Statistics and Its Interface Volume 9 (2016) 389–394

An EM algorithm for click fraud detection

Xuening Zhu*, Da Huang†, Rui Pan‡, and Hansheng Wang*

This paper is concerned with the problem of click fraud detection. We assume each visitor of a website carries a latent indicator, which labels him/her as a regular or malicious user. Information such as number of clicks, number of page views (PVs) and time difference between consecutive clicks are cooperated in our newly proposed statistical model. We allow those random variables to share the same distribution but with different parameters according to the visitor's type. An EM algorithm is then suggested to obtain the maximum likelihood estimator. As a result, click fraud detection can be implemented by estimating the posterior malicious probability of each visitor. Simulation studies are conducted to assess the finite sample performance. We also demonstrate the usefulness of the proposed method via an empirical analysis of a real life example on search engine marketing.

AMS 2000 subject classifications: Primary 62H30.
Keywords and phrases: Click fraud detection, EM algorithm, Maximum likelihood estimator, Search engine marketing.

1. INTRODUCTION

Online advertising is an effective way to promote products to customers. It is a large and rapidly growing business. By a recent report of Interactive Advertising Bureau (www.iab.net), for the first six months of 2014, revenues of internet advertising increased 15.1% over the same period of 2013. Advertiser, publisher and broker are the three main players in the online advertisement market. Advertisers are those who intend to attract customers and promote their products through advertisements. Publishers are providers of online platforms, such as websites, on which the advertisements may be displayed. A broker, defined by [6], is a searching service provider through which customers may find their interested advertisements. Typical brokers include Google, Yahoo, Baidu, and many others. In the online advertisement business, advertisers pay money to publishers for their display services. In turn, if a customer accesses advertisements through a broker's searching service, the publisher should share corresponding earnings with the broker.

Fraudulent click happens if a visitor with malicious intention deceptively clicks on an advertisement. The purpose is either to increase a publisher's revenue or cause financial loss to its competitors. As predicted by Solve Media (www.solvemedia.com), fraudulent clicks will cost online advertisers about 11.6 billion USD in 2014. This makes click fraud detection an important research topic. Past researches can be roughly divided into two categories. Methodologies involved in the first category rely on full traffic information, which refer to the click behaviors happened across different advertisers. See [7], [8], [5], [10] and [4] for more details. However, full traffic information is only available to brokers, these methods cannot be implemented directly by individual advertisers. To overcome the constraint of data accessibility, researchers developed various approaches applicable for individual advertisers. For instance, [11] proposed the method of duplicate detection to fight against click fraud. Later, [1] investigated the same problem by proposing a burst detection method.

All aforementioned methods focus on detecting clicks with abnormal high frequencies. Other information, such as time difference between consecutive clicks, number of pages viewed by a visitor, are not considered. In order to take advantage of various information resources, we propose a novel statistical model. Specifically, our work is motivated by the following empirical evidences. First, within a given time period (e.g., one hour), number of clicks generated by malicious visitors is unusually large. Second, visitors with smaller number of PVs are more likely to be deceptive. Lastly, the time difference between fraudulent clicks is usually smaller than normal.

In our model, we assume that each visitor (identified by IP address) carries a latent indicator, which labels him/her as a regular or malicious user. According to the visitor's type, we then assume that the number of clicks should follow a Poisson distribution with different parameters. Similar assumption is made for the number of PVs. Moreover, we employ exponential distribution to model time difference between consecutive clicks. Subsequently, an EM algorithm is developed to estimate the unknown parameters. The Bayesian formula can be used to evaluate each visitor's malicious probability. Extensive simulation studies are

*Supported in part by National Natural Science Foundation of China (NSFC, 11131002, 11271032, 71532001, 11525101), Center for Statistical Science at Peking University, and the Business Intelligence Research Center at Peking University.
†Corresponding author. Supported in part by the Youth Innovation Team Project for Arts and Social Science of Fudan University, China Statistical Research (CSR, 2015LY77) and National Natural Science Foundation of China (NSFC, 71531006, 11571080, 11571081, 91546104).
‡Supported by the Program for Innovation Research in Central University of Finance and Economics.

conducted in order to assess the method's finite sample performance. At last, we illustrate the proposed method by an empirical analysis.

