Feature Engineering for Click Fraud Detection

Clifton Phua CWCPHUA@I2R.A-STAR.EDU.SG
Eng-Yeow Cheu EYCHEU@I2R.A-STAR.EDU.SG
Ghim-Eng Yap GEYAP@I2R.A-STAR.EDU.SG
Kelvin Sim SHSIM@I2R.A-STAR.EDU.SG
Minh-Nhut Nguyen MNNGUYEN@I2R.A-STAR.EDU.SG
starrystarrynight Team
Data Analytics Department (DAD)
Institute for Infocomm Research (I^2R)
*Agency for Science, Technology and Research (A*STAR)*
1 Fusionopolis Way, #21-01 Connexis (South Tower), Singapore 138632

Editor: Vincent van Gogh and Don McLean

Abstract

For our winning entry based on the test dataset, we applied Generalized Boosted regression Models (GBM) to 118 predictive features, which consisted of 67 click behavior, 40 click duplication, and 11 high-risk click behavior features. Our most important data insight, from both domain knowledge and experimentation, is that invalid or fraudulent clicks have particular temporal and spatial characteristics which make them distinguishable from normal clicks. We were surprised that simple statistical features can retain powerful predictive power over time and reduce the chances of over-fitting the training data. We used R for feature engineering and classification (GBM and RandomForest), MySQL for storing the data, and WEKA for trying out alternative classification schemes.

Keywords: Feature Engineering, Feature Selection, Domain Knowledge, Click Fraud Detection, Classification, Generalized Boosted Regression Models

1. Background

Prior to entering this competition in mobile advertising fraud detection, a few of us as data scientists have already done significant work in credit application/transactional fraud detection (Phua et al., 2012) and other related domains (Phua et al., 2010, 2004). Also, we entered the competition because our organization's mission is to support Singapore's economy (BuzzCity Pte. Ltd. is a Singapore company), and this is a competition about a real-world problem where we can benchmark and hone our feature engineering and model building skills on data in the raw form. Our team has learned from other winners of data science competitions (such as Nokia's place category classification (Zhu et al., 2012), KDD Cup 2010 (Yu et al., 2010), and Netflix grand prize (Koren, 2009)) that feature engineering (such as extraction and selection of features) is even more important than model building (such as choice and fine-tuning of algorithms) to achieve the best predictive performance. This competition has been a very exciting and rewarding experience for the team (see the Results section for a better understanding).

2. Problem

The Fraud Detection in Mobile Advertising (FDMA) Competition[1] is organized by Singapore Management University (SMU) in partnership with BuzzCity Pte. Ltd.. Teams were asked to highlight potentially fraudulent publishers (mobile website or application owners), based on privacy enhanced (partially encrypted) training data of publisher description and click behavior. A sample of the two largest publishers of each `status` in training dataset, a "Fraud" and an "OK" publisher, is provided in Table 1 and a sample of three clicks from each publisher is provided in Table 2. There is another "Observation" `status`, consisting of a small number of new publishers or publishers with strange click behavior, which our team had treated as "OK".

partnerid	bankaccount	address	status
8iaxj		14vxbyt6sao00s84	Fraud
8jljr			OK

Table 1: Publisher sample in raw training data. There are missing values in `bankaccount` and `address`. In pay per click online advertising, "Fraud" involves a large number of intentional click charges with no real interest in the advertisements, using automated scripts or click farms. The perpetrators can be the publishers themselves or their competitors, or the competitors of advertisers.

id	iplong	agent	partnerid	cid	cntr	timeat	category	referer
13417867	3648406743	GT-I9100	8iaxj	8fj2j	ru	2012-02-09 00:00:00	ad	26okyx5i82hws84o
13417870	3756963656	Samsung_S5233	8jljr	8geyk	in	2012-02-09 00:00:00	es	15vynjr7rm00gw0g
13417872	693232332	SonyEricsson_K70	8jljr	8gkkx	ke	2012-02-09 00:00:00	es	
13417893	2884200452	Nokia_6300	8jljr	8gp95	vn	2012-02-09 00:00:01	es	
13418096	3648406743	GT-I9100	8iaxj	8fj2m	ru	2012-02-09 00:00:08	ad	24w9x4d25ts00400
13418395	781347853	GT-I9003	8iaxj	8fj2j	ru	2012-02-09 00:00:20	ad	4im401arl30gc0gk

Table 2: Click sample in raw training data. There are missing values in `referer` and `agent`. The raw features include IP address of a clicker (`iplong`), mobile device model used by the visitor (`agent`), campaign ID of a particular advertisement campaign (`cid`), country of the surfer (`cntr`), type of publisher (`category`) which we feel should be tied to a publisher instead of a particular click, or an URL where the ad banner is clicked (`referer`). Other raw features such as browser type, operating system, and carrier information are not provided.

