually can not be obtained directly from the posterior sample mean and variance for the non-zero samples. Since it is a truncated normal distribution and non-symmetric, the mean of the non-zero posterior samples tends to be higher than the real value of μ_{jt} . Let $q_{jt} = \sum_{k=1}^N 1(\beta_{jt}^{(k)} \neq 0)$, we find $\hat{\mu}_{jt}$ and $\hat{\sigma}_{jt}^2$ via maximizing the density function

$$\mathcal{L} = (2\pi\hat{\sigma}_{jt}^2\Phi^2(\hat{\mu}_{jt}/\hat{\sigma}_{jt}))^{-\frac{q_{jt}}{2}} \exp\left(-\frac{\sum_{k=1}^N (\beta_{jt}^{(k)} - \hat{\mu}_{jt})^2 1(\beta_{jt}^{(k)} \neq 0)}{2\hat{\sigma}_{jt}^2}\right). \tag{31}$$

We find the optimal solution to equation (31) by alternatively fitting $(\hat{\mu}_{jt}|\hat{\sigma}_{jt}^2)$ and $(\hat{\sigma}_{jt}^2|\hat{\mu}_{jt})$ to maximize the function using [6].

The online modeling component is the same as that in Section 2.2.

2.4 Multiple Instance Learning

When we look at the procedure of expert labeling in the moderation system, we noticed that experts do the labeling in a “bagged” fashion: i.e. when a new labeling process starts, an expert picks the most “suspicious” seller in the queue and looks through all of his/her cases posted in the current batch (e.g. this day); if the expert determines any of the cases to be fraud, then all of the cases from this seller are labeled as fraud. In literature the models to handle such scenario are called “multiple instance learning” [30]. Suppose for each seller i at time t there are K_{it} number of cases. For

all the K_{it} cases the labels should be identical, hence can be denoted as y_{it} . For probit link function, through data augmentation denote the latent variable for the l -th case of seller i as z_{ilt} . the multiple instance learning model can be written as

$$y_{it} = 0 \text{ iff } z_{ilt} < 0, \forall l = 1, \dots, K_{it}; \tag{32}$$

otherwise $y_{it} = 1$, and

$$z_{ilt} \sim N(\mathbf{x}'_{ilt}\boldsymbol{\beta}_t, 1), \tag{33}$$

where $\boldsymbol{\beta}_t$ can have any types of priors that are described in Section 2.1 (Gaussian), Section 2.2 (spike and slab), and Section 2.3 (spike and slab with bounds).

Model fitting. The model fitting procedure via Gibbs sampling is very similar to those in the previous sections. While the process of sampling the conditional posterior of $\boldsymbol{\beta}_t$ remains the same, the process of sampling $\pi(\mathbf{z}_t|\mathbf{y}_t, \mathbf{x}_t, \boldsymbol{\beta}_t)$ is different. For seller i at time t ,

$$\pi(z_{ilt}|y_{it} = 0, \mathbf{x}_{ilt}, \boldsymbol{\beta}_t) \sim N(\mathbf{x}'_{ilt}\boldsymbol{\beta}_t, 1), \tag{34}$$

truncated by 0 as upper bound for all $l = 1, \dots, K_{it}$. If $y_{it} = 1$, it implies at least one of the $z_{ilt} > 0$ for all $l = 1, \dots, K_{it}$. We construct pseudo label \tilde{y}_{ilt} such that $\tilde{y}_{ilt} = 0$ if $z_{ilt} < 0$; otherwise $\tilde{y}_{ilt} = 1$. The density

$$\pi(\tilde{y}_{1it}, \dots, \tilde{y}_{iK_{it}t}|y_{it} = 1, \mathbf{x}_{it}, \boldsymbol{\beta}_t) \tag{35}$$

$$= \frac{\prod_{l=1}^{K_{it}} (\Phi(\mathbf{x}'_{ilt}\boldsymbol{\beta}_t))^{\tilde{y}_{ilt}} (1 - \Phi(\mathbf{x}'_{ilt}\boldsymbol{\beta}_t))^{1-\tilde{y}_{ilt}}}{\sum_{\sum_{l=1}^{K_{it}} \tilde{y}_{ilt} > 0} \prod_{l=1}^{K_{it}} (\Phi(\mathbf{x}'_{ilt}\boldsymbol{\beta}_t))^{\tilde{y}_{ilt}} (1 - \Phi(\mathbf{x}'_{ilt}\boldsymbol{\beta}_t))^{1-\tilde{y}_{ilt}}}.$$

