ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 3, March 2016

Exploration of Sorting Swindle For Mobile Apps

KV. Deepa Lakshmi[1], T.Nirmalraj[2]
M.Phil (CS) Scholar[1], Assistant Professor[2]
Department of CSA Department of CSA
SCSVMV University, Enathur, Kanchipuram SCSVMV University, Enathur, Kanchipuram

Abstract: Ranking artifice in the adaptable App bazaar refers to counterfeit or ambiguous activities which accept a purpose of bumping up the Apps in the acceptance list. Indeed, it becomes added and added common for App developers to use adumbral means, such as inflating their Apps' sales or announcement affected App ratings, to accomplish baronial fraud. While the accent of preventing baronial artifice has been broadly recognized, there is bound compassionate and analysis in this area. To this end, in this paper, we accommodate a holistic appearance of baronial artifice and adduce a baronial artifice apprehension arrangement for adaptable Apps. Specifically, we aboriginal adduce to accurately locate the baronial artifice by mining the alive periods, namely arch sessions, of adaptable Apps. Such arch sessions can be leveraged for audition the bounded aberration instead of all-around aberration of App rankings. Furthermore, we investigate three types of evidences; In addition, we adduce an enhancement based accession adjustment to accommodate all the evidences for artifice detection. Finally, we appraise the proposed arrangement with real-world App abstracts calm from the ios App Store for a continued time period. In the experiments, we validate the capability of the proposed system, and appearance the scalability of the apprehension algorithm as able-bodied as some regularity of baronial artifice activities.

KeyWords: MobileApps, RankingFraudDetection

I. 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‟s 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 "bot farms" or "human water armies" to inflate the App downloads, ratings and reviews in a very short time. For example, an article from VentureBeat 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‟s top free leaderboard 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‟s App store

2. LITERATURE REVIEW

2.1 INVESTGATION OF RANKING RATING AND REVIEW USING STATISTICAL HYPOTHESES TESTS

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[4] a s 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. To this end in this paper[7], we provide a holistic view of ranking fraud and propose a ranking fraud detection system for mobile Apps. Specifically, we first[8] propose to accurately locate the ranking fraud by mining

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 3, March 2016

the active periods, namely leading sessions, of mobile Apps. Such[8] leading sessions can be leveraged for detecting the local anomaly instead of global anomaly of App rankings.

Furthermore, we investigate three types of evidences, 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[10], we propose an optimization based aggregation method to integrate all the evidences for fraud detection. Finally, we evaluate the proposed system[11] with realworld App data collected from the iOS App Store for a long time period. In the experiments, w e validate the effectiveness of the proposed system, an d show the scalability of the detection algorithm as well as some regularity of ranking fraud activities

2.2 LATENT DIRICHLET ALLOCATION

We describe [12] latent Dirichlet allocation (LDA), a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document. We present efficient approximate inference techniques based on variational methods and an EM algorithm for empirical Bayes parameter estimation. We report[15] results in document modeling, text classification, and collaborative filtering, comparing to a mixture of unigrams model and the probabilistic LSI model.

2.3 HIERARCHY FRAUD DETECTION FOR MOBILE APPS AND EFFICIENT SEARCHING APPS

The primary aim of this project is to enhance the prevention of ranking frauds in mobile apps using the MAC address. In the existing system[17] the leading event and leading session of an app is identified from the collected historical records. Then three different types of evidences are collected from the user feedbacks namely ranking based evidence, rating based evidence and review based evidence. These three evidences are aggregated by using evidence aggregation method. In the proposed system additionally, we are proposing[20] two enhancements. Firstly, we are using Sentiword dictionary to identify the exact reviews scores. Secondly, the fake feedbacks by a same person for pushing up that app on the leader board are restricted. Two different constraints are considered for accepting the feedback given to an application. The first constraint is that an app can be rated only once from a user login. And the second is implemented with the aid of MAC address that limits the number of user login logged per day from a MAC address as five.

2.4 PARAMETER ESTIMATION FOR TEXT ANALYSIS

Presents parameter estimation methods common with discrete probability distributions, which is of particular interest in text modeling. Starting with maximum likelihood, a posteriori and Bayesian estimation, central concepts like conjugate distributions and Bayesian networks are reviewed. As an application[21], the model of latent Dirichlet allocation (LDA) is explained in detail with a full derivation of an approximate inference algorithm based on Gibbs sampling, including a discussion of Dirichlet hyperparameter estimation.

3.ARCHITECTURE DIAGRAM

4. METHODOLOGY JAVA TECHNOLOGY

Java technology is both a programming language and a platform.

