Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.6, Dec. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.6.1499

부분적 관찰정보기반 견고한 안드로이드 앱 추천 기법*

오 하 영,†‡ 구 은 희
아주대학교

POMDP Based Trustworthy Android App Recommendation Services*

Hayoung Oh,†‡ EunHee Goo
Ajou University

요 약

스마트폰의 사용 및 다양한 앱 들의 출시 등이 기하급수적으로 증가되면서 악성 앱 또한 동시에 증가됐다. 기존의 앱 추천 시스템들은 온라인상에서 보이는 다른 사용자들의 평점, 댓글 및 인기 카테고리 등의 정적인 정보 분석을 기반으로만 동작한다는 한계가 있었다. 본 논문에서는 처음으로 스마트폰에서 실제로 사용되는 앱의 동적인 정보들을 현실적으로 사용하여 정적인 정보와 동적인 정보를 동시에 고려하는 견고한 앱 추천 시스템을 제안한다. 즉, 본 논문에서는 앱의 사용되는 시간, 앱의 사용 빈도수 및 앱과 앱 간의 상호 작용과 안드로이드 커널과의 접촉 횟수 등을 측정 가능한 수준에서 부분적으로 반영하여 견고한 안드로이드 앱 추천 시스템을 제안한다. 성능평가 결과 제안하는 기법이 견고하고 효율적인 앱 추천 시스템임을 증명했다.

ABSTRACT

The use of smartphones and the launch of various apps have increased exponentially, and malicious apps have also increased. Existing app recommendation systems have been limited to operate based on static information analysis such as ratings, comments, and popularity categories of other users who are online. In this paper, we first propose a robust app recommendation system that realistically uses dynamic information of apps actually used in smartphone and considers static information and dynamic information at the same time. In other words, this paper proposes a robust Android app recommendation system by partially reflecting the time of the app, the frequency of use of the app, the interaction between the app and the app, and the number of contact with the Android kernel. As a result of the performance evaluation, the proposed method proved to be a robust and efficient app recommendation system.

Keywords: Partially Observable Markov Decision Process (POMDP) based Recommender System, prediction shift

I. Introduction

Over the past decade, people use smart phones for many of the same purposes as desktop computers and mobile phones have evolved from simple devices to sophisticated de- vices that can run third-party software. Phone owners are free to customize their phones by installing third-party applications of their

Received(07. 18. 2017), Modified(11. 06. 2017) , Accepted(11. 17. 2017)
* This work was supported by the Ajou University research fund and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2017R1D1A1B03035557).
† 주저자, hyoh79@gmail.com
‡ 교신저자, hyoh79@gmail.com(Corresponding author)

choosing. Mobile phone manufacturers support third-party application developers by providing development plat-forms and software stores (e.g., in crowded Android Market and Apple App Store) where developers can distribute their applications. In addition to an open API, the Android operating system also provides a rich inter-application message passing system. This encourages inter-application collaboration and reduces developer burden by facilitating component reuse. But even though a result of their popularity and functionality, smart phones are a burgeoning target for malicious activities and all applications can be treated as potentially malicious.

And due to the explosive growth number of mobile applications (apps) with Google's Android Market boasting more than 150,000 applications, it is rather difficult for users to find most proper apps. Therefore, there is an urgent need to provide an effective and robust app recommendation service.

To alleviate these problems, existing many industry solutions have often been proposed that mostly leverage the users' application download history and possibly their ratings to personalized recommend applications that might interest them, such as Amazon's book recommendations [1][2][3].

However, the downloading history of an application is a weak indicator of whether the user really likes that application, particularly if the application is free and the user just wants to try it out. They may never use it again or may have uninstall it that they do not like or feel so-so about.

And using application ratings by asking the users to explicitly assign the rating value as the score to the (user, app) pair suffers from tedious manual labeling. And potential data sparsity problems exist since only a small percentage of users may be willing to consistently rate the applications they use. For example, we often see a widely popular application only receives 2-3% ratings from its users. The limited and possibly outdated rating data thus will reduce the recommendation quality.

