Random forests for the detection of click fraud in online mobile advertising

Daniel Berrar BERRAR.D.AA@M.TITECH.AC.JP
Interdisciplinary Graduate School of Science and Engineering
Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan

Editor: Unknown

Abstract

Click fraud is a serious threat to the pay-per-click advertising market. Here, we analyzed the click patterns associated with 3081 publishers of online mobile advertisements. The status of these publishers was known to be either fraudulent, under observation, or honest. The goal was to develop a model to predict the status of a publisher based on its individual click profile. In our study, the best model was a committee of random forests with imbalanced bootstrap sampling. The average precision was 49.99% on the blinded validation set and 42.00% on the blinded test set. Our analysis also revealed interesting discrepancies between the predicted and actual status labels.

Keywords: click fraud; online mobile advertising; click profiles; random forests

1. Problem formulation

In pay-per-click online advertising, an advertiser provides a commissioner with its advertisement. The commissioner then makes a contract with a publisher that displays the advertisement on web sites. As the publisher's revenue depends on the number of clicks per advertisement, there is an incentive for dishonest publishers to generate clicks artificially, either by employing people to click on the advertisement or by using computer scripts that simulate human click behavior. This type of internet crime is called *click fraud*, and it is a serious threat for the pay-per-click advertising market (Immorlica et al., 2005). It is therefore important to develop algorithms that can monitor a publisher's behavior and reliably predict whether a publisher is likely to be fraudulent.

The goal of the FDMA 2012 data mining challenge[1] was to develop such an algorithm. The training set contained more than 3 million clicks associated with 3081 publishers, which were categorized into honest publishers (OK, 95.1%), publishers under scrutiny (Observation, 2.6%), and fraudulent publishers (Fraud, 2.3%). The provided training set included information about the time stamp of each click, the IP address of the computer that generated the click, the country where that computer was located, etc. Based on this information, predictive models were developed and then applied to a blinded validation set of 3064 publishers. The performance of the models was evaluated based on the average precision AP,

[1] `http://palanteer.sis.smu.edu.sg/fdma2012/`

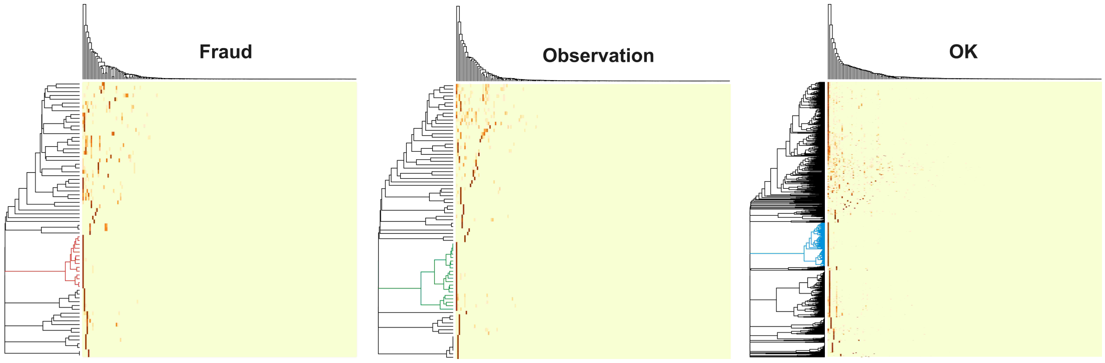

Figure 1: Geo tracking of clicks. Columns show the 197 countries of origin; rows show the publishers from the training set. For each publisher, the percentage of clicks from each country is color-coded. Darker colors reflect higher percentages; yellow is 0% and dark red is 100%. Rows and columns are clustered based on complete linkage. For the publishers with status OK, only those publishers associated with at least 50 clicks are shown.

$$AP = \frac{1}{m} \sum_{i=1}^{k} precision(i) \tag{1}$$

where $precision(i)$ denotes the fraction of correct fraud predictions up to the position i in the list of predictions, where k publishers are ordered based on decreasing values of the probability of fraud; m is the number of publishers that are actually labeled as being fraudulent. The final evaluation of the models was based on the average precision on a blinded test set comprising the data of 3000 publishers.

