International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

A Review on Ranking Based Fraud Detection in Android Market

D. Janet[1], Vikrant Chole[2]

[1]Department of Computer Science and Engineering, GHRAET, Nagpur

[2]Professor, Department of Computer Science and Engineering, GHRAET, Nagpur

Abstract: ***Ranking fraud in the mobile App market refers to fraudulent or deceptive activities which have a purpose of bumping up the Apps in the popularity list. Indeed, it becomes more and more frequent for App developers to use shady means, such as inflating their Apps' sales or posting phony App ratings, to commit ranking fraud. While the importance of preventing ranking fraud has been widely recognized, there is limited understanding and research in this area. In this Paper we are reviewing various a holistic view of ranking fraud and propose a ranking fraud detection system for mobile Apps. Specifically, we first study various ranking fraud. Furthermore, we investigate different methodologies and characterised in into three types of evidences in fraud detection, i.e., ranking based evidences, rating based evidences and review based evidences, by modelling Apps' ranking, rating and review behaviours through statistical hypotheses tests. In addition, we will also propose an optimization based aggregation method to integrate all the evidences for fraud detection.***

Keywords: Mobile Apps, ranking fraud detection, evidence aggregation, historical ranking records, rating and review

1. Introduction

The number of mobile Apps has grown at a breathtaking rate over the past few years. For example, as of the end of April 2013, there are more than 1.6 million Apps at Apple App store and Google Play. To stimulate the development of mobile Apps, many App stores launched daily App leader boards, which demonstrate the chart rankings of most popular Apps. Indeed, the App leader board is one of the most important ways for promoting mobile Apps. A higher rank on the leader board usually leads to a huge number of downloads and million dollars in revenue. Therefore, App developers tend to explore various ways such as advertising campaigns to promote their Apps in order to have their Apps ranked as high as possible in such App leader boards. However, as a recent trend, instead of relying on traditional marketing solutions, shady App developers resort to some fraudulent means to deliberately boost their

Apps and eventually manipulate the chart rankings on an App store. This is usually implemented by using so-called farms man water armies to inflate the App downloads, ratings and reviews in a very short time. For example, an article from Venture Beat reported that, when an App was promoted with the help of ranking manipulation, it could be propelled from number 1,800 to the top 25 in Apple top free leader board and more than 50,000-100,000 new users could be acquired within a couple of days. In fact, such ranking fraud raises great concerns to the mobile App industry. For example, Apple has warned of cracking down on App developers who commit ranking fraud in the Apple App store. In the literature, while there are some related work, such as web ranking spam detection, online review spam detection, and mobile App recommendation, the problem of detecting ranking fraud for mobile Apps is still under-explored. To fill this crucial void, in this paper, we propose to develop a ranking fraud detection systemfor mobile Apps. Along this line, we identify several important challenges. First, ranking fraud does not always happen in the whole life cycle of an App, so we need to detect the time when fraud happens. Such challenge can be regarded as detecting the local anomaly instead of global anomaly of mobile Apps. Second, due to the huge number of mobile Apps, it is difficult to manually label ranking fraud for each App, so it is important to have a scalable way to automatically detect ranking fraud without using any benchmark information. Finally, due to the dynamic nature of chart rankings, it is not easy to identify and confirm the evidences linked to ranking fraud, which motivates us to discover some implicit fraud patterns of mobile Apps as evidences. Then, with the analysis of Apps' ranking behaviours, we find that the fraudulent Apps often have different ranking patterns in each leading session compared with normal Apps. Thus, we characterize some fraud evidences from Apps' historical ranking records, and develop three functions to extract such ranking based fraud evidences. Nonetheless, the ranking based evidences can be affected by App developers' reputation and some legitimate marketing campaigns, such as "limited-time discount". As a result, it is not sufficient to only use ranking based evidences. Therefore, we further propose two types of fraud evidences based on Apps' rating and review history, which reflect some anomaly patterns from Apps' historical rating and review records. In addition, we develop an unsupervised evidence-aggregation method to integrate these three types of evidences for evaluating the credibility of leading sessions from mobile Apps. It is worth noting that all the evidences are extracted by modelling Apps' ranking, rating and review behaviours through statistical hypotheses tests. The proposed framework is scalable and can be extended with other domain generated evidences for ranking fraud detection. Finally, we evaluate the proposed system with real-world App data collected from the Apple's App store for a long time period, i.e., more than two years.