The rest of the article is organized as follows. In Section 2, we describe the proposed model and its solution via an EM algorithm. Simulation studies and a real example are presented in Section 3. Finally, we conclude our findings and discuss possible future improvements in Section 4.

2. THE METHODOLOGY

2.1 Model and notation

Assume a total number of n independent visitors, indexed by $i = 1, \cdots, n$, browse a website during a specific time period (e.g., one hour). For the ith visitor, we assign a latent variable Z_i indicating his type. In particular, $Z_i = 0$ if the ith visitor is regular, otherwise we set $Z_i = 1$ to label him as a malicious one. Z_is are then assumed to be independently and identically distributed according to the prior probability $P(Z_i = k) = \pi_k$ for $k \in \{0, 1\}$, where $\pi_0 + \pi_1 = 1$. As a result, π_1 represents a visitor's malicious probability. We further define $Z_{ik} = I(Z_i = k)$, where $I(\cdot)$ is the indicator function. Next, we record his click number by C_i and denote the number of PVs generated by the jth click by P_{ij}, where $j = 1, \cdots, C_i$. By definition, we should require $P_{ij} \geq 1$. To measure the ith visitor's click frequency, we use Δ_{ij} to represent the time difference between the jth and $(j+1)$th clicks, where $1 \leq j \leq C_i - 1$. As a result, Δ_{ij} is computable only for the visitors with $C_i \geq 2$.

Given $Z_i = k$, we assume that $C_i - 2$ and $P_{ij} - 1$ are Poisson distributed with parameters λ_k and μ_k, respectively. Further, Δ_{ij} is assumed to follow an exponential distribution with parameter δ_k. For the ith visitor, define $P_i = \sum_{j=1}^{C_i} P_{ij}$ as the total number of PVs and $\Delta_i = \sum_{j=1}^{C_i - 1} \Delta_{ij}$ the accumulated time difference. Assume that all P_{ij}s and Δ_{ij}s are independent, it can be easily shown that $P_i - C_i$ follows a Poisson distribution with parameter $C_i\mu_k$, and Δ_i follows a Gamma distribution with shape parameter $C_i - 1$ and scale parameter δ_k. For convenience, we use $D_i = (C_i, P_i, \Delta_i)'$ to denote the information provided by the ith visitor. We further require P_i and Δ_i to be independently distributed conditional on C_i. Denote the conditional probability density function $f(\cdot|Z_i = k, \theta)$ as $f_k(\cdot|\theta)$, then the density function may be written as

$$
\begin{aligned}
f_k(D_i|\theta) &= f_k(C_i|\theta) f_k(P_i|C_i, \theta) f_k(\Delta_i|C_i, \theta) \\
&= \frac{\lambda_k^{C_i-2}}{(C_i-2)!} e^{-\lambda_k} \frac{(C_i\mu_k)^{P_i - C_i}}{(P_i - C_i)!} e^{-C_i\mu_k} \\
&\quad \frac{\Delta_i^{C_i-2}}{\delta_k^{C_i-1}(C_i-2)!} e^{-\Delta_i/\delta_k}. \qquad (1)
\end{aligned}
$$

where $\theta = (\pi_1, \lambda_0, \lambda_1, \mu_0, \mu_1, \delta_0, \delta_1)'$ is the collection of parameters. Furthermore, using $\mathbb{D} = \{D_i, i = 1, \cdots, n\}$ to denote all data we observed and $\mathbb{Z} = \{Z_i, i = 1, \cdots, n\}$ the collection of user's type. As a result, the log-likelihood function can be written as

