IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

INVESTGATION OF RANKING RATING AND REVIEW USING STATISTICAL HYPOTHESES TESTS

Prof.Mariappan.R*, Ms.Deivanai.A, Ms.B.Hema, Ms.Thamizharasi.V
[*] Dept of Information Technology Velammal Institute of Technology Chennai, India
Dept of Information Technology Velammal Institute of Technology Chennai, India
Dept of Information Technology Velammal Institute of Technology Chennai, India
Dept of Information Technology Velammal Institute of Technology Chennai, India

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 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, we provide a holistic view of ranking fraud and propose a ranking fraud detection system for mobile Apps. Specifically, we first propose to accurately locate the ranking fraud by mining the active periods, namely leading sessions, of mobile Apps. Such 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, we propose an optimization based aggregation method to integrate all the evidences for fraud detection. Finally, we evaluate the proposed system with real-world 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.

KEYWORDS: Apps, Ranking Fraud Apprehension, Evidence Reckoning, Historical Records, Rating and Review

INTRODUCTION

Web spam refers to all forms of malicious manipulation of user generated data so as to impudence usage patterns of the data. The number of mobile Apps has grown at a breath Taking rate over the past few years. For example, as of the end of April 2013, there are more than 1.6million Apps at Apple's App store and Google Play .

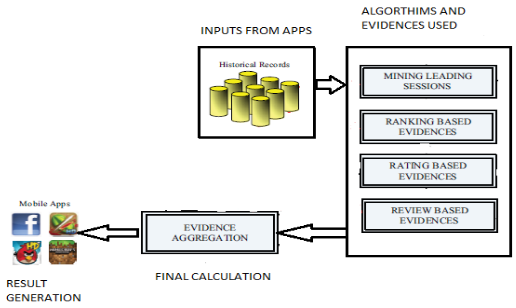
To stimulate the development of mobile Apps, many App stores launched daily App leader boards, which demonstrate the chart rankings of most popular Apps.Indeed, the App leader boar's one of the most important way for promoting mobile Apps. A higher rank on the leader board usually leads to huge number of downloads and million dollars in the 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.

Margins, column widths, line spacing, and type styles are built-in; examples of the type styles are provided throughout this document and are Identified in italic type, hence within parentheses, following the example. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow. Indeed, our careful observation reveals that mobile Apps are not always ranked high in the leader board, but only in some leading events, which form different leading sessions. Note that we will introduce both leading events Ease of Use and leading sessions in detail later. In other words, ranking fraud usually happens in these leading sessions. Therefore, detecting ranking fraud of mobile Apps is actually to detect ranking fraud within leading sessions of mobile

Apps. Specifically, we first propose a simple yet effective algorithm to identify the leading sessions of each App based on its historical ranking records. 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.
Hence Thus, we have characterized 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 f 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. Figure 1 shows the framework of our ranking fraud.

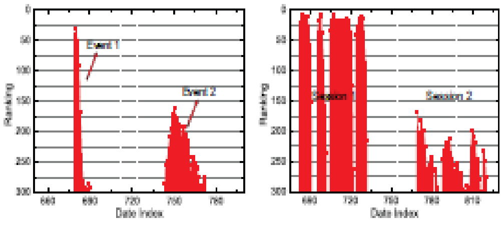
It is worth noting that all the evidences are extracted By modelling Apps' ranking, rating and review behaviours through statistical hypotheses tests. The propose framework is scalable and can be extended with other domain generate d evidences for ranking fraud detection. Finally, we evaluate the proposed system with real-world App data collected from the Apple's App s tore for a long time period, i.e., more than two years. Experimental results show the effectiveness of the proposed system, the scalability of the detection algorithm as well as some regularity of ranking fraud activities.According to the definitions introduced in, a leading session is composed of several leading events. Therefore, w should first analyze the basic characteristics of leading evens for extracting fraud evidences hence .By the analyzing the Apps' historical ranking records, we observe those that Apps' ranking behaviors in a leading even always satisfy a specific ranking pattern, which consists of the three different ranking phase, namely, rising phase, maintaining phase and recession phase.