As shown in Table 3, the publishers and clicks data were each split into three sets of training data (for model building), validation data (for ranking on the public leader-board to allow competing teams to know how each other is doing), and test data (for final evaluation and determination of the winners who have not over-fitted to the validation set). Although the data is highly imbalanced - only 2.3 % of all publishers are fraudulent - our team had

1. The FDMA competition website (http://palanteer.sis.smu.edu.sg/fdma2012/) provides description about click fraud, statistics and raw features of advertisement data, and average precision evaluation measure.

reasonable confidence in the predictiveness of our features and did not manipulate the class distribution using minority class over-sampling or majority class under-sampling.

	train set	validation set	test set
date (day of week)	9 to 11 Feb 2012 (Thurs to Sat)	23 to 25 Feb 2012 (Thurs to Sat)	8 to 10 Mar 2012 (Thurs to Sat)
publisher count (fraud %)	3,081 (2.3 %)	3,064 (unknown)	3,000 (unknown)
clicks count (fraud %)	3,173,834 (4.0 %)	2,689,005 (unknown)	2,598,815 (unknown)

Table 3: Data overview.

The FDMA competition had a very short duration of one month from 1 to 30 Sep 2012, with only two submissions per day allowed on the validation set. Average precision (Zhu, 2004) was the scoring metric, and probably chosen by the organizers as it "tends to favor algorithms that detect more hits earlier on". Our team achieved an average precision of 0.5155 on the test set, which was ranked first place in the competition[2]. In the following sections, we describe the final training dataset from feature engineering, use of specific parameters in Generalized Boosted regression Models (GBM), and other details which enabled us to win the competition.

3. Features

The purpose of this section is to show that we created and used an appropriate number and type of features for each publisher.

3.1 Overall

In Figure 1, we show a correlation plot between some of our features including `status`, which we used to ensure feature diversity - by excluding new features which are too similar to existing ones. In Figure 2, using specific parameters GBM described in the next section, we obtained the relative influence or importance of 118 predictive features in the final training dataset (in addition, there are two other features: `partnerid` which is not used for model building and `status` which is the class or dependent feature). Adding all the features' relative influence will sum up to a score of one hundred. On one extreme, there are a few features with relative influence above three. On another extreme, there are a few features with negligible influence on the results such as `category`-related features (see the Potential subsection on a discussion of leveraging the predictiveness of the raw `category` feature). Also, the average relative influence per feature is about 0.88. This final training dataset and other details are available at the first author's homepage (https://sites.google.com/site/cliftonphua/) for researchers to experiment with different classification schemes.

2. Interestingly, our *starrystarrynight* team was ranked fourth with an average precision of 0.5938 on the validation set, visible on the public leader-board.

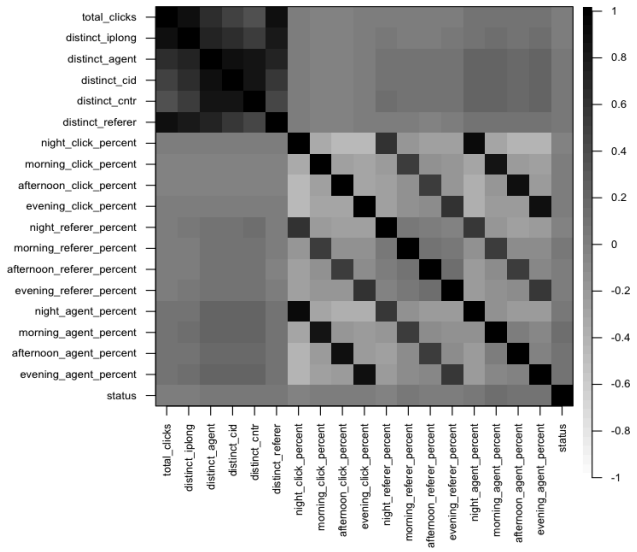


Figure 1: Correlation plot of some click behavior features in final training dataset.