2. Data pre-processing and feature extraction

2.1 Basic attributes for each publisher

For each publisher, we considered the following basic attributes: (*i*) total number of clicks; (*ii*) number of clicks from the same computer (inferred from attribute **iplong**); (*iii*) distinct IP addresses (inferred from attribute **iplong**); (*iv*) distinct parts of the IP addresses; (*v*) publisher's channel type (inferred from attribute **category**); (*vi*) phone models used by clickers (inferred from attribute **agent**); (*vii*) advertisement campaign (inferred from attribute **cid**); and (*viii*) number of clicks from different countries (*geo tracking*; inferred from attribute **cntr**).

Click fraud, notably manual click fraud, is known to correlate with the geographical location of the clicker. The heatmaps in Figure 1 visualize the geo tracking of clicks from the training set with respect to the status of the publisher. For all three groups (Fraud, Observation, and OK), the clicks originate from only a few countries (cf. columns). For about half of the publishers in each group (cf. rows), the associated clicks are distributed across these countries, whereas for about the other half, the majority of clicks come from only a very small number of countries. In fact, in all three groups, we find such clusters.

Geo tracking alone, however, is not reliable for click fraud detection. There may be various reasons that could explain the observed clusters, such as, obviously, the type and the target of the advertisements. Furthermore, malicious scripts could generate fraudulent clicks, and these scripts could run on computers in different geographical locations. In that case, we might fail to detect any clusters.

2.2 Click profiles

For each publisher and each unique IP address, we investigated the *click profile*, i.e., the time delay between consecutive clicks. For the majority of fraudulent publishers in the training set, we observed that the number of unique IP addresses was below 3000. Only for two fraudulent publishers, we observed that clicks were coming from more than 3000 unique IP addresses. To derive the click profile, we discarded all publishers for which we observed clicks coming from more than 3000 unique IP addresses. This approach was of course far from being ideal, but it reduced the computational time considerably.[2]

2.2.1 Long click profile

We assumed that many consecutive clicks from the same IP address in short time intervals were suspicious. So for each publisher, we counted how many clicks from the same IP address occurred each day in less than 5s, between 5s and 10s, between 10s and 20s, between 20s and 30s, and so on up to the interval > 300s. Furthermore, we required that at least 10 clicks must have come from each IP address. Table 1 shows an example of a set of consecutive clicks from the same IP address for the publisher 8ih09.

Table 1: Example set of 25 consecutive clicks from the same IP address for publisher 8ih09 (data from training set).