2. Background

Web Advertising

Advertising on the Internet is pervasive, and allows for services such as websites, search, and email to be provided to customers for free by including advertisements (ads) as part of the content displayed to the user. Website owners and other service providers (called publishers in advertising jargon) typically include ads through a third party called an ad provider, which handles nding and selecting advertisements, as well as paying publishers for ads shown to their users. On the Web, this is typically implemented as an <iframe> or <script> HTML element embedded in the publisher's webpage, with a src attribute that points to the ad provider's ad server. When the web page is loaded y a browser, the ad is populated via an ad request, which contains the publisher's ID and information about the user that is used to select a relevant ad (known as targeting information). The ad server returns three pieces of content once an ad is selected: the ad content URL, a click URL, and a pixel URL.

The ad content is typically hosted by the ad provider (usually through a CDN) instead of the digital marketer who owns the ad, ensuring the content will be vailable when the ad is loaded. Marketers who are paying for their ads to be distributed by the ad provider want to guarantee the ad provider is not fraudulently billing them, so they themselves host a tracking pixel (or web bug) that is aded by browsers along with the ad so that the marketers can independently verify that ads are being requested.

Finally, the click URL indicates which web page should be opened when a user clicks on an ad. The click URL typically points to the ad provider's ad server, which records the icks and then redirects the user to the marketer's landing page. A complete ad request, response, and display of the ad and pixel to the user is called an impression, and opening the click URL is a click. Publishers are paid based on how many impressions and clicks their content generates.

Web Ad Fraud

Unscrupulous publishers may inate their ad revenues by having automated bots visit their website and click on ads.This is referred to as ad fraud (or click fraud), and is a serious security issue as digital marketers who pay to have their ads shown online will not receive any business benet for ads shown to bots. Although hard numbers on the amount of ad fraud is hard to determine, conservative estimates suggest 10% of Web ad traffic is due to fraud [11]. In order to receive revenue, fraudsters must remain undetected while issuing large numbers of ad requests and clicks. To do so, they employ a number of techniques.

First, the ratio of click requests to requested ads is kept low (around 1% [28]) to avoid suspicion, as ads are rarely clicked on by real users. This means fraudsters issue far more ad requests than click requests. Second, fraudsters do not rely on a single publisher account, but rather have many accounts from many ad providers which they rotate through while issuing requests [28]. Not only does this mitigate the impact of any single account being detected, it also decreases the magnitude of fraudulent requests for each publisher ID and ad provider.

Finally, fraudsters use botnets as the bots run code that consistently visits the fraudsters' webpages in the background and clicks on the ads located there, so that the fraudsters receive revenue. Botnets allow fraudsters to remain stealthy as the bots are real user devices which have been compromised.

Android App Advertising

Many Android applications are distributed for free on app markets, and use ads embedded in the user interface of the app to make money for the developer. The developer must register with an Android ad provider, which provides the developer with a publisher ID and an ad library to include in their app. The library is responsible for fetching and displaying ads when the app is being run. Requesting an ad for an app is analogous to doing so on the web: an ad request is made over HTTP to the ad server which includes the developer's publisher ID and user targeting information.

The ad server returns the ad's content URL, click URL, and any tracking pixel URLs which must be fetched to display the ad. In fact, many ad libraries choose to implement making requests and displaying ads simply by loading a traditional HTML ad element in a web view. The primary difference between web and Android app advertising is that ad libraries are implemented in application code, and often contain special application-only logic, for example automatically collecting user targeting information or refreshing the ad.

3. Literature Survey

Xiong and Zhu proposed a ranking fraud detection system for android mobile apps[1]. In this paper particularly, authors showed that ranking fraud happened in primary sessions for each app from its past ranking records. Then, they identified ranking based, rating based and review based evidences for finding ranking fraud. Additionally, authors proposed an optimization based aggregation method to combine all the evidences for evaluating the reliability of leading sessions from mobile apps. Priyanjai and pankaj proposed methods for evaluation of analysis and design pattern of android apps based on cloud computing and data mining. The Authors developed mechanism ASEF and SAAF for android apps to achieve security. In this authors describe a methodology that performs apps security and provide user friendly interface on a mobilephone. [3] provides a methodical study on the different techniques of malicious application detection in android mobiles.