$$
\begin{aligned}
\ell(\theta|\mathbb{D}, \mathbb{Z}) &= \log \prod_{i=1}^{n} \prod_{k=0}^{1} \left\{ \pi_k f_k(C_i, P_i, \Delta_i) \right\}^{Z_{ik}} \\
&= \sum_{i=1}^{n} \sum_{k=0}^{1} Z_{ik} \Bigg\{ A_i + \ln \pi_k + (C_i - 2) \ln \lambda_k - \lambda_k \\
&\quad + (P_i - C_i) \ln \mu_k - C_i\mu_k - \frac{\Delta_i}{\delta_k} - (C_i - 1) \ln \delta_k \Bigg\},
\end{aligned}
$$

where A_is are normalizing constants with respect to the ith visitor.

2.2 The EM algorithm

Since Z_i is latent, the corresponding indicator Z_{ik} is also unobservable. We then follow the idea of [3] and develop here an EM algorithm to estimate the parameters of interest. In particular, after setting a initial $\hat{\theta}^{(0)}$, we iterate the following steps to achieve the estimators. Specifically, in the mth iteration,

E-Step. Estimate Z_{ik} by its posterior mean $Z_{ik}^{(m)}$. Here,

$$
\begin{aligned}
Z_{ik}^{(m)} &= E(Z_{ik}|D_i, \hat{\theta}^{(m-1)}) \\
&= \frac{\hat{\pi}_k^{(m-1)} \hat{f}_k(D_i|\hat{\theta}^{(m-1)})}{\sum_{k=0}^{1} \hat{\pi}_k^{(m-1)} \hat{f}_k(D_i|\hat{\theta}^{(m-1)})}, \qquad (2)
\end{aligned}
$$

where $\hat{f}_k(D_i|\hat{\theta}^{(m-1)})$ can be estimated from equation (1) by substituting the parameters θ by the current estimators $\hat{\theta}^{(m-1)}$.

M-Step. Given $Z_{ik}^{(m)}$, find $\hat{\theta}^{(m)}$ by maximizing $\ell(\theta|D_i, Z_{ik}^{(m)})$ as shown in (2). Particularly, we have

$$
\begin{aligned}
\hat{\lambda}_k^{(m)} &= \frac{\sum_{i=1}^{n} (C_i - 2) Z_{ik}^{(m)}}{\sum_{i=1}^{n} Z_{ik}^{(m)}}, \\
\hat{\mu}_k^{(m)} &= \frac{\sum_{i=1}^{n} (P_i - C_i) Z_{ik}^{(m)}}{\sum_{i=1}^{n} C_i Z_{ik}^{(m)}}, \\
\hat{\delta}_k^{(m)} &= \frac{\sum_{i=1}^{n} \Delta_i Z_{ik}^{(m)}}{\sum_{i=1}^{n} (C_i - 1) Z_{ik}^{(m)}}, \\
\hat{\pi}_k^{(m)} &= \frac{\sum_{i=1}^{n} Z_{ik}^{(m)}}{n}.
\end{aligned}
$$

Repeat the above steps until the EM algorithm converges and the final results are the desired estimators.

In order to protect a website from online attack, one needs to evaluate the malicious likelihood for each visitor in time. Technically, this amounts to compute the posterior probability for $Z_i = 1$, which is given by

$$
\hat{p}(Z_i = 1) = \frac{\hat{\pi}_1 \hat{f}_1(D_i|\hat{\theta})}{\sum_{k=0}^{1} \hat{\pi}_k \hat{f}_k(D_i|\hat{\theta})}. \qquad (3)
$$

As a result, a visitor is more likely to be a malicious one if $\hat{p}(Z_i = 1)$ is greater than some pre-specified threshold.

3. NUMERICAL STUDIES

3.1 Parameter estimation

To demonstrate the finite sample performance of the proposed method, we present here some simulation examples. The sample sizes are fixed to be $n = 500, 1000, 2000$, and 5000. Furthermore, the fraudulent probability is set to be $\pi = 0.3$ through all experiments. Accordingly, Z_i is generated with $P(Z_i = 1) = \pi$ and $P(Z_i = 0) = 1 - \pi$. We next consider the following five examples here.