id	agent	cid	cntr	date	timeat	cateogry	referrer	same URL	gap
14090783	Opera_Mini	8flxe	fr	2012-02-09	08:53:21.0	in	24940f5c4q688oc8	0	0
14096272	Opera_Mini	8flxd	fr	2012-02-09	09:02:01.0	in	24940f5c4q688oc8	1	520
14096576	Opera_Mini	8flyo	fr	2012-02-09	09:02:30.0	in	24940f5c4q688oc8	1	29
14135449	Opera_Mini	8flyo	fr	2012-02-09	09:59:33.0	in	24940f5c4q688oc8	1	3423
14149730	Opera_Mini	8flyo	fr	2012-02-09	10:18:31.0	in	24940f5c4q688oc8	1	1138
14153291	Opera_Mini	8flyp	fr	2012-02-09	10:23:32.0	in	14qhcdsqvou88kos	0	301
14153584	Opera_Mini	8flyu	fr	2012-02-09	10:23:57.0	in	14qhcdsqvou88kos	1	25
14154864	Opera_Mini	8flyp	fr	2012-02-09	10:25:42.0	in	24940f5c4q688oc8	0	105
14197361	Apple_iPhone	8flyo	fr	2012-02-09	11:23:23.0	in	3gza50jfnzcw44wc	0	3461
14197602	Apple_iPhone	8jdc9	fr	2012-02-09	11:23:42.0	in	14qhcdsqvou88kos	0	19
14198413	Apple_iPhone	8flxc	fr	2012-02-09	11:24:50.0	in	14qhcdsqvou88kos	1	68
14198584	Apple_iPhone	8flyp	fr	2012-02-09	11:25:05.0	in	14qhcdsqvou88kos	1	15
14199113	Apple_iPhone	8flys	fr	2012-02-09	11:25:51.0	in	14qhcdsqvou88kos	1	46
14201181	Apple_iPhone	8flxe	fr	2012-02-09	11:28:31.0	in	14qhcdsqvou88kos	1	160
14206726	Apple_iPhone	8flxf	fr	2012-02-09	11:35:50.0	in	23ge85exom8084s0	0	439
14217945	Apple_iPhone	8flyu	fr	2012-02-09	11:50:32.0	in	23ge85exom8084s0	1	882
15754245	Opera_Mini	8gpd5	fr	2012-02-10	10:18:52.0	in	3cu2xmfag82sosk4	0	0
15764598	Opera_Mini	8gpd5	fr	2012-02-10	10:33:56.0	in	4jyefurnmxkwoo4w	0	904
15768527	HTC_Vision	8gdka	fr	2012-02-10	10:39:47.0	in	1rbl5y69ej34gg8w	0	351
15768829	HTC_Vision	8gpd5	fr	2012-02-10	10:40:17.0	in	3cu2xmfag82sosk4	0	30
15777019	Opera_Mini	8gpd5	fr	2012-02-10	10:52:51.0	in	3cu2xmfag82sosk4	1	754
783581	Opera_Mini	8gpd5	fr	2012-02-11	10:34:38.0	in	1rbl5y69ej34gg8w	0	0
901789	SPH-P100	8gpd5	fr	2012-02-11	13:01:12.0	in	1rbl5y69ej34gg8w	1	8794
902642	SPH-P100	8gdka	fr	2012-02-11	13:02:09.0	in	3cu2xmfag82sosk4	0	57
903031	SPH-P100	8gpd5	fr	2012-02-11	13:02:38.0	in	3cu2xmfag82sosk4	1	29

[2] Computational time was a critical factor because of the time limits of the competition.

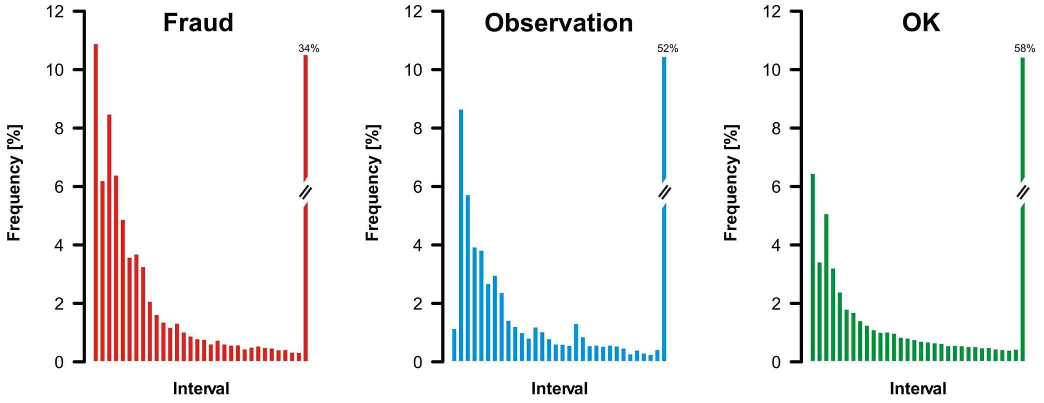

Figure 2: Click frequency per interval based on all long click profiles. Intervals are $[0s, 5s],]5s, 10s],]10s, 20s], ...]300s, +\infty)$.