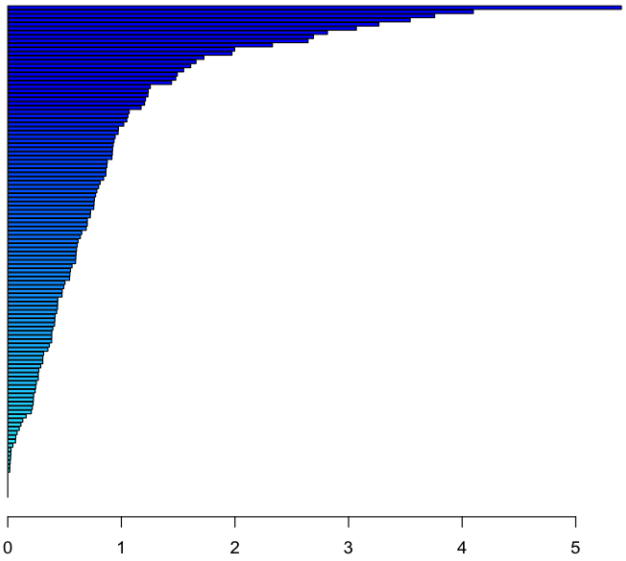


Figure 2: Relative influence of all features in final training dataset.

The 118 predictive features can be grouped into three types of features: 67 click behavior (57%), 40 click duplication (34%), and 11 high-risk click behavior (9%); and their average rank of all their features (based on GBM output) is 69, 35, and 69 respectively - meaning that duplicated clicks are likely to be invalid clicks. We use simple statistical features based on average, standard deviation, and percentages, and none of our features are created directly from `status` or specific values from raw encrypted features, such as `bankaccount`, `address`, `iplong`, `cid`, and `referer`.

3.2 Temporal and Spatial

In Table 4, we list the top-10 features of each type to show that our features capture some temporal and spatial aspects of clicks for each publisher.

Within the one minute interval, fraudulent clicks have significantly more duplicates than normal ones. For click duplication features, the shorter intervals produce better results after we tested one, five, fifteen, thirty, and sixty minutes intervals using Chao-Shen entropy (Chao and Shen, 2003). Chao-Shen entropy is a non-parametric estimation of Shannon's index of diversity. It combines the Horvitz-Thompson estimator (Horvitz and Thompson, 1952) and the concept of sample coverage proposed by Good (1953) to adjust for unseen observations in a sample. For example, there are multi-feature duplicates such as `avg_spiky_ReAgCnIpCi` (average number of the same referer, agent, cntr, iplong, and cid being duplicated in one minute), as well as single feature duplicates such as `std_spiky_iplong` (standard deviation of iplong being duplicated in one minute).

For our top click behavior and duplication features, we created conditional features based on finer-grained time intervals to better capture temporal dynamics of click fraud behavior.

We divided a day into four six-hour periods: night (12am to 5:59am), morning (6am to 11:59am), afternoon (12pm to 5:59pm), and evening (6pm to 11:59pm). For example, `night_referer_percent` is the number of distinct referers at night divided by the total number of distinct referers, and `night_avg_spiky_referer` is the average number of the same referrer being duplicated within one minute at night. Also, we divided an hour into four fifteen-minute periods: first (0-14), second (15-29), third (30-44), and last (45-59). For example, `second_15_minute_percent` is the number of clicks between 15th to 29th minute divided by total number of clicks.

Fraudulent clicks tend to come from some countries (or finer-grained spatial regions) more than others; for example, businesses in India and Indonesia are hardest hit by fraud (Kroll, 2012). Most of BuzzCity's clicks on mobile advertisements also come from these two countries. We tested the top five, ten, fifteen, twenty, and twenty-five high-risk countries (out of two hundred over countries), and found that the top ten high-risk countries works best. For example, `cntr_in_percent` and `cntr_id_percent` are the percentages of invalid clicks originating from India and Indonesia respectively. The main reason for large numbers of invalid clicks coming from `cntr_sg_percent` or Singapore could be due to BuzzCity's penetration tests being conducted from there.

