Web Model of Practical Control System for Auction Threat Detection

Author Details:

Bonthu Subrahmanyam

MCA,M .Tech(cse)

Asst.Professor,

Department of MCA,

BVC Institute of Technology and Science, Batlapalem, Amalapuram, Andhrapradesh.

(Affiliated to JNTU, Kakinada)

ABSTRACT:

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

Categories and Subject Descriptors

[Artiftcial Intelligence]: Learning—*Parameter Learning* ; [Database Management]: Database Applica- tions—*Data Mining*

General Terms

Algorithms, Experimentation, Performance

Keywords

Web Auction, Scam Detection, Web Modeling, Web 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 web, which benefits everyone in terms of con- venience and profitability. The traditional web 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. Web auction however is a dif- ferent 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,

he winner gets the item with their highest winning bid.

Similar to any platform supporting financial transactions, web auction attracts criminals to commit scam. The vary- ing types of auction scam 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 pay- ments to be wired directly to them via bank-to-bank wire transfer. Furthermore, some criminals apply phishing tech- niques to steal high-rated seller's accounts so that potential buyers can be easily deceived due to their good rating. Vic- tims of scam transactions usually lose their money and in most cases are not recoverable. As a result, the reputation of the web auction services is hurt significantly due to scam crimes.

To provide some assurance against scam, E-commerce sites often provide insurance to scam victims to cover their loss up to a certain amount. To reduce the amount of such compensations and improve their web reputation, e- commerce providers often adopt the following approaches to control and prevent scam. The identifies of registered users are validated through email, SMS, or phone verifica- tions. A rating system where buyers provide feedbacks is commonly used in e-commerce sites so that scamulent sellers can be caught immediately after the first wave of buyer com- plaints. 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 scams with a moderation system, there are still many outstanding and challenging cases. Criminals and scamulent sellers frequently change their accounts and IP addresses to avoid being caught. Also, it is usually infea- sible for human experts to investigate every buyer and seller to determine if they are committing scam, especially when the e-commerce site attracts a lot of traffic. The patterns of scamulent sellers often change constantly to take advantage of temporal trends. For instance, scamulent sellers tend to sell the "hottest" products at the time to attract more po- tential victims. Also, whenever they find a loophole in the scam detection system, they will immediately leverage the weakness.

In this paper, we consider the application of a moderation system for scam detection in a major Asian on- line auction site, where hundreds of thousands of new auc- tion cases are

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 *scam 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 se- lecting 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 scam detection significantly over the human- tuned weights [38]. In [38] the authors considered the sce- nario of building offline models by using the previous 30 days data to serve the next day. Since the response is bi- nary (scam or non-scam) 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 scams and reducing customer complaints given the same workload from human experts. However, of- fline models often meet the following challenges: (a) Since the auction scam 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 dif- ficult 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 web models. (b) Since the scamulent 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 scamulent, all the cases from this seller will be suspended immediately. Therefore smart scamulent 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 label- ing, the high cost makes it almost impossible to obtain a very large sample. Therefore for such systems (i.e. rela- tively small sample size with many features with temporal pattern), web feature selection is often required to pro- vide good performance. Human experts are also willing to see the results of web feature selection to monitor the effectiveness of the current set of features, so that they can understand the pattern of scams and further add or removesomefeatures.

[1] Due to the company security policy, we can not reveal any details of those features.