Compared with other solutions, the proposed scheme is a online practical scheme considering dynamic operation between inter-app communication for robust and mobile Recommend Systems (RSs). The proposed scheme chose to passively observe the Kernel level and a series of logs sorted by the time stamp according to users' action sequence. Based on them, the proposed scheme defends the possibility of malicious apps and refers how the applications are being used based on inter-app communication with an assumption that the more an application is being used suggests that the more the user likes it.

To the best of our knowledge, the proposed scheme is the first robust mobile application recommendation system that leverages the user's actual application usage patterns and robustness with analyzing inter-application communications. Based on all mobile clients (MCs)' inter-application usage records, the proposed scheme employs an enhanced model based Collective Filtering (eCF) algorithm for individualized robust recommendations. eCF is more scalable and can address better the data sparsity problem.

These recommendations are specific to the user, but leverage data collected from many other users. While not directly exposing the MCs' personal tastes or

behavior patterns, eCF algorithms do implicitly link related applications driven by inter-App communication to provide a foundation for personalized recommendations.

We discuss the proposed scheme's design and implementation on the off-the-shelf Google Android phones, and the evaluation shows that the proposed prediction algorithm provided reasonably accurate usage estimate of the recommended applications after they were installed. We also found the proposed scheme to be satisfied with the MC s interacted with robust recommended applications longer than other applications.

II. Proposed Scheme

In a downlink scenario with arbitrary number of Access Points (APs) and MCs, we propose a practical online algorithm for Trustworthy Mobile App Recommendation Service in Android. Each AP selects the best App candidate online-manner by estimating the App status (i.e., stable or unstable due to Malware) and short-term App satisfaction of MC (i.e., Usage Frequency, Recency, and Duration) based on inter-App communication.

2.1 Estimating the App status

AP exploits the inter-App communication profiling extracted from the integrated system logs including system calls, which are implicitly equivalent to a sophisticated message passing system, in which Intents are used to link applications. That is, we examine the security challenges of App communication from the perspectives of Intent senders and Intent recipients. Since inter-App communication functions from unknown developers could handle private information (e.g., financial data, passwords and personal photographs).

Our intuition on inter-App communication based approach is that we can monitor two criteria (Mean and Standard Deviation of robustness) of App status with the information level concept [4]: Using Eq.(1), the proposed scheme computes Agreementi for each App (i) by setting the mean of the calling number of inter-App communication (i.e., an implicit Intent without the in-tended recipient). Let i denote the target App and j denote the collection of Apps related with App (i) for inter-App communication (i.e., Apps (j) = {App(1),,,App(k) }).

V_i^j is the total calling number of each App(j) by App (i) and Cj is the total number of inter-Apps (j).

In Eq.(2), $Trust_i$ for each App(i) is normalized by V_{max} with respect to C_{max}. After the this phase, finally the proposed scheme estimates the App condition.

$$Agreement_i = \left\| \frac{\Sigma v_i^j}{C_j + 1} \right\|, \quad (1)$$

where $\|\cdot\|$ *is the nearest integer function.*

$$Trust_i = \frac{Agreement_i}{V_{max}} \times C_{max} \quad (2)$$

V_{max} is the maximum calling number value and C_{max} is upper threshold of $Trust_i$ calculated based on global *Standard Deviation* (STD) value used for checking whether the App(i) is in an unstable condition or not. The unstable condition is where App(i) may be malicious since the $Trust_i$ is over C_{max}.

Our intuition of dynamic C_{max} control is as follows: (i) If the STD of the calling

number of inter-App communication is large, we increase C_{max} in proportion to STD. (ii) Otherwise, we decrease C_{max} in proportion to STD. We mathematically express our intuition as follows:

$$C_{max} = \Upsilon \times STD_{global}, \quad (3)$$

where Υ *is set to maximize the detection accuracy by observing* $C_{max} \in \{2,3,4\}$.

In Eq.(3), we set the value of Y after observing the App state using three possible C_{max}. We found that the overall error is minimized when C_{max} is equal to 2 (vise versa to maximize the overall detection accuracy).