A leading session is composed of several leading events. By analyzing the Apps' historical ranking records, we observe tha t Apps' ranking behaviors in a leading event always satisfy a specific ranking pattern,which consistsof three different rankin g phases, namel y, rising phase, maintaining phase and recession phase. Besides ratings, most of the App stores also allow usersto write some textual comments as App reviewsSuchreviews can reflect the personal perceptions and usageexperiences of existing users for particula r mobile Apps . Indeed , the review manipulation is one of the most importantperspective of App ranking fraud.

IDENTIFYING LEADING SESSIONS FOR APPS

By analyzing the historical ranking records of mobile Apps, we observe that Apps are not always ranked high in the leader board, but only in some leading events. Note that we apply a ranking threshold K* which is usually smaller than K here because K may be very big(e.g., more than 1000), and the ranking records beyond K* (e.g., 300) are not very useful for detecting the ranking manipulations. Furthermore, we also find that someApps have several adjacent leading events which are close to each other and form a leading session

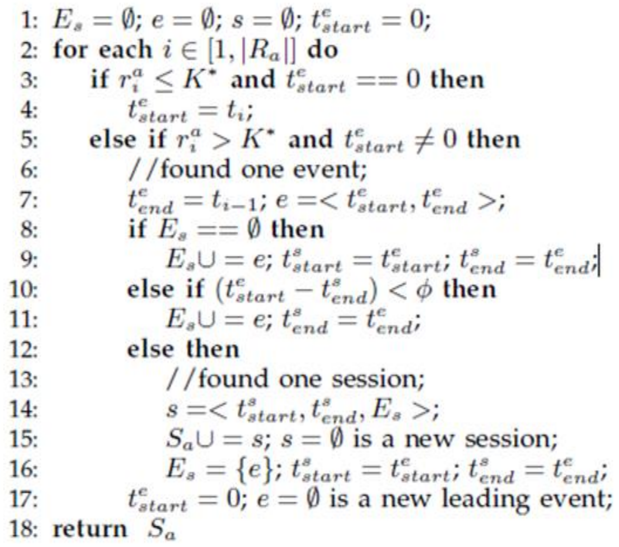


(a) Leading Events (b) Leading Sessions

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.

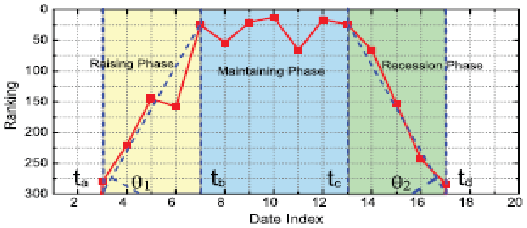
Specifically, Algorithm demonstrates the pseudo code Of mining leading sessions for a given App a



```
1:  $E_s = \emptyset$; $e = \emptyset$; $s = \emptyset$; $t^e_{start} = 0$;
2:  for each $i \in [1, |R_a||$ do
3:      if $r^a_i \leq K^*$ and $t^e_{start} == 0$ then
4:          $t^e_{start} = t_i$;
5:      else if $r^a_i > K^*$ and $t^e_{start} \neq 0$ then
6:          //found one event;
7:          $t^e_{end} = t_{i-1}$; $e =< t^e_{start}, t^e_{end} >$;
8:          if $E_s == \emptyset$ then
9:              $E_s \cup = e$; $t^s_{start} = t^e_{start}$; $t^s_{end} = t^e_{end}$;
10:         else if $(t^e_{start} - t^s_{end}) < \phi$ then
11:             $E_s \cup = e$; $t^s_{end} = t^e_{end}$;
12:         else then
13:             //found one session;
14:             $s =< t^s_{start}, t^s_{end}, E_s >$;
15:             $S_a \cup = s$; $s = \emptyset$ is a new session;
16:             $E_s = \{e\}$; $t^s_{start} = t^e_{start}$; $t^s_{end} = t^e_{end}$;
17:         $t^e_{start} = 0$; $e = \emptyset$ is a new leading event;
18: return $S_a$
```