Our contribution. In this paper we study the problem of building web models for the auction scam detection mod- eration system, which essentially evolves dynamically over time. We propose a Bayesian probit web model frame- work 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 web modeling. Similar to [38], we consider the expert knowledge to bound the rule-based coefficients to be posi- tive. Finally, we consider to combine this web model with multiple instance learning [30] that gives even better empiri- cal performance. We report the performance of all the above models through extensive experiments using scam detection datasets from a major web 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 web 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 web auction scams for a major Asian site where hundreds of thousands of new auc- tion cases are posted every day. Every new case is sent to the proactive anti-scam moderation system for pre-screening to assess the risk of being scam. The current system is featured by:

- Rule-based features: Human experts with years of experience created many rules to detect whether a user is scam or not. An example of such rules is "blacklist", i.e. whether the user has been detected or complained as scam before. Each rule can be regarded as a binary feature that indicates the scam likeliness.
- Linear scoring function: The existing system only supports linear models. Given a set of coefficients (weights) on features, the scam score is computed as the weighted sum of the feature values.
- Selective labeling: If the scam score is above a cer- tain 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. scam or clean. Cases with higher scores have higher priorities in the queue to be reviewed. The cases whose scam score are below the threshold are determined as clean by the system without any human judgment.
- Scam churn: Once one case is labeled as scam by human experts, it is very likely that the seller is not trustable and may be also selling other scams; hence all the items submitted by the same seller are labeled as scam too. The scamulent 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 scamulent sellers.

Motivated by these specific attributes in the moderation system for scam detection, in this section we describe our Bayesian web modeling framework with details of model fitting via Gibbs sampling. We start from introducing the web 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 web probit regression framework so that the feature importance can dy- namically 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.

Web Probit Regression

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

$$P[y_{it} = 1|x_{it}, \beta_t] = \Phi(x_{it}' \beta_t), \quad (1)$$

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 ex- pressed in a hierarchical form as follows: For each observa- tion 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 $z_{it} <= 0$, then $y_{it} = 0$. z_{it} can then be modeled by a linear regression

$$z_{it} \sim N(x_{it}'\beta_t, 1). \quad (2)$$

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

$$\beta_t \sim N(\mu_t, \Sigma_t), \quad (3)$$

where μ_t and Σ_t are prior mean and prior covariance matrix respectively.

Model fttting. Since the posterior $\pi(\beta_t | y_t, x_t, \mu_t, \Sigma_t)$ does not have a closed form, this model is fitted by using the latent vector z_t through Gibbs sampling. For each iteration we first sample $(z_t | y_t, x_t, \beta_t)$ and then sample $(\beta_t | z_t, y_t, \mu_t, \Sigma_t)$. Specifically, for each observation i at time t, sample

$$\pi(z_{it}|y_{it} = 1, x_{it}, \beta_t) \sim N(x_{it}' \beta_t, 1), \quad (4)$$

truncated by 0 as lower bound. And

$$\pi(z_{it}|y_{it} = 0, x_{it}, \beta_t) \sim N(x_{it}' \beta_t, 1), \quad (5)$$

truncated by 0 as upper bound. Then sample

$$\pi(\beta_t|z_t, y_t, x_t) = (\beta_t|z_t, x_t) \sim N(\hat{m}_t, \hat{V}_t), \quad (6)$$

where

$$\hat{V}_t = (\Sigma^{-1} + x_t' x_t)^{-1}, \hat{m}_t = \hat{V}_t(\Sigma^{-1}\mu_t + x_t' z_t). \quad (7)$$

By iterative sampling the conditional posterior of 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 $\hat{\mu}_t$ and sample covariance $\hat{\Sigma}_t$, to serve as the posterior mean and covariance for β_t respectively.

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

$$\mu_{t+1} = \hat{\mu}_t, \Sigma_{t+1} = \hat{\Sigma}_t/\delta, \quad (8)$$

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 "forget- ting"). 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 web modeling technique has been commonly used in literature (see [1] and [37] for ex- ample). In practice, for simplicity we let $\Sigma_0 = \sigma^2 I$ and $\hat{0}$

$\Sigma_{t+1} = \mathrm{diag}(\Sigma_t)/\delta$ to ignore the covariance among the co-efficients of β_t.

Besides the probit link function used in this paper, an- other common link function for the binary response is lo- gistic [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 augmenta- tion, 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.