For the clicks in Table 1, we see the following interval frequencies:
$(0, 0, 2, 4, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11)$, where the i^{th} element represents the count for the i^{th} interval. For example, we observe two clicks with a time gap between 10s and 20s (id 14197602 with 19s and id 14198584 with 15s).[3] For each publisher, we proceeded analogously for all IP addresses and derived the cumulative interval frequency count, which is the *long click profile.* For the publisher 8ih09, for instance, this profile is
$(0, 0, 17, 25, 26, 30, 20, 21, 19, 14, 7, 5, 9, 9, 7, 2, 7, 8, 4, 3, 7, 3, 3, 2, 1, 1, 2, 3, 1, 1, 1, 41)$, where the i^{th} element represents the cumulative count for the i^{th} interval. We see that clicks from the same IP address tend to occur in relatively short sequences for 8ih09, which is in fact labeled as Fraud. By contrast, let us consider the profile of a publisher labeled as OK, for example, 8i7wi. Its long click profile is
$(0, 0, 4, 3, 0, 0, 2, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1, 1, 0, 0, 76)$; here, short click sequences are less frequent. Another publisher, 8jkh1, is labeled as Observation, and its long click profile is
$(6, 13, 11, 19, 22, 8, 13, 13, 21, 17, 13, 14, 11, 7, 12, 9, 5, 8, 7, 7, 5, 8, 1, 5, 6, 5, 2, 1, 2, 2, 5, 404)$.

Figure 2 shows the click frequencies per interval, derived from all long click profiles per group. We see that, overall, consecutive clicks that follow one another rather quickly occur more often for fraudulent publishers than for those with status Observation or OK.

2.2.2 Short click profile

The short click profile was derived in the same way as the long click profile, except that at least 5 (and not 10) consecutive clicks must have come from the same IP address. As an example, Table 2 shows a set of six consecutive clicks from the same IP address for the publisher 8ih09.

[3] We also kept track of the relative counts, that is, we divided the interval counts by the respective set size.

Table 2: Example set of six consecutive clicks from the same IP address for publisher 8ih09 (data for training set).

id	agent	cid	cntr	date	timeat	gap
14174651	HTC_Vision	8gkyx	fr	2012-02-09	10:53:31.0	0
14174832	HTC_Vision	8gp4w	fr	2012-02-09	10:53:45.0	14
14175578	HTC_Vision	8gdka	fr	2012-02-09	10:54:46.0	61
14175753	HTC_Vision	8gkyx	fr	2012-02-09	10:55:01.0	15
14175923	HTC_Vision	8gdka	fr	2012-02-09	10:55:16.0	15
14211056	Apple_iPhone	8flxc	fr	2012-02-09	11:41:33.0	2777

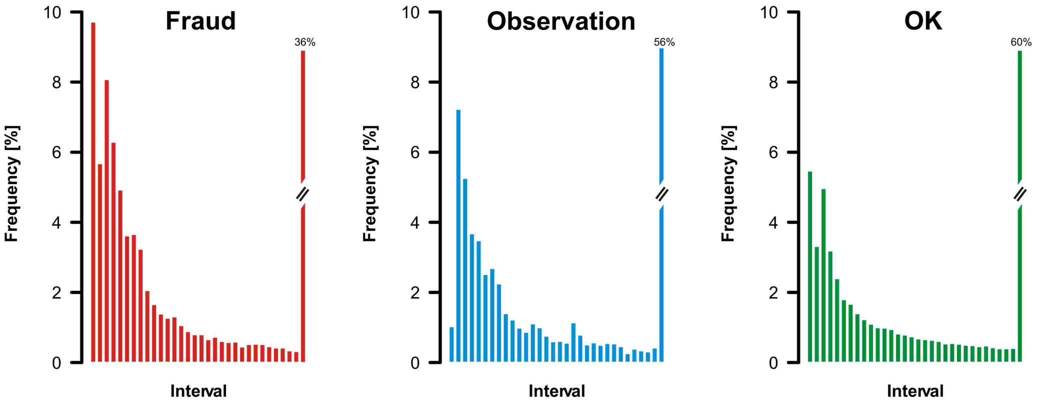

Figure 3: Click frequency per interval based on all short click profiles. Intervals are $[0s, 5s],]5s, 10s],]10s, 20s], ...]300s, +\infty)$.