To sample z_{ilt} when $y_{it} = 1$, we first sample \tilde{y}_{ilt} for all $l = 1, \dots, K_{it}$ using Equation (35). Then we sample z_{ilt} by

$$\pi(z_{ilt}|y_{it} = 1, \tilde{y}_{ilt} = 1, \mathbf{x}_{ilt}, \boldsymbol{\beta}_t) \sim N(\mathbf{x}'_{ilt}\boldsymbol{\beta}_t, 1), \tag{36}$$

truncated by 0 as lower bound. And

$$\pi(z_{ilt}|y_{it} = 1, \tilde{y}_{ilt} = 0, \mathbf{x}_{ilt}, \boldsymbol{\beta}_t) \sim N(\mathbf{x}'_{ilt}\boldsymbol{\beta}_t, 1), \tag{37}$$

truncated by 0 as upper bound.

The estimation of the posterior of $\boldsymbol{\beta}_t$ and the online modeling component are the same as those in the previous sections.

3. RELATED WORK

Online auction fraud is always recognized as an important issue. There are articles on websites to teach people how to avoid online auction fraud (e.g. [35, 14]). [10] categorizes auction fraud into several types and proposes strategies to fight them. Reputation systems are used extensively by websites to detect auction frauds, although many of them use naive approaches. [31] summarized several key properties of a good reputation system and also the challenges for the modern reputation systems to elicit user feedback. Other representative work connecting reputation systems with online auction fraud detection include [32, 17, 28], where the last work [28] introduced a Markov random field model with a belief propagation algorithm for the user reputation.

Other than reputation systems, machine learned models have been applied to moderation systems for monitoring and detecting fraud. [7] proposed to train simple decision trees to select good sets of features and make predictions. [23]

developed another simple approach that uses social network analysis and decision trees. [38] proposed an offline logistic regression modeling framework for the auction fraud detection moderation system which incorporates domain knowledge such as coefficient bounds and multiple instance learning.

In this paper we treat the fraud detection problem as a binary classification problem. The most frequently used models for binary classification include logistic regression [26], probit regression [3], support vector machine (SVM) [12] and decision trees [29]. Feature selection for regression models is often done through introducing penalties on the coefficients. Typical penalties include ridge regression [34] (L2 penalty) and Lasso [33] (L1 penalty). Compared to ridge regression, Lasso shrinks the unnecessary coefficients to zero instead of small values, which provides both intuition and good performance. Stochastic search variable selection (SSVS) [16] uses “spike and slab” prior [19] so that the posterior of the coefficients have some probability being 0. Another approach is to consider the variable selection problem as model selection, i.e. put priors on models (e.g. a Bernoulli prior on each coefficient being 0) and compute the marginal posterior probability of the model given data. People then either use Markov Chain Monte Carlo to sample models from the model space and apply Bayesian model averaging [36], or do a stochastic search in the model space to find the posterior mode [18]. Among non-linear models, tree models usually handles the non-linearity and variable selection simultaneously. Representative work includes decision trees [29], random forests [5], gradient boosting [15] and Bayesian additive regression trees (BART) [8].

