LETTER

Fraud Detection in Comparison-Shopping Services: Patterns and Anomalies in User Click Behaviors

Sang-Chul LEE†, Christos FALOUTSOS†, Dong-Kyu CHAE††, Nonmembers, and Sang-Wook KIM†††, Member

SUMMARY     This paper deals with a novel, interesting problem of detecting frauds in comparison-shopping services (CSS). In CSS, there exist frauds who perform excessive clicks on a target item. They aim at making the item look very popular and subsequently ranked high in the search and recommendation results. As a result, frauds may distort the quality of recommendations and searches. We propose an approach of detecting such frauds by analyzing click behaviors of users in CSS. We evaluate the effectiveness of the proposed approach on a real-world clickstream dataset.

key words: fraud detection, comparison-shopping services, user behavior analysis

1. Introduction

Recently, a comparison-shopping service (CSS), such as Shopping.com, PriceGrabber.com, and Naver shopping (www.shopping.naver.com), has attracted a great deal of attention from online shoppers. Given a query, CSS provides a comprehensive comparison of items in terms of their prices and features [1]. This makes shoppers conveniently compare the items and decide what to buy among them just by clicking one in CSS, without visiting a number of e-commerce sites scattered over the Internet. Due to its convenience, more and more shoppers are making their purchase decisions by using CSS these days [1].

Generally, sellers are interested in only those items ranked high in their search and recommendation results. Figure 1 shows the screenshot of Shopping.com that provides two results for a keyword “laptop”. In the figure, the search and recommendation results are shown in the left and right boxes, respectively. Among a huge number of items relevant to the keyword, the items in the two results would be popular ones with its users. Unless a shopper knows exactly what she wants to purchase, she may click some of those popular items and choose one of them to buy. Indeed, as an item has a higher rank in the search and recommendation results, it is more likely to get a number of clicks leading to purchases. Therefore, sellers try to explore various ways such as advertising campaigns to promote their items to become popular, thereby being ranked high in the both results [2].

However, in CSS, the popularity of an item can be manipulated due to its characteristic of the ranking mechanism: inherently, CSS is unable to know how many times an item has been purchased since it just redirects its users (i.e., shoppers) to individual e-commerce sites by providing a link to the page built for buying the item maintained in those sites. Thus, it is unaware of being purchased, which happen in e-commerce sites rather than CSS. This makes CSS evaluate the popularity of an item only relying on the number of clicks on the item. Subsequently, this motivates fraudulent sellers to click their items excessively in CSS to manipulate the rankings of their items in search or recommendation results, rather than relying on traditional marketing solutions [2]. Such fraudulent actions may result in a significantly distorted quality of search and recommendation services in CSS. Therefore, it is crucial for CSS providers to identify such frauds from a huge number of users.

This paper addresses the issue of detecting frauds who perform excessive clicks on their target items. Along this line, we identify several challenges as follows: (1) it is not feasible for us to classify users in question as frauds or normal shoppers manually due to the huge number of users engaged in CSS and the massive size of click logs; (2) the data of frauds in CSS is not available yet to the public; (3) the frauds in CSS behave in a different way from the frauds in different domains, such as click frauds in advertisement networks [3], social frauds in social network services [5], [6], and ranking frauds in a mobile App store [2], which could make existing methods of fraud detection used in other domains unsuccessful in the CSS domain.

To address the above challenges, this paper offers the following contributions: (1) we analyze the real-world data generated by CSS users and discover their activity patterns
that help distinguish click behaviors of normal users from those of fraudulent users; (2) we propose three anomaly scores to compute how much far the click pattern of a user is from the normal ones, and then propose two measures combining the three scores to find frauds in CSS; (3) through extensive experiments, we show the effectiveness of our approach by identifying frauds in the real-world CSS as well as evaluating accuracy in finding synthetically injected frauds.

2. Overview

Our research was performed with a click log dataset obtained from Naver shopping, one of the biggest CSS in Korea. For a period of eight months, the dataset has been collected via tracing of click-through events generated by 10K sampled users. A click event is characterized by <userID, itemID, timestamp>, indicating userID clicked itemID at the timestamp. Finally our dataset consists of 10,000 users, 301,840 items, and 422,610 clicks.