Web Feature Selection through SSVS

For regression problems with many features, proper shrink- age 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 scam- ulent sellers change their behavioral pattern quickly: Some rule-based feature that does not help today might helps a lot tomorrow. Therefore it is necessary to build an web feature selection framework that evolves dynamically to pro- vide both optimal performance and intuition. In this paper we embed the stochastic search variable selection (SSVS) [16] into the web 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

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

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^2_{jt}. Such prior is called the "spike and slab" prior in the literature [19] but how to embed it to web 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 $\pi(z_t|y_t, x_t, \beta_t)$

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

$$\pi(z_{it}|y_{it} = 1, x_{it}, \beta_t) \sim N(x_{it}' \beta_t, 1), \quad (10)$$

truncated by 0 as lower bound. And

$$\pi(z_{it}|y_{it} = 0, x_{it}, \beta_t) \sim N(x_{it}' \beta_t, 1), \quad (11)$$

truncated by 0 as upper bound. Denote

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

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

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

To sample $\pi(\beta_{jt}|\beta_{-j,t}, z_t, y_t, p_{0jt}, \mu_{jt}, \sigma_{jt})$,

$$\pi(\beta_{jt}|\beta_{-j,t}, z_t, y_t, p_{0jt}, \mu_t, \Sigma_t) \quad (14)$$

$$= \pi(\beta_{jt}|\beta_{-j,t}, z_t, p_{0jt}, \mu_t, \Sigma_t)$$

$$\propto \left[\prod_{i=1}^{n_t} \exp\left(-\frac{(\tilde{z}_{it} - x_{ijt}\beta_{jt})^2}{2}\right)\right] \left[p_{0jt}1(\beta_{jt} = 0) + \frac{1-p_{0jt}}{\sqrt{2\pi\sigma_{jt}^2}} \exp\left(-\frac{(\beta_{jt} - \mu_{jt})^2}{2\sigma_{jt}^2}\right)\right]$$

$$\propto \hat{\gamma}_{jt}1(\beta_{jt} = 0) + (1 - \hat{\gamma}_{jt})N(\hat{m}_{jt}, \hat{V}_{jt}),$$

where

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

$$\hat{m}_{jt} = \hat{V}_{jt}\left(x_{jt}' \tilde{z}_t + \frac{\mu_{jt}}{\sigma_{jt}^2}\right), \quad (16)$$

p_{0jt}

Web 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

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

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

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.

Coefficient Bounds

Incorporating expert domain knowledge into the model is often important and has been proved to boost the model per- formance (see [38] for instance). In our moderation system, the feature set x is proposed by experts with years of experi- ence in detecting auction scams. Most of these features are

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

Since the conditional posterior $\pi(\beta_{jt}|\beta_{-j,t}, z_t, y_t, p_{0jt}, \mu_{jt}, \sigma_{jt}) \sim \hat{\gamma}_{jt}1(\beta_{jt} = 0) + (1 - \hat{\gamma}_{jt})N(\hat{m}_{jt}, V_{jt})$, it implies that to sam- ple 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 $\beta^{(k)}_{jt}$, $k = 1, \ldots, N$. We estimate the posterior distribution of $\pi(\beta_{jt}|y_t, p_{0jt}, \mu_{jt}, \sigma_{jt})$ by

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

where

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

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

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

in fact "rules", i.e., any violation of one rule should ideally increase the probability of the seller being

extent. A simple example of such rules is the "blacklist", i.e. whether the seller has ever been detected or complained as scam before. However, for some of such rules simply applying probit regression as described in Section 2.1 or lo- gistic 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 fea- tures. 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

$$\beta_{jt} \sim p_{0jt} 1(\beta_{jt} = 0) + (1 - p_{0jt})N(\mu_{jt}, \sigma^2_{jt})1(\beta_{jt} > 0), \quad (24)$$

where $N(\mu_{jt}, \sigma^2_{jt})1(\beta_{jt} > 0)$ means β_{jt} is sampled from $N(\mu_{jt}, \sigma^2_{jt})$, 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