In Algorithm 1, we denote each leading event *e* and session *s* as tuples < *testart*, *teend* > and < *tsstart*, *ts end*,*Es* >respectively, where *Es* is the set of leading events in session *s*. specifically, we first extract individual leading even *e* for the given App *a* (i.e., Step 2 to 7) from the beginning time. For each extracted individual leading

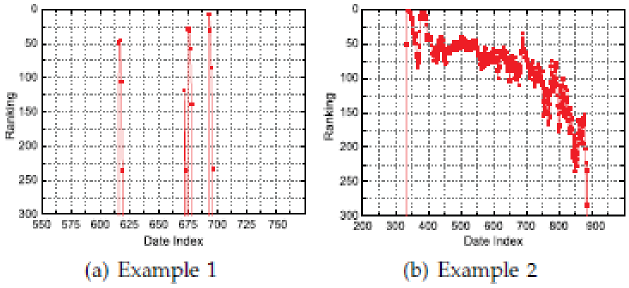
event *e* , we check the time span between *e* and the current leading session *s* to decide whether they belong to the same leading session based on Definition 2 the. Particularly, if (*testart*−*tsend*) < *ϕ*, *e* will be considered as a new leading session (i.e., Step 8 to 16). Thus, this algorithm can identify leading events and sessions by scanning *a*'s historical ranking records only once.



EXTRACTING EVIDENCES FOR RANKING FRAUD RECKONING

RANKING 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 Game loft, may have some leading events with large values of 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. According to the fact the definitions introduced in Section 2, a leading session is composed of several leading the events. Therefore, we should first analyze the basic characteristics of leading events for extracting fraud evidences. By analyzing the Apps' historical ranking records, we observe that Apps' ranking behaviours in a leading event always satisfy a specific ranking pattern, which consists of three the different ranking phases, namely, pising phase, maintaining phase and recession phase. Specifically, in each leading event, an App's ranking first increases to a peak position in the leader board (i.e., rising phase), then keeps those such peak position for a period (i.e., maintaining phase), and finally decreases till the end of the event (i.e., recession phase). Figure shows an example of different ranking phases of a leading event. T Indeed, such a ranking pattern shows an important understanding of leading event. In the following, we form ally define the three ranking phases of a leading event.



(a) Example 1 (b) Example 2

Therefore, for both App developers and marketing Firms, the earlier the ranking expectation meets, the more Money can be earned. Moreover, the after reaching maintaining the expected ranking for a required period, the manipulation will be stopped and the ranking of t malicious App will decrease dramatically. As a result, the suspicious leading events may contain very short rising and recession phases. Meanwhile, the cost of ranking manipulation with high ranking expectations is quite expensive due to unclear ranking principles of App stores and the fierce competition between App developers. Therefore, the leading event of fraudulent Apps often has very short maintaining phase with high ranking positions. Figure (a) shows an example of ranking records from one of the reported suspicious Apps [5]. We can see that his App has several impulsive leading events with high ranking positions. In contrast, the ranking behaviours of abnormal App's leading event may be completely different. For example, Figure 4 (b)

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

shows an example of ranking records from a popular th App "Angry Birds: Space", which contains a leading event with a long time range (i.e. more than one year), especially for the recession phase. In fact, once a normal App is ranked high in the leader board, it often owns lots of honest fans and may attract More and more users the to download. Therefore, this App will be ranked high in hence the leader board for a long time. Based on the above discussion, hence we propose some ranking based signatures of leading sessions to construct fraud Evidences for ranking fraud detection

EVIDENCE 1. As shown in Figure 3, we use two shape parameters θ1 and θ2 to quantify the ranking patterns of the rising phase and the recession phase o App a's leading event e, which can be computed by

$$\mathbb{P}(\mathcal{P}(\lambda_s) \geq |E_s|) = 1 - e^{-\lambda_s} \sum_{i=0}^{|E_s|} \frac{(\lambda_s)^i}{i!}.$$