2.2 Estimating short-term App satisfaction of MC

In this section we describe the details of latent values for each App for our system model. We first introduce the general idea of the static satisfaction value (SSV) based on the ratings and relating dynamic value (RDV) of each app as well as how they affect the user's App-downloading-using behaviors.

The SSV is obtained when the MC registers the rating value of the opinion about the usefulness or Malware of each App. The RDV (i.e., Frequency, Recency and Duration) only exists after the MC really uses the App as well as installation while the App can call other related Apps in the Kernel level according to users' action sequence. Generally speaking, a large SSV and RDV suggest that the App looks quite attractive and is therefore likely to satisfy a new MC to use it.

Frequency measures how often the M C used the target App in a given time period; Recency measures how recently the MC used the App; and Duration measures how long the MC spent the time with the App. The three values can be calculated considering inter-App communication.

By combining these three values, it can provide a good estimate of how much the MC likes to use the target App with a series of logs sorted by the time stamp.

The usage score of the target app is thus represented as

$$U_{app} = \alpha v_f + \beta v_r + \gamma v_d, \quad (4)$$

where α, β, ɣ are the weights based on their relative importance. And V_f, V_r, V_d are the values of Frequency, Recency and Duration, respectively. The combination of these three measurements reflects the application dynamic taste by the MC and other Apps considering inter-App communication. That is, the applications that have been used more frequently, more recently and more time are likely to be favored by the MC and other Apps regardless of the static review rating value of RSs.

Supposedly the proposed RS is reflected in its additional dynamic descriptions such as SSV as well as the static rating values, RDV. Thus we use such information to directly quantify the vector of each Apps in multi-dimensions and ratings in one-dimension. Specifically, one-dimension of RS has the static rating values while in multi-dimensions, each App has the Mean and SVD based on the calling number of inter-App communication and each pair between the MC and App has U_{app}.

The top-k app recommendation is to select k Apps that are most likely to be used by a new MC. In the proposed scheme, to recommend apps to a MC, an AP selects a number, say k, apps from a

pool of candidate apps and sort them based on the probability that the target MC may more use them based on the enhanced prediction shift (ePS) as follows:

The mathematical form of ePS can be stated as follows:

$$ePS = \frac{1}{N} \sum_{mc \in MC_T} \sum_{i \in I_T} |p'_{mc,i} - p_{mc,i}|, \quad (5)$$

where N is the number of element values in the enhanced RS set,

$p'_{mc,i}$ is the predicted rating value of MC on App i in the enhanced RS set, and $p_{mc,i}$ is that in the one-dimensional set including only rating values, respectively. And let MC_T and I_T be the set of MCs and Apps in the enhanced RS set with App status and static faction, respectively.

Finally the best among a different set of candidate Apps is to achieve the maximum mobile recommendation satisfaction based on potentially the maximum ePS improvement.

2.3 In a downlink scenario with arbitrary number of APs and MCs

As a last step of the proposed scheme, each AP will recommend the best App candidates name by $TruRec_{app,k,I}$ (Trust-aware mobile App Recommendation), K_{th} recommendation set at AP_i. It is given by

$$\begin{aligned} &TruRec_{apps,\kappa,i} \\ &= \max_{apps} \max_{\kappa} P_{app,\kappa,i} \\ &= \max_{apps} \max_{\kappa} \sum_{m=1}^{M} P_{app,\kappa,i}(c_m) \\ &= \max_{apps} \max_{\kappa} \sum_{m=1}^{M} \sum_{app=1}^{n} \{[b^{\kappa}_{app}(c_m) \cdot U_{app} \cdot t^{\kappa}_{app}(c_m) \cdot (1 - \theta PS^{\kappa}_{app}) \\ &\quad]\}, for \quad \kappa = 1, 2, \cdots, \eta \end{aligned} \quad (6)$$

After defining the contribution of K_{th} App candidate to the proper recommendation to a MC(c_m) at AP_i, $P_{app,k,i}(c_m)$, we define the total improvement by the proper Apps recommendation as the sum of the there commendation satisfaction improvements at all MCs(i.e.,c_m=1,…,M)(i.e.,$P_{app,k,i}$).