For the clicks shown in Table 2, we observe the following interval frequencies: $(0, 0, 3, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$, where the i^{th} element represents the count for the i^{th} interval. For example, we observe three clicks with a time gap between 10s and 20s. We proceeded analogously for all publishers and all unique IP addresses and derived the *short click profile* as the cumulative count. For publisher 8ih09, for example, the short click profile is $(0, 0, 20, 25, 26, 30, 20, 22, 19, 14, 7, 5, 9, 10, 7, 2, 7, 8, 4, 3, 7, 3, 3, 2, 1, 1, 2, 3, 1, 1, 1, 44)$. By contrast, consider the profile of a publisher labeled as OK, for example, 8i7wi. Its short click profile is $(0, 0, 7, 4, 2, 0, 2, 1, 0, 2, 2, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 2, 1, 1, 2, 0, 0, 152)$; here, short sequences are less frequent. The publisher 8jkh1 is under observation, and its short click profile is $(7, 15, 17, 21, 23, 11, 13, 14, 21, 20, 14, 17, 12, 7, 13, 9, 6, 8, 9, 9, 5, 8, 2, 6, 8, 6, 2, 1, 2, 4, 6, 557)$.

Figure 3 shows the click frequencies per interval, derived from all short click profiles per group. We observe that, overall, quick consecutive clicks occur more often in fraudulent publishers than in those with status Observation or OK.

2.2.3 Profile of clicks coming from the same url

The long and short click profiles ignored the URL where an advertisement had been clicked on. It is certainly plausible, however, that a fraudulent (human) clicker does not navigate too often from one web site to another. To derive a pattern of clicks coming from the same URL, we therefore asked for each publisher: how many clicks came from the same IP address *and* the same URL in less than 5s, between 5s and 10s, between 10s and 20s, and so on up to the interval > 300s. At least 5 clicks must have come from each IP address. A problem with this approach, however, was that the information about the URL was missing for many clickers.

Consider again Table 1. The column **referrer** contains encrypted information about the URL. The column **same URL** contains a flag, indicating whether the clicker has left ($= 0$) or stayed ($= 1$) on the same URL. We considered only those time gaps that refer to the same URL; thus, we ignored the gap of 19s for **id** 14197602, for example, because here, the clicker has navigated from 3gza50jfnzcw44wc to 14qhcdsqvou88kos.

For the example shown in Table 1, we observe the following interval frequencies: $(0, 0, 1, 3, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6)$. We proceeded analogously for all publishers and all unique IP addresses and derived the click profile as the cumulative count. For publisher 8ih09 (Fraud), for example, the click profile is $(0, 0, 14, 15, 9, 16, 10, 11, 10, 6, 3, 2, 6, 7, 0, 1, 4, 1, 1, 1, 1, 2, 0, 1, 0, 0, 1, 0, 1, 0, 1, 14)$. The publisher 8i7wi (OK) has the profile $(0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 3)$, and the publisher 8jkh1 (Observation) has the profile $(3, 3, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3)$.

Figure 4 shows the click frequencies per interval, derived from all click profiles from identical URLs. Again, we observe that, overall, quick consecutive clicks occur more often in fraudulent publishers than in those with status Observation or OK.

2.2.4 Redflag

For each publisher, we checked if there were at least 5 clicks from the same IP address and the same URL and with a time gap of less than 20s. If so, we incremented a flag (*redflag*) for that publisher. In Figure 5, publishers from the training set are ranked from left to right based on decreasing values of redflag. Fraudulent publishers (red) and those under observation (blue) are concentrated towards the left hand side. Thus, the larger redflag, the more suspicious is the publisher. Redflag is in fact a significant indicator of fraudulent behavior ($P < 0.001$, Kruskal-Wallis test).

3. Methods used and experimental configuration

According to the rules of the competition, each team was allowed to submit the predictions of two models for the final evaluation. Below, we describe our two models; all experiments were carried out in R (R Development Core Team, 2009).

3.1 Ensemble of random forests built on reduced data set

For the first model, we used only the basic attributes and the long click profile. The algorithm was random forests (Breiman, 2001), which first generates a number of unpruned