Online modeling (learning) [4] considers the scenario that the input is given one piece at a time, and when receiving a batch of input the model has to be updated according to the data and make predictions and servings for the next batch. The concept of online modeling has been applied to many areas, such as stock price forecasting (e.g. [22]), web content optimization [1], and web spam detection (e.g. [9]). Compared to offline models, online learning usually requires much lighter computation and memory load; hence it can be widely used in real-time systems with continuous support of inputs. For online feature selection, representative applied work include [11] for the problem of object tracking in computer vision research, and [21] for content-based image retrieval. Both approaches are simple while in this paper the embedding of SSVS to the online modeling is more principled.

Multiple instance learning, which handles the training data with bags of instances that are labeled positive or negative, is originally proposed by [13]. Many papers has been published in the application area of image classification such as [25, 24]. The logistic regression framework of multiple instance learning is presented in [30], and the SVM framework is presented in [2].

4. EXPERIMENTS

We conduct our experiments on a real online auction fraud detection data set collected from a major Asian website. We consider the following online models:

- **ON-PROB** is the online probit regression model described in Section 2.1.
- **ON-SSVSB** is the online probit regression model with

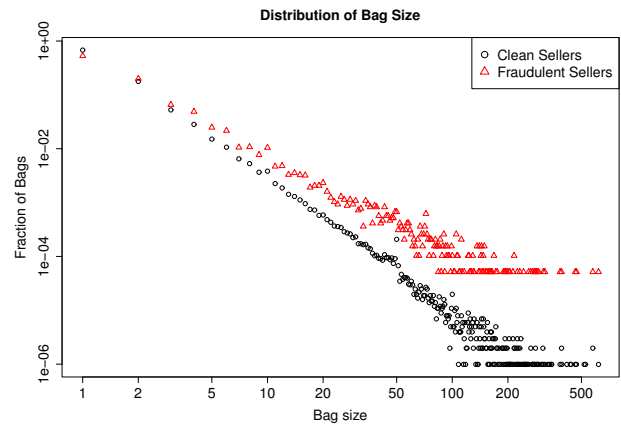


Figure 1: Fraction of bags versus the number of cases per bag (“bag size”) submitted by fraudulent and clean sellers respectively. A bag contains all the cases submitted by a seller in the same day.

“spike and slab” prior on the coefficients, and the coefficients for the binary rule features are bounded to be positive (see Section 2.2 and 2.3).

- **ON-SSVSBMIL** is the online probit regression model with multiple instance learning and “spike and slab” prior on the coefficients. The coefficients for the binary rule features are also bounded to be positive (Section 2.4).

For all the above online models we ran 10000 iterations plus 1000 burn-ins to guarantee the convergence of the Gibbs sampling.

We compare the online models with a set of offline models that are similar to [38]. For observation i , we denote the binary response as y_i and the feature set as \mathbf{x}_i . For multiple instance learning purpose we assume seller i has K_i cases and denote the feature set for each case l as \mathbf{x}_{il} . The offline models are

- **Expert** has the human-tuned coefficients set by domain experts based on their knowledge and recent fraud-fighting experience.

- **OF-LR** is the offline logistic regression model that minimizes the loss function

$$\mathcal{L} = \sum_i y_i \log(1 + \exp(-\mathbf{x}'_i \boldsymbol{\beta})) + (1 - y_i) \log(1 + \exp(\mathbf{x}'_i \boldsymbol{\beta})) + \rho \|\boldsymbol{\beta}\|_2, \quad (38)$$

where ρ is the tuning L2 penalty parameter that can be learned by cross-validation.

- **OF-MIL** is the offline logistic regression with multiple instance learning that optimizes the loss function

$$\mathcal{L} = \sum_i -y_i \log\left(1 - \prod_{l=1}^{K_i} \frac{1}{1 + \exp(\mathbf{x}'_{il} \boldsymbol{\beta})}\right) + (1 - y_i) \sum_{l=1}^{K_i} \log(1 + \exp(\mathbf{x}'_{il} \boldsymbol{\beta})) + \rho \|\boldsymbol{\beta}\|_2. \quad (39)$$

Model	Rate of Missed Complaints	Batch Size	Best δ
Expert	0.3466	–	–
OF-LR	0.4479	–	–
OF-MIL	0.3149	–	–
OF-BMIL	0.2439	–	–
ON-PROB	0.2483	Day	0.7
ON-SSVSB	0.1863	Day	0.7
ON-SSVSBMIL	0.1620	Day	0.7
ON-SSVSBMIL	0.1330	1/2 Day	0.8
ON-SSVSBMIL	0.1508	1/4 Day	0.9
ON-SSVSBMIL	0.1581	1/8 Day	0.95

Table 1: The rates of missed customer complaints for all the models given 100% workload rate.