EXAMPLE 1. Recall that $(\lambda_0, \mu_0, \delta_0)$ and $(\lambda_1, \mu_1, \delta_1)$ are parameters related to regular and fraudulent visitors. In this example, we set them to be $(1, 2, 100)$ and $(5, 2, 100)$. Note that, for regular and fraudulent visitors, the only difference lies in the parameter for the number of clicks (i.e., λ_k). The visitor's behavior of page view and click frequency is set to be the same between two kinds of visitors. As a result, for a regular visitor, the average click number is much smaller than that of a fraudulent one.

EXAMPLE 2. In this example, two sets of parameters are fixed to be $(1, 2, 100)$ and $(3, 2, 30)$ respectively. The differences between two kinds of visitors lie in the click number as well as in the time difference. Since δ_k represents the average of time difference measured in seconds, it can be seen that the time difference between consecutive clicks of a regular visitor is about one minute and a half, while it is thirty seconds for a malicious visitor. These settings represent the basic assumption is that the fraudulent visitor not only has larger click number than the regular visitor, but also smaller time difference between two clicks.

EXAMPLE 3. In the third example, we assume that fraudulent and regular visitors can be distinguished in the sense that their click number, PVs and time difference all have different patterns. As a result, the corresponding parameters are set to be $(1, 2, 100)$ and $(3, 1, 30)$. Similar to Example 2, malicious visitors have larger number of clicks and smaller time difference. What's more, they also view fewer pages than the regular visitors.

EXAMPLE 4. The fourth setting is a challenging one. We assume two groups of visitors only have very small difference so that the parameters are set to be $(1, 2, 100)$ and $(1.5, 1.5, 90)$. We want to examine whether our method can distinguish two types of visitors when their activity patterns are similar.

EXAMPLE 5. We test the robustness of our proposed method in this example. More precisely, we investigate the performance of our method once the distributions of variables (i.e., number of clicks, number of PVs, and time difference) are different from underlying assumptions. To this end, the number of clicks is generated from a geometric distribution with parameters G(0.85) for regular visitors and G(0.4) for malicious ones. Furthermore, the number of PVs is also assumed to follow a geometric distribution with parameters G(0.5) and G(0.9) for regular and malicious visitors respectively. At last, time difference is generated from a log normal distribution with the same variance 1 but different means of 5 and 3.5 for two kinds of visitors.

3.2 Assessment criteria

In this section, we propose some measures to gauge the performance of our proposed method. For each simulation setup, the experiment is randomly replicated R times. For $r = 1, \cdots, R$, let $\hat{\theta}^{(r)} = (\hat{\pi}_1^{(r)}, \hat{\lambda}_0^{(r)}, \hat{\mu}_0^{(r)}, \hat{\delta}_0^{(r)}, \hat{\lambda}_1^{(r)}, \hat{\mu}_0^{(r)}, \hat{\delta}_0^{(r)})'$ be the parameter set estimated by the EM algorithm in the r-th simulation replication.

We firstly consider measures of estimation accuracy. In particular, we evaluate the mean absolute error for each parameter. Take λ_0 as an example, we are interested in the mean absolute error

$$\text{MAE}(\lambda_0) = \frac{1}{R}\sum_{r=1}^{R} |\hat{\lambda}_0^{(r)} - \lambda_0|,$$

where $\hat{\lambda}_0^{(r)}$ is the estimation of λ_0 in the rth simulation replication. We are also interested in the overall estimation performance. Correspondingly, we define a mean squared error as

$$\text{MSE} = \frac{1}{R}\sum_{r=1}^{R} \|\hat{\theta}^{(r)} - \theta\|.$$

In addition, we consider some ratio of identification correctness. Specifically, we assign each visitor a posterior probability according to (3) after obtaining the parameter estimation. Once the posterior probability exceeds a prespecified threshold, τ, the corresponding visitor is regarded as a fraudulent visitor. That is, $\hat{Z}_i = 1$ if $\hat{p}(Z_i = 1) > \tau$, otherwise $\hat{Z}_i$ is set to be 0. We then consider the following three measures, which are