Intuitively, normal users and frauds act very differently during their usage of CSS: while normal users may perform various actions such as comparing, thinking, and retrieving, frauds may mainly focus on clicking specific items repeatedly. Basically, this difference manifests as distinctive click patterns in our clickstream dataset. According to this intuition, we first analyze the dataset with respect to (1) inter-arrival time (IAT), (2) diurnal activity (DA), and (3) eigen-score (ES), each of which is expected to give hints to understand behavioral differences between frauds and normal users. Through each analysis, we discover activity patterns appearing in normal users. Based on the discovery, for each user u, we compute three anomaly scores of IAT difference (IuAT), DA difference (IuDA), and ES difference (IuES) to quantify the degree of anomalous behavior for her compared to normal users, in the range of 0 and 1: the larger the score, the farther her pattern is from the normal ones. After computing the three anomaly scores, we combine them into the final suspiciousness score, which can be interpreted as a likelihood of a user being a fraud. We finally find the top-k users having the highest suspiciousness scores as frauds.

3. Anomaly Scores

3.1 Inter-Arrival Time Difference

Inter-arrival time (IAT) indicates the time interval between a pair of successive clicks conducted by an individual user. Its distribution is one of important features to distinguish normal users and frauds [5]. Figure 2 (a) shows the IAT distribution of all the users in our dataset. The x-axis indicates IAT in seconds and the y-axis does the ratio of pairs of successive clicks having the corresponding IAT.

We observe that the majority of IATs for users range from 1 to 100 seconds. This distribution seems to appear due to various actions of normal users in shopping such as finding items, examining items, and comparing their prices. Indeed, we confirmed that IATs in CSS follow the log-logistic distribution, well modeled by odds ratio power law, which also coincides with existing theories found in human communication dynamics such as e-mail and SMS communications [7].

In contrast, we report IAT of a suspicious user in our dataset in Fig. 2 (b). Her IAT distribution looks much different from that of other normal users. In particular, almost 80% of successive clicks have IATs of 9 to 14 seconds. This is because, as pointed out in Sect. 2, this suspicious user heavily focused on clicking on a specific item, thereby showing a pattern very different from those of normal users.

Now, we introduce how to compute the first anomaly score, IuAT. For a user u, we count the frequencies of successive click pairs according to their IAT, and derive u’s IAT distribution, denoted as Iu. We also derive distribution of all users’ IAT as Inormal. Here, we set the upper bound of IAT as 1,200, which indicates each user’s session (i.e., the sequence of her clicks during a single visit) is regarded to end if she does nothing for 20 minutes, following [5]. We employ the Kullback-Leibler divergence as a distance function between the two distributions Iu and Inormal:

$$D_{KL}(I_u||I_{normal}) = \sum_i I_u(i) \log \frac{I_u(i)}{I_{normal}(i)}$$

where $D_{KL}(I_u||I_{normal})$ indicates Kullback-Leibler divergence of $I_u$ from $I_{normal}$ and $i$ does the index of elements for both vectors. Since $D_{KL}$ is an asymmetric function, we take the average of $D_{KL}(I_u||I_{normal})$ and $D_{KL}(I_{normal}||I_u)$ to compute $a^{IAT}_u$. Finally, we normalize $a^{IAT}_u$ of every user in the range of 0 and 1 by min-max normalization.

3.2 Diurnal Activity Difference

Diurnal activity (DA) is known useful to understand behavioral differences between frauds and normal users [6]. We report the hourly distribution of all the users’ clicks in Fig. 3 (a), where the x-axis indicates the time (in hour) in a day and the y-axis does the ratio of clicks made in the corresponding time. The graph shows that users clearly exhibit the typical diurnal pattern: there are few behaviors during the sleeping time, 3am to 9am; behaviors after work, 9pm to 12pm, are the most active; there are temporary declines during lunch and dinner times. In contrast, Fig. 3 (b) shows the DA graph of the same suspicious user introduced in Sect. 3.1. During the eight-month tracing, all
the clicks of hers were performed only during 7 to 10 PM.