click behavior			**click duplication**			**high-risk click behavior**		
rank	feature	relative influence	rank	feature	relative influence	rank	feature	relative influence
6	std_per_hour_density	3.12	2	std_spiky_iplong	3.81	1	cntr_id_percent	5.49
12	total_clicks	1.76	3	avg_spiky_ReAgCnIpCi	3.69	7	cntr_sg_percent	2.69
14	brand_Generic_percent	1.71	4	night_ avg_spiky_referer	3.66	49	cntr_other_percent	0.72
15	avg_distinct_referer	1.62	5	avg_spiky_agent	3.29	72	cntr_us_percent	0.44
19	std_total_clicks	1.43	8	avg_spiky_referer	2.67	77	cntr_th_percent	0.39
23	night_ referer_percent	1.19	9	avg_spiky_ReAgCn	2.59	84	cntr_uk_percent	0.34
24	second_ 15_minute_percent	1.18	10	night_ avg_spiky_ReAgCnIpCi	2.49	91	cntr_in_percent	0.26
27	distinct_referer	1.15	11	afternoon_ avg_spiky_ReAgCnIpCi	1.98	103	cntr_ng_percent	0.05
29	std_distinct_referer	1.12	13	afternoon_ avg_spiky_agent	1.75	104	cntr_tr_percent	0.04
30	morning_ click_percent	1.1	16	std_spiky_referer	1.6	106	cntr_ru_percent	0.04

Table 4: Top-10 features by type in final training dataset.

3.3 Potential

In Table 5, we show that there is some potential for an alternative approach using the `category` of each fraudulent publisher. Fraudulent *m*obile *c*ontent and *ad*ult content publishers tend to produce a lot more invalid clicks, especially at night and morning periods, than the other fraudulent publishers. In contrast, fraudulent *e*ntertainment and life*s*tyle and *p*remium *p*ortal publishers produce a lot less invalid clicks, and tend to have relatively more invalid clicks during afternoon and evening periods. We attempted to split the datasets and build models separately by category, but did not have enough time to integrate/normalize the different sets of prediction scores in a meaningful way.

During the last day of competition, we built features which capture each publisher's top-k most frequent `referer`, `agent`, `cntr`, `iplong`, and `cid` values in percentage, where

category	publisher count	fraud clicks (fraud %)	night fraud clicks (fraud %)	morning fraud clicks (fraud %)	afternoon fraud clicks (fraud %)	evening fraud clicks (fraud %)
*ad*ult	10	**47226 (37%)**	**15435 (12%)**	6439 (5%)	**11299 (9%)**	**14053 (11%)**
*m*obile content	**23**	41941 (33%)	13589 (11%)	**9284 (7%)**	9623 (8%)	9445 (7%)
*co*mmunity	12	16411 (13%)	7218 (6%)	3301 (3%)	2612 (2%)	3280 (3%)
*e*ntertainment and lifestyle	14	14433 (11%)	2649 (2%)	3265 (3%)	3573 (3%)	4946 (4%)
*se*arch, portal, services	4	3180 (3%)	682 (1%)	572 (0%)	689 (1%)	1568 (1%)
*p*remium *p*ortal	6	2926 (2%)	351 (0%)	608 (0%)	732 (1%)	904 (1%)
*in*formation	3	893 (1%)	49 (0%)	284 (0%)	428 (0%)	132 (0%)
total	72	127010 (100%)	**39973 (31%)**	23753 (19%)	28956 (23%)	34328 (27%)

Table 5: High risk categories in final training dataset.

$k = 3$. We did not feel confident enough to include these features without finding the optimal k value.

Lastly, as most of our features are continuous, our machine learning algorithms could produce better models from discrete or binary features using discretization, such as information entropy minimization heuristic (Fayyad and Irani, 1992) or equal frequency binning.