Where *K_* is the ranking threshold in Definition 1. Intuitively, a large *θ*1 may indicate that the App h been bumped to a high rank within a short time, and a large *θ*2 may indicate that the App has dropped from a high rank the bottom within a short time. Therefore, a leading session, which has more leading events with large *θ*1and *θ*2 values, has higher probability of having ranking fraud. He we define a fraud signature *θs* for a leading session as follows.

$$\overline{\theta}_s = \frac{1}{|E_s|} \sum_{e \in s} (\theta_1^e + \theta_2^e),$$

Where |*Es*| is the number of leading events in session *s*. Intuitively, if a leading session *s* contains significantly Higher *θs* compared with other leading sessions of Apps In the leader board, it has high probability of having Ranking fraud. To capture this, hence we propose to apply Statistical hypothesis test for computing t he th significance Of *θs* for each leading session. We specifically, we define Two statistical hypotheses as follows and compute the P-value of each leading session.

$$\mathbb{P}(\mathcal{N}(\mu_{\overline{\theta}}, \sigma_{\overline{\theta}}) \geq \overline{\theta}_s) = 1 - \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{\overline{\theta}_s - \mu_{\overline{\theta}}}{\sigma_{\overline{\theta}}\sqrt{2}}\right)\right),$$

Where erf(*x*) is the Gaussian Error Function as follows,

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

Intuitively, a leading session with a smaller p-value P has more chance to r eject the HYPOTHESIS 0 and accept HYPOTHESIS 1. This means it has more chance of committing ranking fraud. Thus, we define the evidence as

$$\Psi_1(s) = 1 - \mathbb{P}(\mathcal{N}(\mu_{\overline{\theta}}, \sigma_{\overline{\theta}}) \geq \overline{\theta}_s).$$

EVIDENCE 2

The number of leading events in a leading session, i.e.,|*Es*|, is also a strong signature ranking fraud. For a normal App, the recession phase indicates the fading of popularity. Therefore, after the end of a leading event, it is unlikely to appear another leading event in a short time unless the App updates its version or carries out some sales promotion. Therefore, if a leading session contains much more leading events compared with other leading sessions of Apps in the Leader board, it has high probability of having ranking fraud. To capture this , we define two statistical hypotheses t compute the significance of |*Es*| for each leading session as follows.

◁ HYPOTHESIS 0*: The signature* |*Es*| *of leading session s is not useful for detecting ranking fraud.*

◁ HYPOTHESIS 1*: The signature* |*Es*| *of leading session s is significantly lager than expectation.*

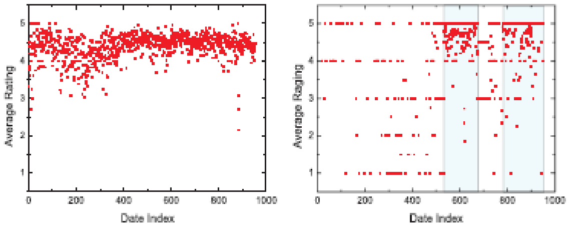
Since |*Es*| always has discrete values, we propose to Leverage the Poisson approximation to calculate the p value

With the above hypotheses. Specifically, we assume |*Es*| follows the Poisson distribution, |*Es*| ∼ *P*(*λs*), where the parameter *λs* can be learnt by the MLE method from the observations of |*Es*| in all Apps' historical leading sessions. Then, we can calculate the p-value as follows,

$$\chi_s = \frac{1}{|E_s|} \sum_{e \in s} \frac{K^* - \overline{r}_m^e}{\Delta t_m^e},$$

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 the example, some Apps created by the famous developers, such as Game loft, may have some leading events with large values of *θ*1 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.