$$\pi(\beta_{jt}|\beta_{-j,t}, z_t, y_t, p_{0jt}, \mu_{jt}, \sigma_{jt}),$$

$$\pi(\beta_{jt}|\beta_{-j,t}, z_t, y_t, p_{0jt}, \mu_t, \Sigma_t) \quad (25)$$

$$\propto \prod_{i=1}^{n_t} [\exp(-\frac{(\tilde{z}_{it} - x_{ijt}\beta_{jt})^2}{2})]$$

$$[p_{0jt}1(\beta_{jt} = 0) + \frac{1 - p_{0jt}}{\sqrt{2\pi\sigma_{jt}^2}} \exp(-\frac{(\beta_{jt} - \mu_{jt})^2}{2\sigma_{jt}^2})1(\beta_{jt} > 0)]$$

$$\propto \hat{\gamma}_{jt}1(\beta_{jt} = 0) + (1 - \hat{\gamma}_{jt})N(\hat{m}_{jt}, \hat{V}_{jt})1(\beta_{jt} > 0),$$

where

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

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

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

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

$$\pi(\beta_{jt}|y_t, p_{0jt}, \mu_{jt}, \sigma_{jt}) \quad (29)$$

$$\sim \hat{p}_{jt}1(\beta_{jt} = 0) + (1 - \hat{p}_{jt})N(\hat{\mu}_{jt}, \hat{\sigma}_{jt}^2)1(\beta_{jt} > 0),$$

where

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

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 distribution and non-symmetric, the mean of the non-zero posterior samples tends to be higher than the real value of $\hat{\mu}_{jt}$. Let $q_{jt} = \sum_{k=1}^{N} 1(\beta_{jt}^{(k)} > 0)$, we find $\hat{\mu}_{jt}$ and $\hat{\sigma}_{jt}^2$ via maximizing the density function

$$\frac{q_{jt}}{\ldots} \sum^{N} (\beta_{jt}^{(k)} - \hat{\mu}_{jt})^2 1(\beta_{jt}^{(k)} > 0)$$

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

$$y_{it} = 0 \text{ iff } z_{ilt} < 0, \ \forall l = 1, \cdots, K_{it}; \quad (32)$$

otherwise $y_{it} = 1$, and

$$z_{ilt} \sim N(x_{ilt}' \beta_t, 1), \quad (33)$$

where β_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 fttting. 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 β_t remains the same, the process of sampling $\pi(z_t | y_t, x_t, \beta_t)$ is different. For seller i at time t,

$$\pi(z_{ilt}|y_{it} = 0, x_{ilt}, \beta_t) \sim N(x_{ilt}' \beta_t, 1), \quad (34)$$

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

$$\pi(\tilde{y}_{i1t}, \cdots, \tilde{y}_{iK_{it}t}|y_{it} = 1, x_{it}, \beta_t) \quad (35)$$

$$= \frac{\prod_{l=1}^{K_{it}} (\Phi(x_{ilt}'\beta_t))^{\tilde{y}_{ilt}}(1 - \Phi(x_{ilt}'\beta_t))^{1-\tilde{y}_{ilt}}}{\sum_{\sum_{l=1}^{K_{it}} \tilde{y}_{ilt}>0} \prod_{l=1}^{K_{it}} (\Phi(x_{ilt}'\beta_t))^{\tilde{y}_{ilt}}(1 - \Phi(x_{ilt}'\beta_t))^{1-\tilde{y}_{ilt}}}.$$

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

$$\pi(z_{ilt}|y_{it} = 1, \tilde{y}_{ilt} = 1, x_{ilt}, \beta_t) \sim N(x_{ilt}' \beta_t, 1), \quad (36)$$

truncated by 0 as lower bound. And

$$\pi(z_{ilt}|y_{it} = 1, \tilde{y}_{ilt} = 0, x_{ilt}, \beta_t) \sim N(x_{ilt}' \beta_t, 1), \quad (37)$$

truncated by 0 as upper bound.