4. Methods

Gradient boosting is a machine learning technique used for classification problems with a suitable loss function, which produces a final prediction model in the form of an ensemble of weak prediction decision trees (Friedman, 2000). We used the implementation of Generalized Boosted regression Models (GBM) in R's `gbm` package (Ridgeway, 2007). The final parameters used on the final training dataset for our best average precision on the test dataset are:

distribution "bernoulli" as the loss function - also tested the alternative "adaboost" distribution

n.trees 5000 number of iterations - tested 100 to 5000 decision trees

shrinkage 0.001 learning rate - tested 0.001 to 0.01

interaction.depth 5 levels of depth of each tree - tested 2 to 5

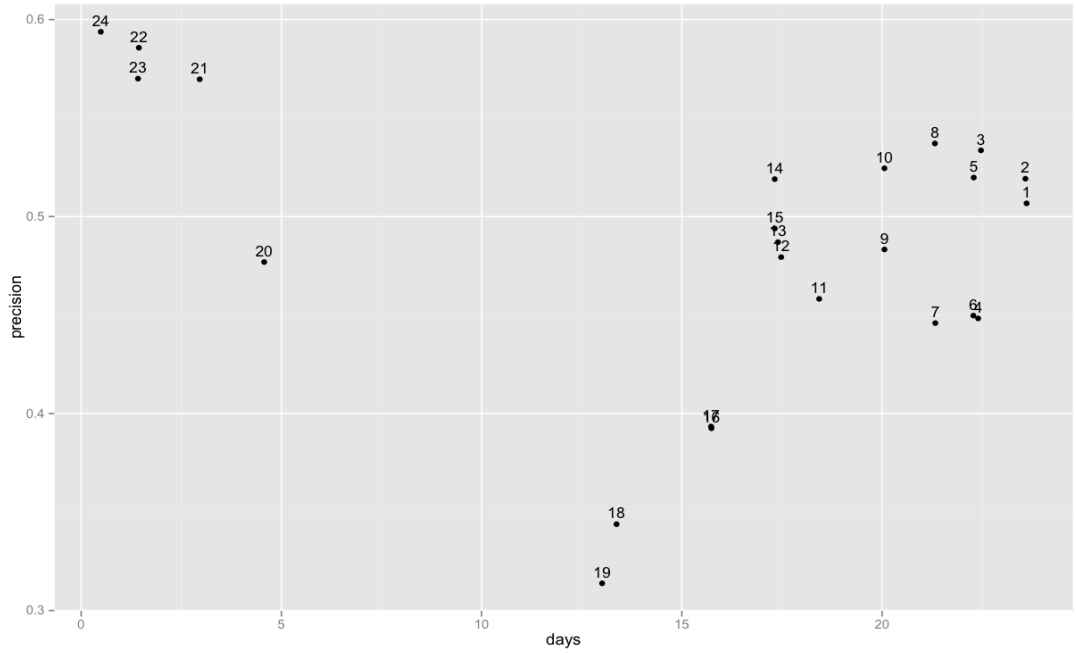
n.minobsinnode 5 minimum observations in terminal node - tested 2 to 5

In the early to mid stages of the competition, when we had many features to experiment with, we used two layers of GBM to select the most important features. During the final stages, we focused on only one layer of GBM as we had identified the three best types of features. When we first started, our team also tried out Random Forest (RF) (Breiman, 2001) (decision trees ensemble algorithm) in R's `randomForest` package, as well as RIPPER (Cohen, 1995) (rule induction algorithm) in WEKA (Hall et al., 2009). As RF and RIPPER did not perform as well as GBM on the validation set, we did not conduct further explorations into them or other classification algorithms. If we did find alternative classification algorithms which perform as well as, if not better, than GBM, we could train a set of base classifiers and combine them with stacking (Wolpert, 1992).

5. Results

In Figure 3, we show all `starrystarrynight` team's 24 submissions' results over time on the validation dataset - to illustrate the uncertainties we faced in the competition. For submissions 1 to 8, as there were no other serious competitors, we topped the leaderboard with an average precision of 0.5371 using 27 click behavior/duplication features and GBM. For submissions 9 to 19, we experimented with various temporal features, with no improvements to average precision. By this time, with about 12 days left in the competition, we had a competition hiatus to attend to various other deadlines. By the time we made submission 20, there was only about 4 days left in the competition and our team had been overtaken by several competitors. For submission 21, we combined all our best features from the last 20 submissions, and improved our average precision to 0.5697. For submission 22, we added the 11 high-risk click behavior features (described in the previous Features section), improved our average precision to 0.5857 with a fourth position on the leaderboard. Submission 23 increased the number of high-risk click behavior features to 34 with no average precision improvement. Our final submission 24, we added conditional features based on finer-grained time intervals (described in the previous Features section) to bump our average precision up to 0.5938, still ranked fourth with only 0.0001 behind `TeamMasdar`.