Specifically, after an App has been published, it can Be rated by any user who downloaded it. Indeed, user rating is one of the most important features of App advertisement. An App which has higher rating may attract more users to download and can also be ranked higher in the leader board. Thus, rating manipulation is also an important perspective Of ranking fraud. Intuitively, if an App has ranking fraud in a leading session *s*, the ratings during the time period of *s* may have anomaly patterns compared with its historical ratings, hen which can be used for constructing rating base evidences. For example, show the distributions of the daily average rating of a popular App "face book "Whats App" an a suspicious App discovered by our approach, respectively

EVIDENCE

For a normal App, the average rating in a specific leading session should be consistent with the aveage value of all historical ratings. In contrast, an App with rating manipulation might have surprisingly high ratings in the fraudulent leading sessions with respect to its historical ratings. Here, we define a fraud signature ΔRs for each leading session as follows,

$$\Delta\mathcal{R}_s = \frac{\overline{\mathcal{R}}_s - \overline{\mathcal{R}}_a}{\overline{\mathcal{R}}_a}, \quad (s \in a)$$

where *Rs* is the average rating in leading session *s*, and *Ra* is the average historical rating of App *a*. Therefore, if a leading session has significantly higher value of $\Delta R\ s$ compared with other leading sessions of Apps in the leader board , it has high probability of having ranking fraud. To capture this we define statistical hypotheses to compute the significance of ΔRs for each leading session as follows.

HYPOTHESIS 0*: The signature* ΔRs *of leading session s is not useful for detecting ranking fraud.*

HYPOTHESIS 1*: The signature* ΔRs *of leading session s is significantly higher than expectation.*

Here, we use the Gaussian approximation to calculate the p-value with the above hypotheses. Specifically, we assume ΔRs follows the Gaussian distribution, $\Delta Rs \sim N(\mu R, \sigma R)$, where μR and σR can be learnt by the MLE method from the observations of ΔRs in a ll Apps' historical leading sessions. Then, we can compute the evidence by

$$\Psi_4(s) = 1 - \mathbb{P}\big(\mathcal{N}(\mu_{\mathcal{R}}, \sigma_{\mathcal{R}}) \geq \Delta\mathcal{R}_s\big).$$

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 perspectives of App ranking fraud. Specification before downloading or purchasing new mobile users often firstly 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 the leader board. Although some previous works overview spam detection have been reported in recent years [14], [19], [21], the problem of detecting the local anomaly of reviews in the leading sessions and capturing them as evidences for ranking fraud detection are still under-explored. To this end, here we propose two evidences based on Apps' review behaviours in leading sessions for detecting ranking fraud.

EVIDENCE

Indeed, most of the the review manipulations are implemented by bot farms due to the highcost of human resource. Therefore, review spamers often post multiple duplicate or near-duplicate reviews on the same App to inflate downloads [19], [21]. In contrast,the normal App always have diversified reviews sinceusers have different personal perceptions and usageexperiences. Based on the above observations, here wdefine a fraud signature *Sim*(*s*), which denotes the average mutual similarity between the reviews within leading thsession *s*. Specifically, this fraud signature canbe computed by following steps.

First, for each review *c* in leading session *s*, we remove all stop words (e.g., "of", "the") and normalize verbs and adjectives (e.g., "*plays* → *play*", "*better* → *good*"). Second we build a normalized words vector

−→*wc* =*dim*[*n*] For each review *c* , where *n* indicates the number of all unique normalized words in all reviews of *s*. The specific, here we have *dim*[*i*] = Σ*freqi;c freqi;c*(1 ≤ *i* ≤ *n*),where *frequency* is the frequency of the *i*-th word in *c*.Finally, we can calculated the similarity between two reviews *ci* and *cj* by the Cosine similarity *Cos*(−→*wci* ,−→*wcj*).Thus, the fraud signature *Sim*(*s*) can be computed by

$$Sim(s) = \frac{2 \times \sum_{1 \leq i < j \leq N_s} Cos(\overrightarrow{w_{c_i}}, \overrightarrow{w_{c_j}})}{N_s \times (N_s - 1)},$$

Where *Ns* is the number of reviews during leading session *s*. intuitively, the higher value of *sin*(*s*) indicates more duplicate / near- duplicate reviews in *s*. Thus, if a leading session has significantly higher value of *Sin*(*s*) compared with other leading sessions of Apps in the leader board, it ha high probability of having ranking fraud. To capture this, we define statistical hypotheses to compute the significance of *Sin*(*s*) for each leading session
as follows.