The overall process of computing the anomaly score, $a_{DA}$, is quite similar to computing $a_{uAT}$. For each user, we discretized the timestamp of her clicks in unit of hour, and derive her DA distribution $D_u$, which corresponds to how many times does $u$ clicks on the corresponding time period (in hours). In the same way, we define a distribution of DA for all users, denoted as $D_{normal}$. We then compute $a_{DA}^u$ by taking the average of $D_{KL}(D_u||D_{normal})$ and $D_{KL}(D_{normal}||D_u)$, and finally do min-max normalization on $a_{DA}^u$ in the range of 0 and 1.

### 3.3 Eigenscore Difference

Consider a matrix composed of blocks, each of which corresponds to a user-item pair and has the numbers of daily clicks for that pair, as shown in Fig. 4. It is also a good criterion of anomalousness to examine whether a user produces high density in a block [8]. Here, the density in a block indicates not only the ratio of non-zero numbers but also the magnitude of numbers in the block. Intuitively, such dense blocks are made by such users who perform excessive clicks on a specific item in a very short time. Along this line, we represent our dataset as an $m \times n$ matrix $A$ (m is # of users and $n$ is # of itemsx# of days), as in Fig. 4. Then, we find dense blocks in the matrix and measure how much relevant each user is to the found dense blocks.

Towards this goal, we employ the singular value decomposition (SVD), one of the most widely used methods to discover such dense blocks [8]. SVD of an $m \times n$ matrix $A$ is a factorization of the form $A = U\Sigma V^T$: $U$ and $V$ are $m \times m$ and $n \times n$ matrices whose columns are called the left-singular vectors and right-singular vectors of $A$, respectively; $\Sigma$ is an $m \times n$ diagonal matrix comprised of its singular values. Here, the top singular values and singular vectors indicate dense blocks in the matrix [8]. Also, for each top left-singular vector, the $m$ absolute values, which are referred to as eigenscores, indicate the degree of relevance of each user to the corresponding dense block: the higher the value, the more engaged in the corresponding dense block [8].

We conducted SVD on our dataset and derived top-$n$ singular values (i.e., dense blocks) and left-singular vectors. For preliminary observation, we set $n$ as 50 (i.e., looking for 50 dense blocks). For each derived left-singular vector, we normalized the eigenscores in the range of 0 and 1. Note that every user has $n$ eigenscores. Among them, we chose the highest eigenscore for each user as her representative eigenscore. The result shows that, while most of users have low representative eigenscores, a small number of users have abnormally high representative eigenscores. These users having the extremely high scores are likely to be frauds.

In order to compute the anomaly score, $a_{uES}$, we compute the distance between the average of all users’ representative eigenscores ($RE_{average}$) and user $u$’s representative eigenscore ($RE_u$), as shown in the following:

$$a_{uES} = |RE_u - RE_{average}|$$ (2)

### 4. Combining Anomaly Scores

This section discusses how to combine the three anomaly scores to derive the final suspiciousness score of users. We employ the extended $p$-norm model [4], which has been widely used in combining the similarity scores between each keyword and a document. It has two variations, $\Phi_{OR}(u)$ and $\Phi_{AND}(u)$, each of which is formulated as follows:

$$\Phi_{OR}(u) = \frac{w_1(a_{uAT}^p)^p + w_2(a_{DA}^p)^p + w_3(a_{uES}^p)^p}{w_1 + w_2 + w_3}$$ (3)

$$\Phi_{AND}(u) = \frac{w_1(1 - a_{uAT}^p)^p + w_2(1 - a_{DA}^p)^p + w_3(1 - a_{uES}^p)^p}{w_1 + w_2 + w_3}$$ (4)

We assign the same value to the three weights $w_1, w_2,$ and $w_3$ because we regard the three anomaly scores equally important. Also, we fix the value of $p$ as 5, which is the most popular one. Also, we observed that both $\Phi_{OR}(u)$ and $\Phi_{AND}(u)$ are not sensitive to $p$ values through our preliminary experiments. Therefore, we fix the value of $p$ as 5, which is the most popular one. With this parameter setting, we use both combined measures to detect frauds in CSS.