$$\begin{aligned}
\text{Error}^{(r)} &= \frac{\sum_{i=1}^{n} I(Z_i \neq \hat{Z}_i)}{n} \times 100\%, \\
\text{TPR}^{(r)} &= \frac{\sum_{i=1}^{n} I(Z_i \times \hat{Z}_i = 1)}{\sum_{i=1}^{n} I(Z_i = 1)} \times 100\%, \\
\text{FPR}^{(r)} &= \frac{\sum_{i=1}^{n} I(Z_i = 0) \times I(\hat{Z}_i = 1)}{\sum_{i=1}^{n} I(Z_i = 0)} \times 100\%.
\end{aligned}$$

Note that $\text{Error}^{(r)}$ measures the overall prediction error of our method in the rth replication. $\text{TPR}^{(r)}$ stands for the ability of our method to correctly predict a fraudulent visitor. Finally, $\text{FPR}^{(r)}$ is the probability that a regular visitor is misclassified to be a malicious one. We set the threshold τ to be $\hat{\pi}^{(r)}$ in the rth replication and report the average Error, TPR and FPR over R replications as our final measures.

We compare the proposed strategy against two other existing methods. They are the burst detection (BD) method

Table 1. Simulation Results with 1,000 Replications. MAEs and MSE are calculated for the parameters of the EM algorithm. Misclassification error (Error), true positive rate (TPR) and false positive rate (FPR) are reported for the EM algorithm, the method of BD, and the DST method

n	MAE							MSE	EM(%)			BD(%)			DST(%)		
	λ_0	μ_0	δ_0	λ_1	μ_1	δ_1	π_1	θ	Error	TPR	FPR	Error	TPR	FPR	Error	TPR	FPR
Example 1																	
500	0.07	0.04	3.26	0.22	0.04	2.82	0.021	4.82	9.36	87.59	8.05	54.03	9.95	38.59	17.44	79.27	16.03
1000	0.05	0.03	2.35	0.16	0.03	2.09	0.015	3.50	9.37	87.53	8.04	54.01	9.99	38.58	16.55	80.76	15.39
2000	0.03	0.02	1.74	0.11	0.02	1.45	0.010	2.51	9.37	87.51	8.03	53.98	10.04	38.55	15.52	82.46	14.66
5000	0.02	0.01	1.06	0.07	0.01	0.91	0.007	1.56	9.38	87.51	8.05	54.00	10.00	38.57	14.65	83.91	14.04
Example 2																	
500	0.05	0.04	3.58	0.16	0.05	1.32	0.021	4.06	12.08	86.58	11.50	28.06	53.23	20.04	16.59	80.69	15.42
1000	0.03	0.03	2.59	0.11	0.03	0.91	0.015	2.91	12.00	86.72	11.45	28.07	53.21	20.05	16.79	80.35	15.57
2000	0.02	0.02	1.93	0.08	0.02	0.65	0.010	2.16	11.97	86.62	11.36	27.99	53.34	20.00	16.94	80.11	15.67
5000	0.02	0.01	1.20	0.05	0.01	0.40	0.007	1.34	11.97	86.71	11.40	27.99	53.35	19.99	17.01	79.99	15.72
Example 3																	
500	0.05	0.04	3.24	0.14	0.03	1.17	0.014	3.67	8.15	90.58	7.60	27.97	53.39	19.98	15.12	83.13	14.37
1000	0.03	0.03	2.32	0.10	0.02	0.78	0.010	2.60	8.14	90.58	7.60	28.03	53.29	20.02	15.55	82.41	14.68
2000	0.02	0.02	1.74	0.07	0.02	0.57	0.007	1.94	8.08	90.65	7.53	28.03	53.28	20.02	15.93	81.79	14.95
5000	0.01	0.01	1.05	0.04	0.01	0.38	0.004	1.19	8.09	90.72	7.59	28.01	53.31	20.01	16.20	81.33	15.15
Example 4																	
500	0.12	0.14	7.65	0.34	0.21	12.20	0.192	16.72	35.32	58.28	32.58	43.07	28.21	30.77	34.73	50.44	28.38
1000	0.09	0.10	4.94	0.23	0.13	7.29	0.152	10.04	34.35	61.10	32.40	43.09	28.19	30.78	34.66	50.56	28.33
2000	0.05	0.06	2.94	0.15	0.09	4.69	0.100	6.14	33.42	62.03	31.48	43.16	28.07	30.83	34.47	50.88	28.20
5000	0.03	0.03	1.66	0.08	0.05	2.87	0.059	3.59	33.22	62.37	31.33	43.09	28.19	30.78	34.50	50.83	28.21
Example 5																	
500	-	-	-	-	-	-	-	-	12.40	88.43	12.75	24.85	58.59	17.75	18.01	78.31	16.44
1000	-	-	-	-	-	-	-	-	12.35	88.81	12.85	24.84	58.60	17.74	18.30	77.84	16.64
2000	-	-	-	-	-	-	-	-	12.40	88.94	12.97	24.86	58.57	17.75	18.34	77.76	16.67
5000	-	-	-	-	-	-	-	-	12.39	89.05	13.01	24.87	58.55	17.76	18.39	77.68	16.71