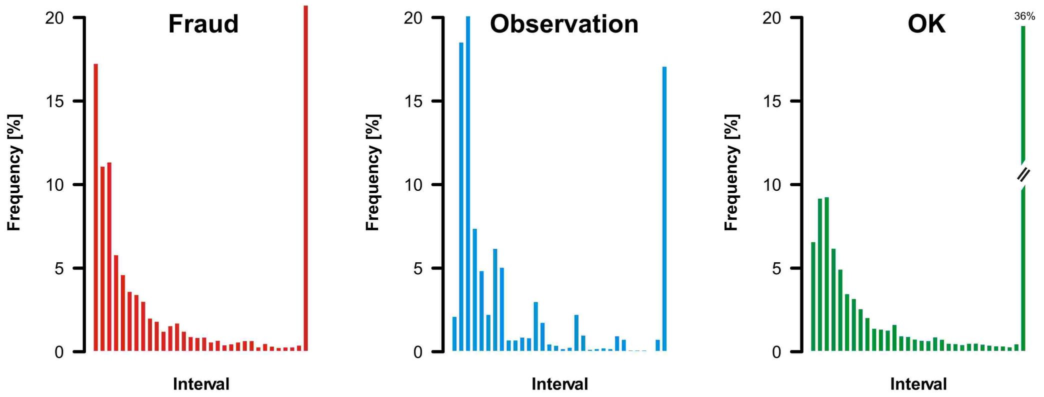

Figure 4: Click frequency per interval based on all click profiles from the same URL. Intervals are $[0s, 5s],]5s, 10s],]10s, 20s], ...]300s, +\infty)$.

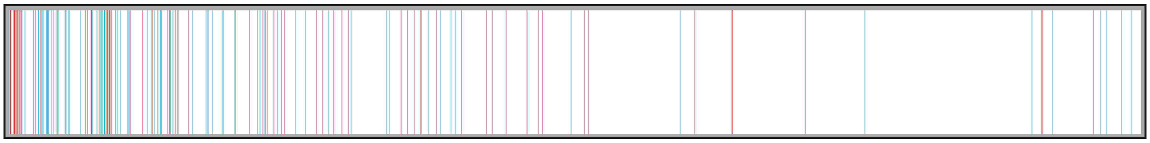

Figure 5: Publishers (from the training set) ranked from left to right based on decreasing values of redflag. Red bars denote publishers with status Fraud; blue bars denote publishers with status Observation; white bars denote publishers with status OK. Fraudulent publishers and those under observation are significantly concentrated towards the left hand side ($P < 0.001$, Kruskal-Wallis test).

decision trees (here, classification and regression trees) from bootstrap samples of the training set. Each tree uses only a random selection of the available attributes. Second, the algorithm combines the trees into one committee ("forest") whose predictions result from aggregating the predictions of the individual trees.

Because of the drastic class imbalance in the training set (Fraud:Observation:OK = 72:80:2929), either cost-sensitive learning or up/down-sampling strategies are necessary. For random forests, both approaches were shown to be on par in terms of performance (Chen et al., 2004). Up/down-sampling, however, is computationally less expensive for largely imbalanced data sets because each tree uses only a small subset of the training set. Given the time constraints, we adopted only one approach: up/down-sampling. A multitude of sample sizes were tested, and the models were evaluated and selected based on the out-of-bag (OOB) error rate.

Our preliminary results suggested that the differences between the predictive performance of the models were not so large. From all models, we finally selected seven (Table 3) and combined them into one *ensemble of random forests*. This is model #1, and it was submitted for the final evaluation.

Table 3: Individual random forests with up- and down-sampling (in % from the respective classes; ntree: number of trees in each forest; nodesize: number of terminal nodes in each tree).

#	**ntree**	**nodesize**	**Fraud**	**Observation**	**OK**	**OOB error** [%]
1	250	5	90	63	41	4.64
2	250	5	90	63	41	4.67
3	250	3	83	75	51	4.64
4	250	3	69	38	34	4.48
5	250	3	69	38	34	4.74
6	250	3	90	44	51	4.54
7	250	4	83	63	68	4.45

3.2 Random forest built on extended data set

In addition to the reduced data set, we included the following data for our second model: short click profile, click profile from identical URLs, and redflag.

The algorithm was random forest with up/down-sampling. We tested again various parameter settings (number of trees, terminal node size, sampling ratios) and selected the final model on the basis of the OOB error rate. The final model consisted of 50 trees with a terminal node size of 3. The percentages for the bootstrap samples were: 97% for class Fraud, 88% for class Observation, and 61% for class OK. This is model #2, and it was submitted for the final evaluation.