HYPOTHESIS 0*: The signature Sim*(*s*) *of leading session s is not useful for detecting ranking fraud.*

HYPOTHESIS 1*: The signature Sim*(*s*) *of leading session s is significantly higher than expectation.*

Here, we use the Gaussian approximation to compute the p-value with the above hypotheses. Specifically, we assume *Sin*(*s*) follows the Gaussian distribution, *Sin*(*s*) ∼ *N*(*µSin, σSin*), where *µSin* and *σSin* can be learnt by the MLE method from the observations of *Sin*(*s*) in all Apps' historical leading sessions. Then, we can compute the evidence by

$$\Psi_6(s) = 1 - \mathbb{P}(\mathcal{N}(\mu_{Sim}, \sigma_{Sim}) \geq Sim(s)).$$

EVIDENCE RECKONING

After extracting three types of fraud evidences, the next challenge is the way 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 [17], [18], score based models [11], [26] and Dempster-Shafer rules [10], [23]. However, some of these methods focus on learning a global ranking f all candidates. This is not proper for detecting ranking fraud for new Apps. Other methods are based on supervised learning techniques, which depend on the labelled training data and are hard to be exploited. we propose an unsupervised approach based on fraud similarity to combine these evidences Specifically, we define the final evidence score Ψ_(*s*)a linear combination of all the existing evidences a Equation 18. Note that, here we propose to use the linear combination because it ha s been proven to be effective and is widely used in relevant domain such as ranking

$$\Psi^*(s) = \sum_{i=1}^{N_\Psi} w_i \times \Psi_i(s), \quad s.t. \sum_{i=1}^{N_\Psi} w_i = 1,$$

where *N* = 7 is the number of evidences, and weight *wi* ∈[0, 1] is the aggregation parameter of evidence Ψ*i*(*s*). Thus, the problem of evidence aggregation becomes how to learn the proper parameters {*wi*} from the training leading sessions.

DISCUSSION

The experimental data sets were collected from the "Top Free 300" and "Top Paid 300" leader boards of Apple's App Store (U.S.) from February 2, 2010 to September 17, 2012. The data sets contain the daily chart rankings 1 o top 300 free Apps and top 300 paid Apps, respectively. More over each data set also contains the user rating and review information. Table 1 shows the detailed data characteristics of our data sets.

TABLE 1
Statistics of the experimental data.

	Top Free 300	Top Paid 300
App Num.	9,784	5,261
Ranking Num.	285,900	285,900
Avg. Ranking Num.	29.22	54.34
Rating Num.	14,912,459	4,561,943
Avg. Rating Num.	1,524.17	867.12

This is not proper for detecting ranking fraud for new Apps. Other methods are based on supervised learning techniques, which depend on the labelled training data and are hard to be exploited. we propose an unsupervised approach based on fraud similarity to combine these evidence Figures 6 (a) and 6 (b) show the distributions of the number of Apps with respect to different rankings in these data sets. In the figures, we can see that the number of Apps with low the rankings is more than that of Apps with high rankings. Moreover, the competition was between free Apps is the more than that between paid Apps, especially in high rankings (e.g., top 25). Figures 7 (a) and 7 (b) show the distribution of the number of Apps with respect to different number of ratings in these data ets. In the figures, we can see that the distribution of App ratings is not even, which indicates that only a small percentage of Apps are very popular.