JAVA PLATFORM

A platform is the hardware or software environment in which a program runs. We‟ve already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it‟s a software-only platform that runs on top of other hardware-based platforms. The Java platform has two components: The Java Virtual Machine (Java VM) The Java Application Programming Interface (Java API) You‟ve already been introduced to the Java VM. It‟s the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of readymade software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 3, March 2016

database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database.

For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN. The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn‟t change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer. The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn‟t as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of “plug-in” database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC‟s framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution. JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after. The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

CONCLUSION

we developed a ranking fraud detection system for mobile Apps. Specifically, we first showed that ranking fraud happened in leading sessions and provided a method for mining leading sessions for each App from its historical ranking records. Then, we identified ranking based evidences, rating based evidences and review based evidences for detecting ranking fraud. Moreover, we proposed an optimization based aggregation method to integrate all the evidences for evaluating the credibility of leading sessions from mobile Apps. An unique perspective of this approach is that all the evidences can be modeled by statistical hypothesis tests, thus it is easy to be extended with other evidences from domain knowledge to detect ranking fraud. Finally, we validate the proposed system with extensive experi-

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 3, March 2016

ments on real-world App data collected from the Apple‟s App store. Experimental results showed the effectiveness of the proposed approach.

FUTURE WORK

In the future, we plan to study more effective fraud evidences and analyze the latent relationship among rating, review and rankings. Moreover, we will extend our ranking fraud detection approach with other mobile App related services, such as mobile Apps recommendation, for enhancing user experience.

7. REFERENCES

[1]. L. Azzopardi, M. Girolami, and K. V. Risjbergen. Investigating the relationship between language model perplexity and irprecision-recall measures. In Proceedings of the 26th International Conference on Research and Development in Information Retrieval (SIGIR‟03), pages 369–370, 2003.

[2]. D. M. Blei, A. Y. Ng, and M. I. Jordan. Lantent dirichlet allocation. Journal of Machine Learning Research, pages 993–1022, 2003.

[3]. 3)Y. Ge, H. Xiong, C. Liu, and Z.-H. Zhou. A taxi driving fraud detection system. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining, ICDM „11, pages 181–190, 2011.

[4]. 4)D. F. Gleich and L.-h. Lim. Rank aggregation via nuclear norm minimization. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD „11, pages 60–68, 2011.

[5]. 5) T. L. Griffiths and M. Steyvers. Finding scientific topics. In Proc. of National Academy of Science of the USA, pages 5228–5235, 2004.

[6]. 6) G. Heinrich. Paramter stimaion for text analysis. Technical report, University of Lipzig, 2008.

[7]. 7)N. Jindal and B. Liu. Opinion spam and analysis. In Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM „08, pages 219– 230, 2008.

[8]. 8)J. Kivinen and M. K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. In Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, STOC „95, pages 209–218, 1995.

[9]. 9)A. Klementiev, D. Roth, and K. Small. An unsupervised learning algorithm for rank aggregation. In Proceedings of the 18th European conference on Machine [2] Anderson, R.E. and Srinivasan, S.E. (2003) „E-Satisfaction and E-Loyalty: A Contingency Framework‟, Psychology & Marketing, 20(2), pp. 123 - 38.

[10]. 10)A. Klementiev, D. Roth, and K. Small. Unsupervised rank aggregation with distance-based models. In Proceedings of the 25th international conference on Machine learning, ICML „08, pages 472–

[11]. 479, 2008.

[12]. 11)A. Klementiev, D. Roth, K. Small, and I. Titov. Unsupervised rank aggregation with domain-specific expertise. In Proceedings of the 21st international jont conference on Artifical intelligence, IJCAI„09, pages 1101–1106, 2009.

[13]. 12)E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. Detecting product review spammers using rating behaviors. In Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM „10, pages 939–948, 2010.

[14]. 13)N. Spirin and J. Han. Survey on web spam detection: principles and algorithms. SIGKDD Explor. Newsl., 13(2):50–64, May 2012.

[15]. 14)M. N. Volkovs and R. S. Zemel. A flexible generative model for preference aggregation. In Proceedings of the 21st international conference on World Wide Web, WWW „12, pages 479–488, 2012.

[16]. 15)Z. Wu, J. Wu, J. Cao, an d D. Tao. Hysad: a semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD „12, pages 985–993, 2012.

[17]. 16)S. Xie, G. Wang, S. Lin, and P. S. Yu. Review spam detection via temporal pattern discovery. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD „12, pages 823–831, 2012.

[18]. 17)B. Yan and G. Chen. Appjoy: personalized mobile application discovery. In Proceedings of the 9th international conference on Mobile systems, applications, and services, MobiSys „11, pages 113–126, 2011.