3.3 Evaluation of click profiles

How well can the three click profiles discriminate the publishers? To address this question, we trained three random forests, each using only one of these click profiles and no further data. Each model consisted of 250 trees, each with 3 terminal nodes. The sampling was 90% for Fraud, 75% for Observation, and 61% for OK. Table 4 shows the classification results of these three models (not submitted for the final evaluation).

Table 4: Classification results [%] of random forests using only click profiles.

Profile used	OOB error	avg precision (validation)
long	4.61	47.67
short	4.64	46.77
Clicks from same URL	4.51	47.89

4. Results and key insights on fraudulent behavior

4.1 Prediction results

Table 5 shows the classification results of the two models that were submitted for the evaluation on the final test set.

Table 5: Classification results [%] of the final two models.

Model #	OOB error	avg precision (validation)	avg precision (test)
1	4.61	49.99	42.00
2	3.66	51.55	36.94

Model #2 (single random forest, 50 trees) achieved a better performance on both the training and the validation set. However, the performance on the final test remarkably deteriorated, compared with the performance of model #1 (ensemble of random forests, 1750 trees) that used only the reduced data set.

4.2 Key insights on fraudulent behavior

Two problems made this competition particularly challenging.

First, there is the problem of concept drift. The three data sets came from different time windows: the training set contains data from 9-11 February 2012, the validation set contains data from 23-25 February 2012, and the test set contains data from 8-10 March 2012. A publisher may appear across different sets with a similar pattern, but the provided, actual status label may be different. For example, a publisher may have been labeled as OK in early February and then as Observation in late February, perhaps because this publisher showed a suspicious pattern. It is plausible that publishers were first rated as OK, and if they showed a suspicious pattern, they received the status Observation; only if that suspicious pattern persisted, then they might have been labeled as Fraud, eventually. Or a publisher might immediately have received the status Fraud, provided that its click profile was considered highly suspicious. These are speculations, though.

Second, there is no “ground truth” about which publishers are indeed really fraudulent, which are truly OK, and which should be under scrutiny. The real status labels must have come from some fraud detection algorithm (here called “ground model”, for short), but how reliable are its predictions? Consider the following example. For one publisher in the training set, we observed 5706 total clicks. All clicks came from the same IP address, and 3307 (58%) occurred in less than 5s...also, is it not suspicious that 1086 consecutive clicks (19%) have an interval of even 0s? It is tempting to speculate that a script generated these clicks. Surprisingly, the status of that publisher is OK in the training set. In the validation set, the click profiles of that publisher may also raise some eyebrows: more than 70% of all 1149 clicks – from one and only IP address – occurred in less than 5s, with 248 clicks having a time gap of 0s. As the validation set is blinded, we cannot know whether the status of that publisher has changed or not. But by manually changing and then resubmitting the predictions of our models, we could infer that this publisher cannot be labeled as Fraud. Its status may or may not have changed to Observation, though. Several similar examples of questionable status labels could be found.

5. Concluding remarks

What makes a publisher suspicious or even fraudulent? This question does not have a clear answer. Particularly, apart from the fraudulent intent to increase payment, there may be various reasons for a suspicious click profile.

In our analysis, we were often surprised by the discrepancies between predicted and actual status labels from the “ground model”. Let us assume for a moment that our developed model could achieve 100% average precision on both the validation and the test set. Our model would then most certainly merely replicate the decision rules of the “ground model”. Well, in the framework of the competition, that would of course be marvellous. But then our model could not give any new insights into fraudulent behavior, as our *thinking about the problem* must have been nearly the same as that of the designer of the “ground model”. A truly interesting question therefore is: how can we explain the discrepancies between all models?

References

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

C. Chen, A. Liaw, and L. Breiman. Using random forests to learn imbalanced data. *Technical Report #666, Department of Statistics, University of California, Berkeley*, pages 1–12, 2004.

N. Immorlica, K. Jain, M. Mahdian, and K. Talwar. Click fraud resistant methods for learning clickthrough rates. *Proceedings of the 1st Workshop on Internet and Network Economics*, pages 34–45, 2005.

R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2009. URL `http://www.R-project.org`. ISBN 3-900051-07-0.