- **OF-BMIL** is the bounded offline logistic regression with multiple instance learning that optimizes the loss function in (39) such that $\beta \geq T$, where T is the pre-specified vector of lower bounds (i.e. for feature j , $T_j = 0$ if we force its weight to be non-negative; otherwise $T_j = -\infty$).

All the above offline models can be fitted via the standard L-BFGS algorithm [39].

This section is organized as follows. In Section 4.1 we first introduce the data and describe the general settings of the models. In Section 4.2 we describe the evaluation metric for this experiment: the rate of missed customer complaints. Finally we show the performance of all the models in Section 4.3 with detailed discussion.

4.1 The Data and Model Setting

Our application is a real fraud moderation and detection system designed for a major Asian online auction website that attracts hundreds of thousands of new auction postings every day. The data consist of around 2M expert labeled auction cases with $\sim 20K$ of them labeled as fraud during September and October 2010. Besides the labeled data we also have unlabeled cases which passed the “pre-screening” of the moderation system (using the Expert model). The number of unlabeled cases in the data is about 6M-10M. For each observation there is a set of features indicating how “suspicious” it is. To avoid future fraudulent sellers gaming around our system, the exact number and format of these features are highly confidential and can not be released. Besides the expert-labeled binary response, the data also contains a list of customer complaints every day, filed by the victims of the fraud. Our data in October 2010 contains a sample of around 500 customer complaints.

As described in Section 2, human experts often label cases in a “bagged” way, i.e. at any point of time they select the current most “suspicious” seller in the system and examine all of his/her cases posted on that day. If any of these cases is fraud, all of this seller’s cases will be labeled as fraud. Therefore we put all the cases submitted by a seller in the same day into a bag. In Figure 1 we show the distribution of the bag size posted by fraudulent and clean sellers respectively. From the figure we do see that there are some proportion of sellers selling more than one item in a day, and the number of bags (sellers) decays exponentially as the bag size increases. This indicates that applying multiple

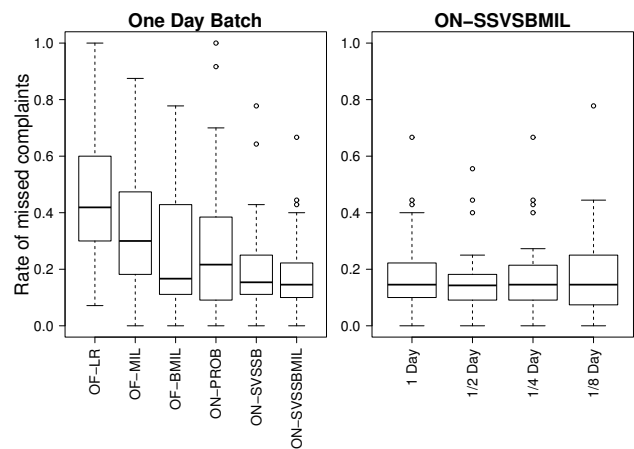


Figure 2: The boxplots of the rates of missed customer complaints on a daily basis for all the offline and online models. It is obtained given 100% workload rate.

instance learning can be useful for this data. It is also interesting to see that the fraudulent sellers tend to post more auction cases than the clean sellers, since it potentially leads to higher illegal profit.