The estimation of the posterior of β_t and the web mod- eling component are the same as those in the previous sec- tions.

$$\prod_{k=1} = (2\pi\hat{\sigma}_{jt}^{2}\Phi^{2}(\hat{\mu}_{jt}/\hat{\sigma}_{jt}))^{-\frac{1}{2}} \exp\left(-\frac{}{2\hat{\sigma}_{jt}^{2}}\right). \quad (31)$$

We find the optimal solution to equation (31) by alterna- tively fitting ($\hat{\mu}_{jt}$ $\hat{\sigma}_{jt}^2$) and ($\hat{\sigma}_{jt}^2$ $\hat{\mu}_{jt}$) to maximize the func- tion using [6].

The web 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 scam, then all of the cases from this seller are labeled as scam. 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

3. RELATED WORK

Web auction scam is always recognized as an important issue. There are articles on websites to teach people how to avoid web auction scam (e.g. [35, 14]). [10] categorizes auction scam into several types and proposes strategies to fight them. Reputation systems are used extensively by web- sites to detect auction scams, 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 on- line auction scam 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 scam. [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 scam detec- tion moderation system which incorporates domain knowl- edge such as coefficient bounds and multiple instance learn- ing.

In this paper we treat the scam detection problem as a bi- nary classification problem. The most frequently used mod- els 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 perfor- mance. Stochastic search variable selection (SSVS) [16] uses "spike and slab" prior [19] so that the posterior of the co- efficients have some probability being 0. Another approach is to consider the variable selection problem as model selec- tion, i.e. put priors on models (e.g. a Bernoulli prior on each coefficient being 0) and compute the marginal poste- rior probably 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 pos- terior mode [18]. Among non-linear models, tree models usually handles the non-linearity and variable selection si- multaneously. Representative work includes decision trees [29], random forests [5], gradient boosting [15] and Bayesian additive regression trees (BART) [8].

Web 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 web 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, web learning usually requires much lighter computation and memory load; hence it can be widely used in real-time systems with continuous sup- port of inputs. For web feature selection, representative applied work include [11] for the problem of object track- ing in computer vision research, and [21] for content-based image retrieval. Both approaches are simple while in this pa- per the embedding of SSVS to the web 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 pub- lished in the application area of image classification such as [25, 24]. The logistic regression framework of multiple in- stance learning is presented in [30], and the SVM framework is presented in [2].

4. EXPERIMENTS

We conduct our experiments on a real web auction scam

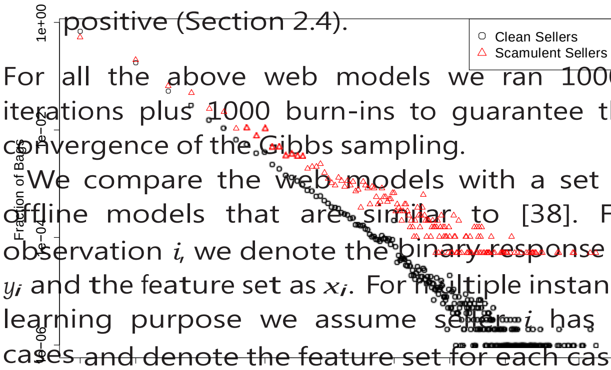


Figure 1: Fraction of bags versus the number of cases per bag ("bag size") submitted by scamulent and clean sellers respectively. A bag contains all the cases submitted by a seller in the same day.

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

ON-SSVSBMIL is the web probit

detection data set collected from a major Asian website. We consider the following web models:

- ON-PROB is the web probit regression model de- scribed in Section 2.1.
- ON-SSVSB is the web probit regression model with

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 web models we ran 10000 iterations plus 1000 burn-ins to guarantee the convergence of the Gibbs sampling.