RELATED WORK

Generally speaking, the related works of this study be grouped into three categories. The first category is about Web ranking spam detection. Specifically, the

Web ranking spam refers to any deliberate actions which bring to selected WebPages an unjustifiable favourable relevant importance [30].
For example, Ntoulas *et al.* [22] have studied various aspects of content-based spam on the Web and presented a number of heuristic methods for detecting content based spam .Zhou*et al* [30] have studied the problem of unsupervised Web ranking spam detection. Specifically, of they proposed an efficient online link spam and spam detection methods using spamicity. Recently,Spirin *et al.2* [25] have reported a survey on Web spam detection, which comprehensively introduces the principles and algorithm in the literature. Indeed, the work of Web ranking spam is mainly based on the analysis of ranking principles of sear engines, such as Page Rank and query term frequency. This different from ranking fraud detection for mobile Apps

CONCLUDING REMARKS

In this paper, 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 bas 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 that all the evidences can be modelled by statistical hypotheses tests, thus it I s easy to be extended with other evidence from domain knowledge to detect ranking fraud. Finally, we validate the proposed system with extensive experiment on real-world App data collected from the Apple's App stores. Experimental results showed the effectiveness of the propos approach. 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 is related services, such as mobile Apps recommendation, for enhancing user experience.

REFERENCES

[1] http://en.wikipedia.org/wiki/cohen's kappa.
[2] http://en.wikipedia.org/wiki/information retrieval.
[3] https://developer.apple.com/news/index.php?id=02062012a
[4] http://venturebeat.com/2012/07/03/apples-crackdown-on-Appranking-manipulation/.
[5] http://www.ibtimes.com/apple-threatens-crackdown-biggestapp-store-ranking-fraud-406764.
[6] http://www.lextek.com/manuals/onix/index.html.
[7] http://www.ling.gu.se/˜lager/mogul/porter-stemmer.
[8] L. Azzopardi, M. Girolami, and K. V. Risjbergen. Investigating the relationship between language model perplexity and ir precision recall measures. In Proceedings of the 26th International Conference on Research and Development in Information Retrieval (SIGIR'03), pages 369–370, 2003.
[9] D. M. Blei, A. Y. Ng, and M. I. Jordan. Lantent dirichlet allocation. Journal of Machine Learning Research, pages 993–1022, 2003
[10] 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.
[11] 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, pages60–68, 2011.
[12] T. L. Griffiths and M. Steyvers. Finding scientific topics. In Proc.of National Academy of Science of the USA, pages 5228–5235, 2004.
[13] G. Heinrich. Parameter estimations for text analysis. Technical report, University of Lipzig, 2008.
[14] 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.
[15] 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.
[16] A. Klementiev, D. Roth, and K. Small. An unsupervised learning algorithm for rank aggregation. In Proceedings of the 18th European conference on Machine Learning, ECML '07, pages 616–623, 2007
[17] A. Klementiev, D. Roth, and K. Small. Unsupervised rank aggregation with

distance-based models. In Proceedings of the 25thinternational conference on Machine learning, ICML ’08, pages 472–479, 2008.

[18] A. Klementiev, D. Roth, K. Small, and I. Titov. Unsupervised rank aggregation with domain-specific expertise. In Proceedings of the21st international joint conference on Artifical intelligence, IJCAI’09,pages 1101–1106, 2009.

[19] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw.Detecting product review spammers using rating behaviours.

[20] Y.-T. Liu, T.-Y. Liu, T. Qin, Z.-M. Ma, and H. Li. Supervised rank aggregation. In Proceedings of the 16th international conference on World Wide Web, WWW ’07, pages 481–490, 2007.

[21] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos,and R. Ghosh. Spotting opinion spammers using behavioural foot prints. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD ’13, 2013.

[22] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In Proceedings of the15th international conference on World Wide Web, WWW ’06, pages83–92, 2006.

[23] G. Shafer. A mathematical theory of evidence. 1976.

[24] K. Shi and K. Ali. Getjar mobile application recommendations with very sparse datasets. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD ’12, pages 204–212, 2012.

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

[26] 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.