The investigation of permission-induced risk in Android apps on a large-scale in three levels. First upon rank all the individual permissions with respect to their probable risk with different methods. Secondly , categorize subsets of risk permissions. Then using several algorithms detect the malapps based on the identified subsets of risky permissions. Search engine optimization techniques[4] , often shortened to SEO, should lead to first positions in organic search results. Some optimization techniques do not change over time, yet still form the basis of SEO.

However, as the Internet and web design evolves dynamically, new optimization techniques arise and die. Thus, [4] look at the most important factors that can help to improve a position in search results. It is important to emphasize, that none of the techniques can guarantee it because search engines have sophisticated algorithms, which measure the quality of Web pages and derive their position in search results from. Users can annotate themselves using free tags in microblogging website such as Sina Weibo. The tags of a user demonstrate [5].

The characteristics of the user and are generally in a random order without any importance or relevance information.It limits the effectiveness of user tags in system recommendation and other applications. Xiang [5] proposed a user tag ranking schema which is based on interactive relations between users. Influence strength between users is considered in our user tag ranking method. Relevance scores between tags and users are also utilized to rank user tags.App store and android market have experienced a significant growth in terms of appnumbers[7]. Since we discover 85% of apps through the ranks, it is important to develop effective appranking analysing tools. Woong presented a method called App Analytic. In this He explored the correlations of app ranking data about popular social networking sites. Specifically, they analyzed correlations between various characteristics of social networking sites on Internet and android market. The results of their correlation analysis reveal that there is a strong positive correlation of the number of appdownloads with the number of registered users and pagerank. They also provide an in-depth analysis on the major factors that impact the correlations.

4. Research Methodology

1. Mining Leading Sessions

There are two main steps for mining leading sessions. First, we need to discover leading events from the App's historical ranking records. Second, we need to merge adjacent leading events for constructing leading sessions. Specifically, Algorithm 1 demonstrates the pseudo code of mining leading sessions for a given App.

2. Extracting Evidences For Ranking Fraud Detection

- **Ranking Based Evidences**
  By analyzing the Apps' historical ranking records, Apps' ranking behaviours in a leading event always satisfy a specific ranking pattern, which consists of three different ranking phases, rising phase, maintaining phase and recession phase. Specifically, in each leading event, an App's ranking first increases to a peak position in the leaderboard (i.e., rising phase), then keeps such peak position for a period (i.e., maintaining phase), and finally decreases till the end of the event (i.e., recession phase). Fig. 3 shows an example of different ranking phases of a leading event. Indeed, such a ranking pattern shows an important understanding of leading event. In the following, we formally define the three ranking phases of a leading event.
- **Rating Based Evidences**
  The ranking based evidences are useful for ranking fraud detection. However, sometimes, it is not sufficient to only use ranking based evidences. For example, some Apps created by the famous developers, such as Gameloft, may have some leading events with large values of u due to the developers'credibility and the "word-of-mouth" advertising effect. Moreover, some of the legal marketing services, such as "limited-time discount", may also result in significant ranking based evidences. To solve this issue, we also study how to extract fraud evidences from Apps' historical rating records
- **Review Based Evidences**
  Besides ratings, most of the App stores also allow users to write some textual comments as App reviews. Such reviews can reflect the personal perceptions and usage experiences of existing users for particular mobile Apps. Indeed, review manipulation is one of the most important perspective of App ranking fraud. Specifically, before downloading or purchasing a new mobile App, users often first read its historical reviews to ease their decision making, and a mobile App contains more positive reviews may attract more users to download. Therefore, imposters often post fake reviews in the leading sessions of a specific App in order to inflate the App downloads, and thus propel the App's ranking position in the leaderboard
- **Evidence Aggregation**
  After extracting three types of fraud evidences, the next challenge is how to combine them for ranking fraud detection. Indeed, there are many ranking and evidence aggregation methods in the literature, such as permutation based models, score based models , and Dempster-Shafer rules . However, some of these methods focus on learning a global ranking for all candidates.

5. Conclusion

A ranking fraud detection system for mobile Apps show that ranking fraud happened in leading sessions and provide a method for mining leading sessions for each App from its historical ranking records. Then, our study identifies that it can be broadly characterised into three category i.e. ranking based evidences, rating based evidences and review based evidences for detecting ranking fraud and an optimization based aggregation method to integrate all the evidences for evaluating the credibility of leading sessions from mobile Apps.