We compare the web 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 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 x_{il}. The offline models are

- Expert has the human-tuned coefficients set by do- main experts based on their knowledge and recent scam- fighting experience.
- OF-LR is the offline logistic regression model that minimizes the loss function

$$L = \sum_i y_i \log(1 + \exp(-x_i'\beta)) + (1 - y_i)\log(1 + \exp(x_i'\beta)) + \rho\|\beta\|_2, \quad (38)$$

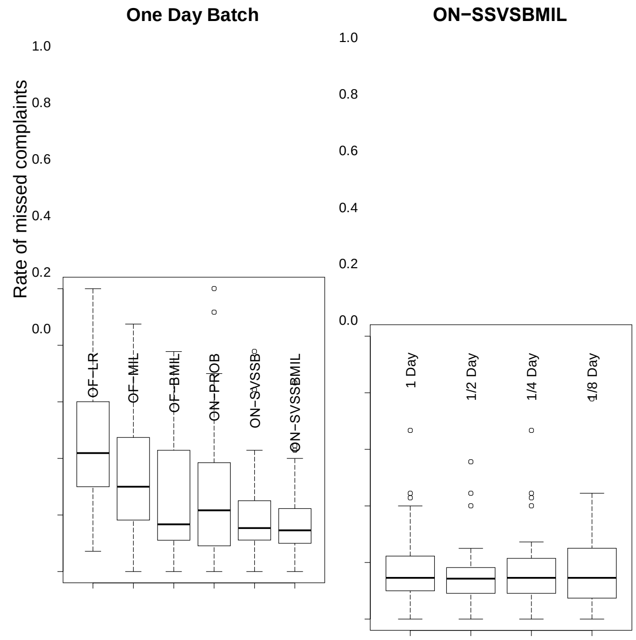
where ρ is the tuning L2 penalty parameter that can be learned by cross-validation.

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

$$L = \sum_i -y_i \log\Big(1 - \prod_{l=1}^{K_i} \frac{1}{1 + \exp(x_{il}'\beta)}\Big) + (1 - y_i)\sum_{l=1}^{K_i} \log(1 + \exp(x_{il}'\beta)) + \rho\|\beta\|_2 . \quad (39)$$

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.

The Data and Model Setting

Our application is a real scam moderation and detection system designed for a major Asian web auction website that attracts hundreds of thousands of new auction postings every day. The data consist of around 2M expert labeled auction cases with 20K of them labeled as scam 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 num- ber of unlabeled cases in the data is about 6M-10M. For each observation there is a set of features indicating how "suspi- cious" it is. To avoid future scamulent 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 scam. 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 scam, all of this seller's cases will be labeled as scam. 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 scamulent and clean sellers re- spectively. 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 cus- tomer complaints on a daily basis for all the offline and web models. It is obtained given 100% work- load rate.

instance learning can be useful for this data. It is also inter- esting to see that the scamulent 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 us- ing the data from September and then test the models on the data from October. For the web 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 begin- ning 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.

Evaluation Metric

In this paper we adopt an evaluation metric introduced in [38] that directly reflects how many scams a model can catch: the *rate of missed complaints*, which is the portion of customer complaints that the model cannot capture as scam. 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 met- rics which only use the labeled observations but ignore the unlabeled ones. This *rate of missed complaints* metric how- ever 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 scor- ing function to determine whether to send it for expert label- ing. If so, experts review it and make a scam or non-scam 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 scam or not, the real scam cases would still show up from the complaints filed by victims of the scams. Therefore, if we want to prove that one machine- learned model is better than another, we have to make sure