### 5. Case Study

For each combined measure, we computed suspiciousness scores of 9,997 users in Naver Shopping. As a result, we identified 7 users who belong to the intersection of two sets of top-10 rankers (i.e., most suspicious users) in terms of $\Phi_{OR}$ and $\Phi_{AND}$. We found some suspicious behaviors by inspecting their click logs manually. The suspiciousness in their behaviors is described in Table 1.

We observe that all the seven users have their own target item(s) aimed for excessive clicks. For example, userID 9587 clicked only one item 244 times, which occupies 93.1% of her entire clicks. Also, userID 8068 clicked...
three items 742 times just in 3 hours and never came back to the site again. Her IAT and DA graphs are shown in Fig. 2 (b) and 3 (b), respectively. We omit those graphs of the other suspicious users due to space limitations because they show tendencies similar to those of userID 8068.

6. Evaluation

This section presents the experimental results of evaluating the accuracy of our two measures. Since there is no ground truth data of frauds in CSS, we generated synthetic frauds and injected them into our clickstream dataset. To the end, we made each fraud select X target items and click Y times for each target item. The value of X was chosen randomly from [2, 4] for each fraud, and the value of Y is chosen randomly from [150, 250] for each fraud-item pair. The click generation started from a random day and continued until a pre-defined number of clicks were all generated.

We considered three types of frauds, bot, burst, and low temperature, all of which may possibly exist in the real-world: (1) bot, the simplest one, simulates not a human being but an automated program that clicks target items at regular time intervals; (2) burst simulates a fraudulent human user who clicks target items at short and random time intervals in a session, having several sessions per day; (3) low temperature simulates a fraudulent user who is a little bit smarter than burst and aims to avoid detection by clicking target items at relatively long and random time intervals, having only one session per day. We summarize the characteristics of three types of frauds in Table 2.

We created 25 frauds for each fraud type and injected them into our dataset. Next, we measured every user’s suspiciousness score and ranked them in the descending order of their score. Then, we evaluated the ranking by measuring its average precision. To avoid the bias from randomness, we carried out 1,000 experiments with different sets of generated frauds and computed the mean value of 1,000 average precisions, i.e., mean average precision (MAP). We compared the MAP of our proposed \(\Phi_u\) and \(\Phi_{and}\) with that of four baselines. The three of baselines are the solely-used anomaly scores, \(a^{AT}_u\), \(a^{DA}_u\), and \(a^{ES}_u\), and the other one is an intuitive anomaly score, denoted as \(a^{clicks}_u\), which simply combines (1) the average number of clicks per item and (2) the average number of clicks per day.

The results are shown in Fig. 5. \(a^{clicks}_u\) is shown to provide low MAP, which demonstrates that it is insufficient to determine whether a user is a fraud by considering just the number of clicks per item or per day. This is because there exist hard shoppers, who repeatedly click several items for comparison purpose but not with the fraudulent purpose. \(a^{AT}_u\) captures bots very well, but misses several bursts and most low temperatures. This is because the IAT of low temperature resembles that of normal users. \(a^{ES}_u\) captures more than 87.5% of frauds for all the fraud types, missing just a small portion of frauds. \(a^{DA}_u\) is shown to provide very poor MAP when used alone. However, we observe that \(a^{DA}_u\) identifies several frauds that \(a^{AT}_u\) and \(a^{ES}_u\) miss, which shows that \(a^{DA}_u\) could be still useful when combined with the other anomaly scores to improve overall MAP.

The proposed \(\Phi_{or}\) and \(\Phi_{and}\) both provide the highest MAP, while \(\Phi_{and}\) shows a MAP slightly higher than \(\Phi_{or}\). Indeed, our proposed measures consider a user’s various aspects of click behaviors with respect to IAT, DA, and ES from the three anomaly scores. This makes our proposed measures outperform the baseline measures considerably.

7. Conclusions

This paper addressed a method of detecting frauds in CSS. We proposed the three kinds of anomaly scores and two ways of combining them to compute the degree of suspiciousness of each user. Our experimental results demonstrate that our propose approach successfully uncovers suspicious users in the real-world CSS and that it accurately identifies synthetically injected frauds in comparison with baselines.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government.
(MSIP; Ministry of Science, ICT & Future Planning) (No. NRF-2017R1A2B3004581).

References