proposed by [1] and the method of Dempster-Shafer Theory (DST) developed by [6]. To conduct the BD method, we follow [1] and calculate the click intensity (i.e., number of clicks over the whole time period) for each user. Then we treat users with the top 30% highest intensities to be fraudulent; see more details in [1]. For the method of DST, probability mass functions are pre-specified for the number of clicks, the number of PVs, and time difference. As a result, the malicious probability for each visitor can be calculated according to the Dempster's Rule. Lastly, we follow [6] to set the fraudulent probability as 0.35. The comparison is implemented by the same simulation datasets for all those five examples. For both methods, measurements such as Error, TPR and FPR can also be evaluated in a similar manner as for the EM algorithm.

3.3 Simulation results

For each experiment setting, we repeat our simulation for $R = 1{,}000$ times. The initial values are randomly selected. Simulation results are given in Table 1. First of all, for all the parameters, the MAE steadily decreases towards 0 as n gets larger across all the examples. Accordingly, the overall MSE for θ declines as n increases. This indicates that the consistency of the estimators can be achieved by our proposed method. Meanwhile, when the two types of visitors are quite different (i.e., Example 3), the misclassification rate is only 8%. This leads to excellent TPR and FRP performances. However, if a malicious visitor can not be easily distinguished from a regular one (i.e., Example 4), we get a relatively large misclassification rate up to nearly 34%. As a result, the TPR in Example 4 is smaller than that of other examples and the corresponding FPR is the highest. Furthermore, it is noteworthy that in Example 5, the misclassification rate is roughly 12%, which implies that our method is reasonably well-performed even the true distribution of data is different from assumptions. Our proposed method has the merit of robustness. At last, it can be seen that our method outperforms the DST and BD methods in the sense of misclassification rate. This implies that our method has a reliable capability for detecting fraudulent visitors.

3.4 A real application

To illustrate the practical value of the proposed method, we present here a real example of search engine marketing (SEM). Our data is provided by a popular search engine in China. The dataset contains a total of 1,115 visitors' online behaviors from September 4th, 2014 to September 30th, 2014. For our research purpose, we only calculate and keep