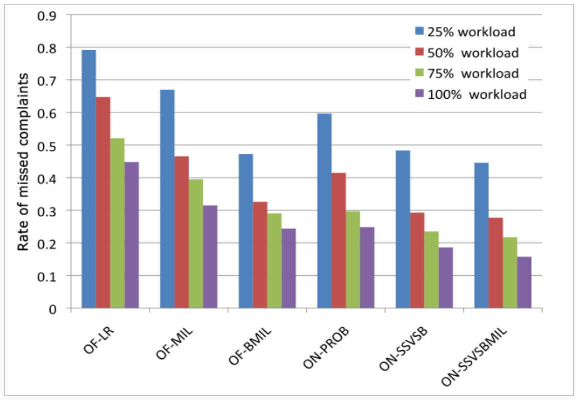


Figure 3: The rates of missed customer complaints for workload rates equal to 25%, 50%, 75% and 100% for all the offline models and web models with daily batches.

that with the same or even less expert labeling workload, the former model is able to catch more scams (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 scam complaints C_m in the M cases. Denote the total number of reported scam 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.

Model Performance

We ran all of the offline and web models on our real auction scam detection data and show the rates of missed customer complaints given 100% workload rate for Oct 2010 in Table 1. Note that for web 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 sim- ilar tuning and found that ω = 0.9 seems to be a good value for all models. From the table it is very obvious that the web models are generally better than the cor- responding offline models (e.g. ON-PROB versus OF-LR, ON-SSVSBMIL versus OF-BMIL), because web models not only learn from the September training period but also update for every batch during the October test period. Com- paring the web models described in this paper, ON-SSVSB is significantly better than ON-PROB since it considers on- line 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	0.74	0.8	0.9	0.95	
Rate of Missed	0.1459 0.1978	0.1335	0.1639	0.1732	

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 web 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 web 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 fig- ures 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 web models to update more frequently to respond to the fast-changing pattern of the scamulent sell- ers, large batch sizes often provide better model fitting than small batch sizes in web 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 δ = 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 scamulent sellers do have a dynamic pattern of generating scams.

Changing patterns of feature values and impor- tance. Embedding SSVS into the web modeling not only helps the scam 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 proba- bilities of being 0 (i.e. $\hat{p_{jt}}$) evolve over time. From the figure we observe four types of features: The "always important" features are the ones that have $\hat{p_{jt}}$ close to 0 consistently. The "always non-useful" features are the ones that have $\hat{p_{jt}}$ always close to 1. There are also several features with $\hat{p_{jt}}$ close to the prior probability 0.5, which implies that we do not have much data to determine whether they are useful

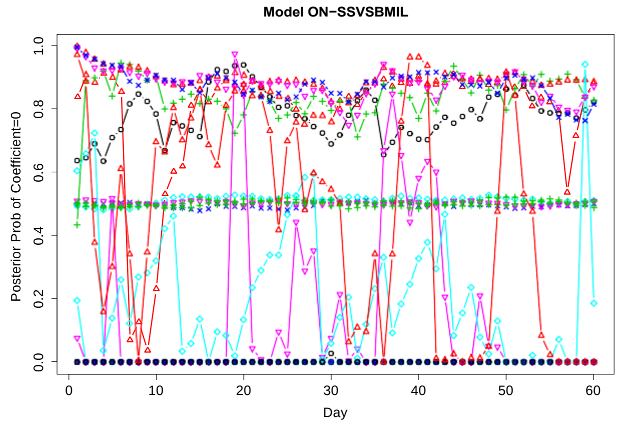


Figure 4: For ON-SSVSBMIL with daily batches, $\delta = 0.7$ and $\omega = 0.9$, the posterior probability of $\beta_{jt} = 0$ (j is the feature index) over time for a selected set of features.