We conduct our experiments for the offline models OF-LR, OF-MIL and OF-BMIL as follows: we train the models using the data from September and then test the models on the data from October. For the online models ON-PROB, ON-SSVSB and ON-SSVSBMIL, we create batches with various sizes (e.g. one day, 1/2 day, etc.) starting from the beginning of September to the end of October, update the models for every batch, and test the models on the next batch. To fairly compare them with the offline models, only the batches in October are used for evaluation.

4.2 Evaluation Metric

In this paper we adopt an evaluation metric introduced in [38] that directly reflects how many frauds a model can catch: the *rate of missed complaints*, which is the portion of customer complaints that the model cannot capture as fraud. Note that in our application, the labeled data was not created through random sampling, but via a pre-screening moderation system using the expert-tuned coefficients (the data were created when only the expert model was deployed). This in fact introduces biases in the evaluation for the metrics which only use the labeled observations but ignore the unlabeled ones. This *rate of missed complaints* metric however covers both labeled and unlabeled data since customers do not know which cases are labeled, hence it is unbiased for evaluating the model performance.

Recall that our data were generated as follows: For each case the 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 determined as clean and not reviewed by anyone. Although for those cases that are not labeled we do not immediately know from the system whether they are fraud or not, the real fraud cases would still show up from the complaints filed by victims of the frauds. Therefore, if we want to prove that one machine-learned model is better than another, we have to make sure

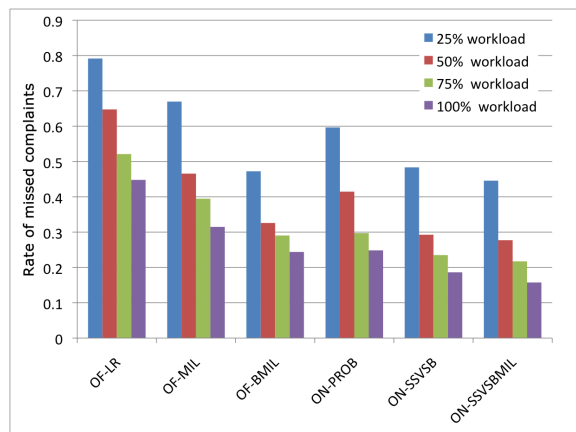


Figure 3: The rates of missed customer complaints for workload rates equal to 25%, 50%, 75% and 100% for all the offline models and online models with daily batches.

that with the same or even less expert labeling workload, the former model is able to catch more frauds (i.e. generate less customer complaints) than the latter one.

For any test batch, we regard the number of labeled cases as the expected 100% workload N , and for any model we could re-rank all the cases (labeled and unlabeled) in the batch and select the first M cases with the highest scores. We call M/N the “workload rate” in the following text. For a specific workload rate such as 100%, we could count the number of reported fraud complaints C_m in the M cases. Denote the total number of reported fraud complaints in the test batch as C , we define the *rate of missed complaints* as $1 - C_m/C$ given the workload rate M/N . Note that since in model evaluation we re-rank all the cases including both labeled and unlabeled data, different models with the same workload rate (even 100%) usually have different rates of missed customer complaints. We argue model A is better than model B if given the same workload rate, the rate of missed customer complaints for A is lower than B .

4.3 Model Performance

We ran all of the offline and online models on our real auction fraud detection data and show the rates of missed customer complaints given 100% workload rate for Oct 2010 in Table 1. Note that for online models we tried δ (one key parameter to control how dynamically the model evolves) for different values (0.6, 0.7, 0.75, 0.8, 0.9, 0.95 and 0.99) and report the best in the table. For ω we also did similar tuning and found that $\omega = 0.9$ seems to be a good value for all models. From the table it is very obvious that the online models are generally better than the corresponding offline models (e.g. ON-PROB versus OF-LR, ON-SSVSBMIL versus OF-BMIL), because online models not only learn from the September training period but also update for every batch during the October test period. Comparing the online models described in this paper, ON-SSVSB is significantly better than ON-PROB since it considers online feature selection and also bounds coefficients as domain knowledge. ON-SSVSBMIL further improves slightly over ON-SSVSB because it considers the “bagged” behavior of the expert labeling process using multiple instance learning.