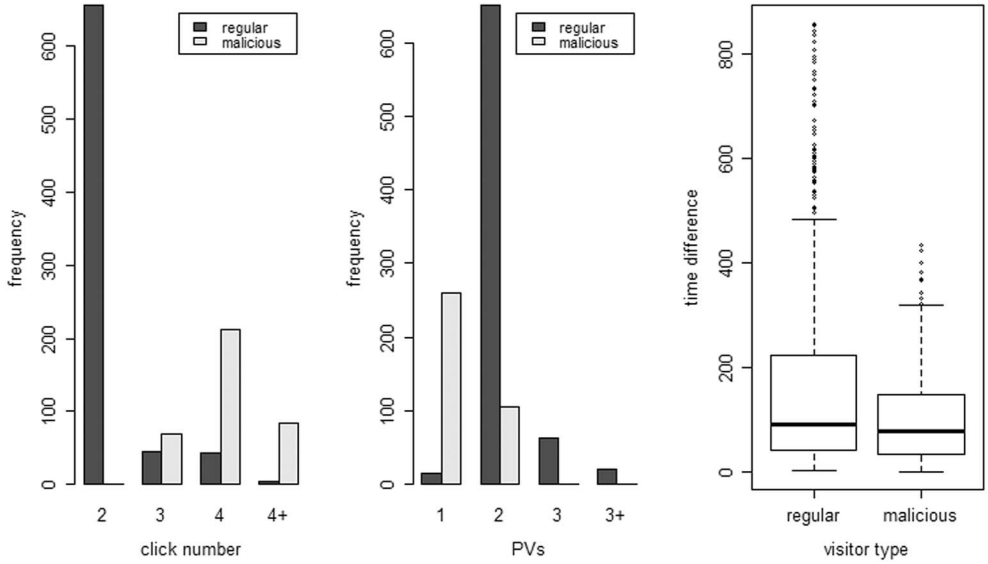

Figure 1. The number of click, number of PVs, and time difference for regular and malicious visitors by the EM algorithm.

Table 2. Estimation Result for SEM dataset

Visitor Type	Parameter Estimation			
	π	λ	μ	δ
Regular	0.677	0.243	0.911	318.802
Malicious	0.323	2.350	0.153	128.264

Table 3. Comparison of Fraud Detection Methods for SEM dataset

	Malicious				Regular		
Method	Ratio	Clicks	PVs	Time	Clicks	PVs	Time
EM	0.33	4.41	1.13	135.27	2.20	1.94	305.96
BD	0.30	2.79	1.56	46.02	2.98	1.72	337.50
DST	0.35	4.39	1.22	133.18	2.13	1.92	312.73

the information required by our model, i.e., each visitor's number of clicks (C_i), total number of PVs (P_i) and the accumulated time difference (Δ_i).

We then apply the EM algorithm to this dataset. The initiative values are randomly selected because the estimation results are insensitive to that. Table 2 presents parameter estimation for two types of visitors. Recall that λ_k, μ_k and δ_k represent the expected number of clicks, number of PVs and time difference respectively for different types of visitors. It can be seen from Table 2 that regular visitors have fewer number of clicks, larger number of PVs and longer time difference than the malicious users. In order to see whether our method can effectively distinguish those two types of visitors, we further calculate the posterior malicious probability for each user. After setting the cut-off value τ to be 0.3, 366 visitors are predicted to be malicious by our method, which accounts for 32.83% of all the visitors.

We also apply the BD and DST methods to this dataset for comparison. The malicious ratio is set to be 30% and 35% for this two methods. In Table 3, we present the mean of number of clicks, number of PVs, and time difference for two types of visitors, which result from all methods. On one hand, it can be found that the regular and malicious visitors detected by both the EM and DST algorithms exhibit great differences in the sense of those three variables. However, the parameters for the DST method have to be pre-specified rather than estimated from the data, which is the main advantage of our proposed algorithm. On the other hand, the malicious visitors identified by the BD method are very similar to the regular ones in the number of clicks and the number of PVs.

Figure 1 provides more detailed comparison between the regular and malicious visitors. From the left panel, we can see that most regular visitors only click twice during the time span of our dataset. However, most malicious users make 4 and more clicks. The middle panel shows that regular visitors view 2 or more pages per click. In contrast, most malicious visitors only browse one web page. The right panel presents boxplots of time difference for two kinds of visitors. As expected, time difference of malicious visitors is much smaller than that of the regular users.