References

[1] Hengshu Zhu, Hui Xiong, Senior Member, IEEE, Yong Ge, and Enhong Chen, Senior Member, IEEE Discovery of Ranking Fraud for Mobile Apps‖ IEEE Transactions On Knowledge And Data Engineering, Vol. 27, No. 1, January 2015.
[2] Pranjali Deshmukh, Pankaj Agarkar ―Mobile Application For Malware Detection‖ International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 02 Issue: 02 | May-2015 www.irjet.net
[3] Anuja A. Kadam ,Pushpanjali M. Chouragade ―A Review Paper on: Malicious Application Detection in Android System‖International Journal of Computer Applications (0975 – 8887) National Conference on Recent Trends in Computer Science & Engineering (MEDHA 2015).

[4] Jakub Zilincan ,Michal Gregus "Improving Rank of a Website in Search Resuts – a Experimental Approach"2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing978-1-4673-9473-4 /15 $31.00 © 2015 IEEE
[5] Xiang Wang,Yan Jia , Ruhua Chen, Bin Zhou , "Ranking User Tags in Micro-blogging Website",978-1-4673-6850-6/15 .2015 IEEE
[6] App Analytic: A Study on Correlation Analysis of App Ranking Data Sun-Young Ihm; Woong-Kee Loh; Young-Ho Park Cloud and Green Computing (CGC), 2013 Third International Conference on Year: 2013 Pages: 561 ‐ 563, DOI: 10.1109/CGC.2013.95 IEEE Conference Publications
[7] Jakub Zilincan ,Michal Gregus "Improving Rank of a Website in Search Resuts – a Experimental Approach"2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing978-1-4673-9473-4 /15 $31.00 © 2015 IEEE
[8] L. Azzopardi, M. Girolami, and K. V. Risjbergen, "Investigating the relationship between language model perplexity and ir precision-recall measures," in Proc. 26th Int. Conf. Res. Develop. Inform. Retrieval, 2003, pp. 369–370.
[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," J. Mach. Learn. Res., pp. 993–1022, 2003.
[10] Y. Ge, H. Xiong, C. Liu, and Z.-H. Zhou, "A taxi driving fraud detection system," in Proc. IEEE 11th Int. Conf. Data Mining, 2011, pp. 181–190.
[11] D. F. Gleich and L.-h. Lim, "Rank aggregation via nuclear norm minimization," in Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2011, pp. 60–68.
[12] T. L. Griffiths and M. Steyvers, "Finding scientific topics," Proc. Nat. Acad. Sci. USA, vol. 101, pp. 5228–5235, 2004.
[13] G. Heinrich, Parameter estimation for text analysis, " Univ. Leipzig, Leipzig, Germany, Tech. Rep., http://faculty.cs.byu.edu/~ringger/CS601R/papers/Heinrich-GibbsLDA.pdf, 2008.
[14] N. Jindal and B. Liu, "Opinion spam and analysis," in Proc. Int. Conf. Web Search Data Mining, 2008, pp. 219–230.
[15] J. Kivinen and M. K. Warmuth, "Additive versus exponentiated gradient updates for linear prediction," in Proc. 27th Annu. ACMSymp. Theory Comput., 2005, pp. 209–218.
[16] A. Klementiev, D. Roth, and K. Small, "An unsupervised learning algorithm for rank aggregation," in Proc. 18th Eur. Conf. Mach. Learn., 2007, pp. 616–623
[17] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in Proc. 19th ACMInt. Conf. Inform. Knowl. Manage., 2010, pp. 939–948.
[18] Y.-T. Liu, T.-Y. Liu, T. Qin, Z.-M. Ma, and H. Li, "Supervised rank aggregation," in Proc. 16th Int. Conf. World Wide Web, 2007, pp. 481–490.
[19] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," in Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2013, pp. 632–640.
[20] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam web pages through content analysis," in Proc. 15th Int. Conf. World Wide Web, 2006, pp. 83–92.
[21] G. Shafer, A Mathematical Theory of Evidence. Princeton, NJ, USA: Princeton Univ. Press, 1976.
[22] K. Shi and K. Ali, "Getjar mobile application recommendations with very sparse datasets," in Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2012, pp. 204–212.
[23] N. Spirin and J. Han, "Survey on web spam detection: Principles and algorithms," SIGKDD Explor. Newslett., vol. 13, no. 2, pp. 50–64, May 2012.
[24] M. N. Volkovs and R. S. Zemel, "A flexible generative model for preference aggregation," in Proc. 21st Int. Conf. World Wide Web, 2012, pp. 479–488.