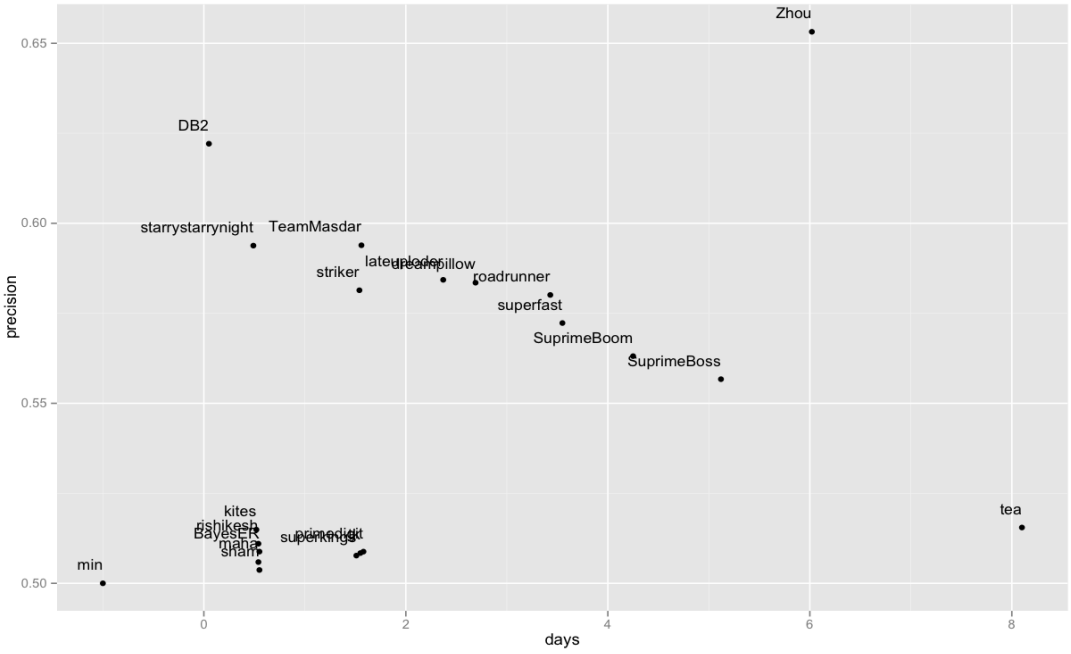
Figure 3: `starrystarrynight` team's average *precision* and their *days* to deadline based on validation data.



In Figure 4, we show the top teams' (>0.5) best average precision on the validation data. The top three teams on the leader-board are `Zhou`, `DB2`, and `TeamMasdar`; with `starrystarrynight` trailing right behind them. The breakaway leader on the validation data `Zhou` had its peak performance six days before the test data submission deadline. Also,

we can observe that many top teams had improved their average precision in the last two days, with DB2 improving its average precision on the leader-board minutes before deadline.

Figure 4: Top teams' best average *precision* and their *days* to deadline based on validation data.



After the leader-board has closed and teams submitted their results based on the test data, `starrystarrynight` is first with 0.5155 average precision, `TeamMasdar` is first runner-up with 0.4642 average precision, and DB2 is second runner-up with 0.4615 average precision. Compared with the other top teams, we have probably fitted our GBM model *just enough* on the training data.

6. Conclusion

On the FDMA website, the organizers hoped that this competition would help answer some important questions, which they have listed, about their click fraud problem. Our team has some quick thoughts on these questions:

What is the underlying click fraud scheme? Simply put, a relatively large number of clicks or rapid duplicate clicks, or a high percentage of clicks from high-risk countries have been shown in this paper to be important fraud indicators.

What sort of concealment strategies commonly used by fraudulent parties? The Tuzhilin Report (Tuzhilin, 2006) on the Google AdWords/AdSense system lists ten possible strategies or sources of invalid clicks. The hard-to-detect click fraud tends to come from organized crime in hard-to-prosecute countries (Chambers, 2012). For

example, hard-to-detect and hard-to-prosecute click fraud uses existing user traffic include 0-size iframes, forced searching, and zombie computers (Wikipedia, 2012).