δ	0.75	0.8	0.9	0.95	0.99
Rate of Missed Complaints	0.1463	0.1330	0.1641	0.1729	0.1973

Table 2: The rates of missed customer complaints for ON-SSVSBMIL (100% workload rate, batch size equal to 1/2 day and $w = 0.9$), with different values of δ .

Finally, almost all the offline and online models, except LR, are better than the Expert model. This is quite expected since machine-learned models given sufficient data usually can beat human-tuned models. In Figure 2 (the left plot) we show the boxplots of the rates of missed customer complaints for 100% workload on a daily basis for all the offline and online models (daily batch). In Figure 3 we plot the rates of missed customer complaints versus different workload rates for all models with daily batches. From both figures we can obtain very similar conclusions as those drawn in Table 1.

Impact of different batch sizes. For our best model ON-SSVSBMIL we tried different batch sizes, i.e. 1 day, 1/2 day, 1/4 day and 1/8 day, and tuned δ for each batch size. The overall model performance is shown in Table 1, and Figure 2 (the right plot) shows the boxplots of the model performance for different batch sizes on a daily basis. It is interesting to observe that batch size equal to 1/2 day gives the best performance. In fact, although using small batch sizes allows the online models to update more frequently to respond to the fast-changing pattern of the fraudulent sellers, large batch sizes often provide better model fitting than small batch sizes in online learning. This brings a trade-off in performance between the adaptivity and stability of the model. From Table 1 and Figure 2 we can clearly see this trade-off and it turns out that 1/2 day becomes the optimal batch size for our application. From the table we also observe that as the batch size becomes smaller, the best δ becomes larger, which is quite expected and reasonable.

Tuning δ . In Table 2, we show the impact of choosing different values of δ for ON-SSVSBMIL with 100% workload rate, batch size equal to 1/2 day and $w = 0.9$. Intuitively small δ implies that the model is more dynamic and puts more weight on the most recent data, while large δ means the model is more stable. When $\delta = 0.99$, it means that the model treats all of the historical observations almost equally. From the table it is obvious to see that δ has a significant impact on the model performance, and the optimal value $\delta = 0.8$ implies that the fraudulent sellers do have a dynamic pattern of generating frauds.

Changing patterns of feature values and importance. Embedding SSVS into the online modeling not only helps the fraud detection performance, but also provides a lot of insights of the feature importance. In Figure 4 for ON-SSVSBMIL with daily batches, $\delta = 0.7$ and $\omega = 0.9$ we selected a set of features to show how their posterior probabilities of being 0 (i.e. p_{jt}) evolve over time. From the figure we observe four types of features: The “always important” features are the ones that have p_{jt} close to 0 consistently. The “always non-useful” features are the ones that have p_{jt} always close to 1. There are also several features with p_{jt} close to the prior probability 0.5, which implies that we do not have much data to determine whether they are useful

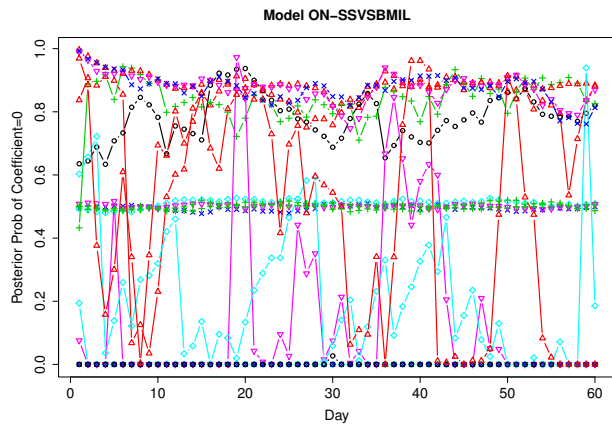


Figure 4: For ON-SSVSBMIL with daily batches, $\delta = 0.7$ and $\omega = 0.9$, the posterior probability of $\beta_{jt} = 0$ (j is the feature index) over time for a selected set of features.