4. CONCLUSION

In this paper, we develop a statistical model for click fraud detection. Each visitor is assumed to accompany with a latent indicator, labeling whether the user is fraudulent or not. Three variables are considered to estimate the visitor's fraudulent probability: number of clicks, number of PVs and time difference between consecutive clicks respectively. According to the visitor's type, the corresponding variables are

assumed to follow the same distributions but with different parameters. We then suggest an EM algorithm to get maximum likelihood estimators and the malicious probability for each visitor is also calculated.

Our work can be further extended in the following directions. Firstly, we develop a unsupervised method for click fraud detection. This suits the situation that the variable of interest is latent. Once the dependent variable is available, we may switch to a semi-supervised or supervised scenario. Secondly, the information considered in this work is very limited. More possible variables could be added in the model and higher accuracy may be achieved. For instance, the time length spent on browsing webpages, the historical records of online browsing or consuming behaviors, and many others. We may surely improve fraud click forecasting with accumulation of information, but the estimation process and computational cost could be heavy due to the increase of complexity. We hope to examine the high-dimensional properties and enhance computational efficiency in the future. Finally, in the real world, fraudulent scenarios could be more complicated. It is possible that many visitors share one IP address, or one visitor access a website via different IP addresses. To solve this problem, we need more information, such as cookies, to help us and identify these users.

ACKNOWLEDGEMENTS

The authors thank the editor, associate editor, and two referees for their insightful comments that have led to significant improvement of this article.

Received 5 March 2015

REFERENCES

[1] Antoniou, D., Paschou, M., Sakkopoulos, E., Sourla, E., Tzimas, G., Tsakalidis, A. and Viennas, E. (2011). Exposing click-fraud using a burst detection algorithm. *Computers and Communications (ISCC), 2011 IEEE Symposium* 1111–1116.

[2] Bardsley, P. and Chambers, R. (1984). Multipurpose estimation from unbalanced samples. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **33** 290–299.

[3] Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* **39** 1–38. MR0501537

[4] Dave, V., Guha, S. and Zhang, Y. (2013). ViceROI: catching click-spam in search ad networks. *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security* 765–776.

[5] Deng, H., King, I. and Lyu, M. R. (2009). Entropy-biased models for query representation on the click graph. *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* 339–346.

[6] Hewett, R. and Agarwal, A. (2013). Protecting advertisers against click frauds. *IT Convergence and Security 2012* **215** 61–69.

[7] Metwally, A., Agrawal, D. and El Abbadi, A. (2007a). Detectives: detecting coalition hit inflation attacks in advertising networks streams. *Proceedings of the 16th International Conference on World Wide Web* 241–250.

[8] Metwally, A., Agrawal, D., El Abbadi, A. and Zheng, Q. (2007b). On hit inflation techniques and detection in streams of web advertising networks. *Distributed Computing Systems, 2007. ICDCS'07. 27th International Conference* 52–52.

[9] Shafer, G. (1976). *A mathematical theory of evidence*, Princeton University Press, Princeton. MR0464340

[10] Yu, F., Xie, Y. and Ke, Q. (2010). Sbotminer: large scale search bot detection. *Proceedings of the Third ACM International Conference on Web Search and Data Mining* 421–430.

[11] Zhang, L. and Guan, Y. (2008). Detecting click fraud in pay-per-click streams of online advertising networks. *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference* 77–84.

Xuening Zhu
Guanghua School of Management
Peking University
Beijing
China
E-mail address: xueningzhu@pku.edu.cn

Da Huang
School of Management
Fudan University
Shanghai
China
E-mail address: dahuang@fudan.edu.cn

Rui Pan
School of Statistics and Mathematics
Central University of Finance and Economics
Beijing
China
E-mail address: panrui_cufe@126.com

Hansheng Wang
Guanghua School of Management
Peking University
Beijing
China
E-mail address: hansheng@gsm.pku.edu.cn