How to interpret data for patterns of dishonest publishers and websites? From a machine learning point-of-view, some decision tree and rule induction algorithms can provide high interpretability to fraud patterns. However, the key step prior to this is still to engineer the best features using domain knowledge and experimentation, and to allow investigators to discern and validate these fraud patterns from top ranked features, even through black-box classification algorithms.

How to build effective fraud prevention/detection plans? Effective fraud detection plans need to have elements of resilience, adaptivity, and quality data (Phua et al., 2012). Resilience is "defense-in-depth" with multiple, sequential, and independent layers of defense. For example, BuzzCity already has rule-based and anomaly-based detectors to place some publishers under observation, and it can consider the addition of classifier-based detectors to their click fraud detection system. In the context of click fraud faced by BuzzCity, the classifier-based detectors need to be adaptive to changing fraud and normal click behavior. The classifier-based detectors also need to use quality data with timely updates when publishers are discovered to be fraudulent. Other than increasing advertisers' awareness of click/conversion ratios, having better customer service and fraud policies, and improving automated filters, BuzzCity can also pursue click fraud more aggressively, switch from a cost per click to cost per action online advertising model, or cultivate trust with advertisers by having independent audits (Jansen, 2007).

Acknowledgments

Our team will like to thank:

1. SMU and BuzzCity for their excellent organization of the FDMA competition, especially with the prompt clarification of questions.
2. Worthy competitors who spurred us to work really hard a few days before the end, particularly team `Zhou`.
3. I^2R for continuous support in our present and future competitions.

References

L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

C. Chambers. Is click fraud a ticking time bomb under google? *Forbes*, 2012. URL `http://www.forbes.com/sites/investor/2012/06/18/is-click-fraud-a-ticking-time-bomb-under-google/`.

A. Chao and T. Shen. Nonparametric estimation of shannons index of diversity when there are unseen species in sample. *Environ. Ecol. Stat.*, 10:429–443, 2003.

W. Cohen. Fast effective rule induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

U. Fayyad and K. Irani. On the handling of continuous-valued attributes in decision tree generation. *Mach. Learn.*, 8(1):87–102, January 1992.

J. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

I. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

D. Horvitz and D. Thompson. A generalization of sampling without replacement from a finite universe. *J. Am. Stat. Assoc.*, 47:663–685, 1952.

B. Jansen. Click fraud. *Computer*, 40:85–86, 2007.

Y. Koren. The bellkor solution to the netflix grand prize, 2009.

Kroll. Kroll advisory solutions global fraud report, 2012. URL `http://www.krolladvisory.com/insights-reports/global-fraud-reports/`. [Online; accessed 28-October-2012].

C. Phua, D. Alahakoon, and V. Lee. Minority report in fraud detection: classification of skewed data. *SIGKDD Explor. Newsl.*, 6(1):50–59, June 2004.

C. Phua, V. Lee, K. Smith-Miles, and R. Gayler. A comprehensive survey of data mining-based fraud detection research. *CoRR*, abs/1009.6119, 2010.

C. Phua, K. Smith-Miles, V. Lee, and R. Gayler. Resilient identity crime detection. *IEEE Trans. on Knowl. and Data Eng.*, 24(3):533–546, March 2012.

G. Ridgeway. Generalized boosted models: A guide to the gbm package. *Update*, 1:1, 2007.

A. Tuzhilin. The lanes gifts v. google report, 2006.

Wikipedia. Click fraud — Wikipedia, the free encyclopedia, 2012. URL `http://en.wikipedia.org/wiki/Click_fraud`. [Online; accessed 28-October-2012].

D. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

H. Yu, H. Lo, H. Hsieh, J. Lou, T. McKenzie, J. Chou, P. Chung, C. Ho, C. Chang, Y. Wei, et al. Feature engineering and classifier ensemble for kdd cup 2010. *Proceedings of the KDD Cup 2010 Workshop*, pages 1–16, 2010.

M. Zhu. Recall, precision and average precision. Technical Report Working Paper 2004-09, University of Waterloo, September 2004.

Y. Zhu, E. Zhong, Z. Lu, and Q. Yang. Feature engineering for place category classification. In *Nokia Mobile Data Challenge Workshop, in conjunction with Pervasive'12, Newcastle*, 2012.