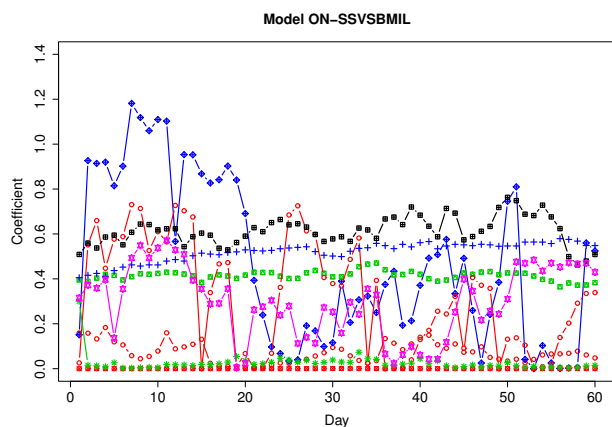


Figure 5: For ON-SSVSBMIL with daily batches, $\delta = 0.7$ and $\omega = 0.9$, the posterior mean of β_{jt} (j is the feature index) over time for a selected set of features.

or not (i.e. the appearance rates of these features are quite low in the data). Finally, the most interesting set of features are the ones that have a large variation of p_{jt} day over day. One important reason to use online feature selection in our application is to capture the dynamics of those unstable features. In Figure 5 we show the posterior mean of a randomly selected set of features. It is obvious that while some feature coefficients are always close to 0 (unimportant features), there are also many features with large variation of the coefficient values.

5. CONCLUSION AND FUTURE WORK

In this paper we build online models for the auction fraud moderation and detection system designed for a major Asian online auction website. By empirical experiments on a real-world online auction fraud detection data, we show that our proposed online probit model framework, which combines online feature selection, bounding coefficients from expert knowledge and multiple instance learning, can significantly

improve over baselines and the human-tuned model. Note that this online modeling framework can be easily extended to many other applications, such as web spam detection, content optimization and so forth.

Regarding to future work, one direction is to include the adjustment of the selection bias in the online model training process. It has been proven to be very effective for offline models in [38]. The main idea there is to assume all the unlabeled samples have response equal to 0 with a very small weight. Since the unlabeled samples are obtained from an effective moderation system, it is reasonable to assume that with high probabilities they are non-fraud. Another future work is to deploy the online models described in this paper to the real production system, and also other applications.

6. REFERENCES

- [1] D. Agarwal, B. Chen, and P. Elango. Spatio-temporal models for estimating click-through rate. In *Proceedings of the 18th international conference on World wide web*, pages 21–30. ACM, 2009.
- [2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. *Advances in neural information processing systems*, pages 577–584, 2003.
- [3] C. Bliss. The calculation of the dosage-mortality curve. *Annals of Applied Biology*, 22(1):134–167, 1935.
- [4] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*, volume 53. Cambridge University Press New York, 1998.
- [5] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [6] R. Brent. *Algorithms for minimization without derivatives*. Dover Pubns, 2002.
- [7] D. Chau and C. Faloutsos. Fraud detection in electronic auction. In *European Web Mining Forum (EWMF 2005)*, page 87.
- [8] H. Chipman, E. George, and R. McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.
- [9] W. Chu, M. Zinkevich, L. Li, A. Thomas, and B. Tseng. Unbiased online active learning in data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 195–203. ACM, 2011.
- [10] C. Chua and J. Wareham. Fighting internet auction fraud: An assessment and proposal. *Computer*, 37(10):31–37, 2004.
- [11] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1631–1643, 2005.
- [12] N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge university press, 2006.
- [13] 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.
- [14] Federal Trade Commission. Internet auctions: A guide for buyers and sellers. <http://www.ftc.gov/bcp/online/pubs/online/auctions.htm>, 2004.