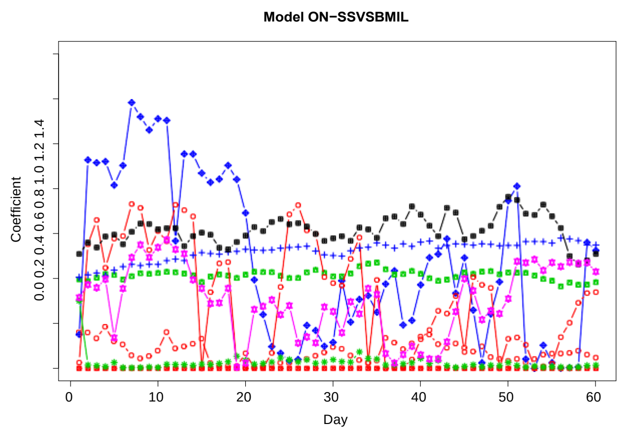


Figure 5: For ON-SSVSBMIL with daily batches, $\delta = 0.7$ and $\omega = 0.9$, the posterior mean of β_{jt} (j is the feature index) over time for a selected set of features.

or not (i.e. the appearance rates of these features are quite low in the data). Finally, the most interesting set of fea- tures are the ones that have a large variation of $\hat{p_{jt}}$ day over day. One important reason to use web feature selection in our application is to capture the dynamics of those unsta- ble 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 web models for the auction scam moderation and detection system designed for a major Asian web auction website. By empirical experiments on a real- word web auction scam detection data, we show that our proposed web probit model framework, which combines web feature selection, bounding coefficients from expert knowledge and multiple instance learning, can significantly improve over baselines and the human-tuned model. Note that this web 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 web 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-scam. Another future work is to deploy the web 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. *Web 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. Scam 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 web 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 scam: An assessment and proposal. *Computer*, 37(10):31-37, 2004.

[11] R. Collins, Y. Liu, and M. Leordeanu. Web 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/cweb/pubs/web/auctions.htm, 2004.

[15] J. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367-378, 2002.

[16] E. George and R. McCulloch. Stochastic search variable selection. *Markov chain Monte Carlo in practice*, 68:203-214, 1995.

[17] D. Gregg and J. Scott. The role of reputation systems in reducing on-line auction scam. *International Journal of Electronic Commerce*, 10(3):95-120, 2006.

[18] C. Hans, A. Dobra, and M. West. Shotgun stochastic search for Şlarge pŤ regression. *Journal of the American Statistical Association*, 102(478):507-516, 2007.

[19] H. Ishwaran and J. Rao. Spike and slab variable selection: frequentist and bayesian strategies. *The Annals of Statistics*, 33(2):730-773, 2005.

[20] T. Jaakkola and M. Jordan. A variational approach to bayesian logistic regression models and their extensions. In *Proceedings of the sixth international workshop on artificial intelligence and statistics*. Citeseer, 1997.

[21] W. Jiang, G. Er, Q. Dai, and J. Gu. Similarity-based web

feature selection in content-based image retrieval. *Image Processing, IEEE Transactions on*, 15(3):702-712, 2006.

[22] K. Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307-319, 2003.

[23] Y. Ku, Y. Chen, and C. Chiu. A proposed data mining approach for internet auction scam detection. *Intelligence and Security Informatics*, pages 238-243, 2007.

[24] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Advances in neural information processing systems*, pages 570-576, 1998.

[25] O. Maron and A. Ratan. Multiple-instance learning for natural scene classification. In *In The Fifteenth International Conference on Machine Learning*, 1998.

[26] P. McCullagh and J. Nelder. *Generalized linear models*. Chapman & Hall/CRC, 1989.

[27] A. B. Owen. Infinitely imbalanced logistic regression. *J. Mach. Learn. Res.*, 8:761-773, 2007.

[28] S. Pandit, D. Chau, S. Wang, and C. Faloutsos. Netprobe: a fast and scalable system for scam detection in web auction networks. In *Proceedings of the 16th international conference on World Wide Web*, pages 201-210. ACM, 2007.

[29] J. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81-106, 1986.

[30] 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.

Author Details:

Bonthu Subrahmanyam

MCA,M .Tech(cse)

Asst.Professor,

Department of MCA,

BVC Institute of Technology and Science,Batlapalem,Amalapuram

(Affiliated to JNTU,Kakinada)