In general App recommendation system[5][6][7], a target $c_m \in M$ can be recommended the App based on the rating values and obtain wanted the App. For this to be guaranteed, sparse static ratings of all other Apps must only be considered with various off-line approaches such as Collaborative filtering (CF), content-based filtering, hybrid and item-based techniques [8][9][10].

$b^k_{app}(c_m)$ describes the verification of the App state (i.e., Benign or Malware) at the target client c_m as a indicator function (i.e., 1 or 0). If Eq.(1) and (2) are satisfied as mentioned in Section2.1, equals to 1 and otherwise, 0. In a similar way,

$t^k_{app}(c_m)$ is a indicator function. If the app included in K_{th} app recommendation candidate is targeted to client cm, it equals to 1 and otherwise, 0.

Our final objective function is to maximize Eq. (6) about all APs given in Eq. (7).

Maximize :

$$\sum_{\forall i \in AP} TruRec_{apps,\kappa,i} \quad (7)$$

Compared with other solutions, the proposed scheme is completely automatic regardless of requiring static manual input (i.e., review ratings) and adaptive to the changes of the user's application taste.

To the best of our knowledge, our scheme is the first mobile application discovery system that leverages the user's actual application usage patterns as well as robustness online.

III. Theoretical Analysis

Due to the sparsity problems on RSs, the proposed scheme utilizes partial information from the target MC and estimates the additional information about the rest of the MCs. To achieve suboptimal solution, we give theoretical analysis with POMDP (Partially Observable Markov Decision Process) on how good the solution can achieve compared with optimal solution based on MDP (Markov Decision Process).

To model our proposed scheme with POMDP, we define the set of states S, the set of actions A, the immediate reward $r(s_t,a_t)$, the transition probabilities $P(s_{t+1}|s_t,a_t)$, and the cumulative rewards at each time epoch t. If there are MC clients and Apps at any given time slot, state S is represented by an MxN matrix, $S=[a_{i,j}]$ MxN, with binary entries $a_{i,j}\in\{0, 1\}$.

$a_{i,j}=1$ indicates the presence of application app_i at client MC_j, whereas $a_{i,j}=0$ indicate so otherwise. Possible actions by AP are the following: 1)Recommend a App to the target MC; 2)Recommend any Apps to the MCs; and 3)Do not recommend.

In case of the transition probability, we consider the Bernoulli model with false probability at each MC, c_m. Therefore, there is an associated transition probability matrix for each action. The immediate reward $r(s_t,a_t)$ for each pair of a state and an action must be chosen such that the sum of these immediate rewards accurately models our objective. Since our objective is to sub-optimize the maximum mobile recommendation satisfaction, we model the immediate rewards as the satisfaction sum by selecting the best mobile Apps $P_{app,k,i}$ for one or more receivers. In our setting, we know the explicit reward amount $r(s',s)$ when the system moves from states to states'. Namely, we can compute $r(s_t,a_t)$ as the expected immediate reward by taking action a as follows:

$$r(s_t,a_t)=\sum_{s'\in S}P(s'_{t+1}|s_t,a_t)r(s'_{t+1},s_t) \tag{8}$$

And we can get the cumulative rewards by using policy π, a sequence of actions at every time step, in Eq. (9).

$$R_j^{\pi}(s_j)=E_{s_j}^{\pi}\left(\sum_{j=t}^{T-1}r_j(s_j,a_j)+r_T(s_T)\right) \tag{9}$$

Finally, the optimal policy

$$\pi^{\star}=\arg\max_{a\in A}R_j^{\pi}(s_j) \tag{10}$$

can be solved using the Backward Induction Algorithm (BIA) to produce the maximum final cumulative reward.

IV. Performance Evaluation

The proposed scheme employs MCs-Access Points (APs) architecture, where all MCs per each AP collects the application's usage records based on inter-application communication and periodically upload them to the AP. The AP makes multi-dimension RS and runs the eCF with Eq.(6) that calculates recommendations for all MCs per the AP.

With a close check on the real data, we reveal that Malware generally leverages the calling number of inter-Apps on top of Kernel level. Therefore, using the information level concept with the calling number of inter-Apps we can exclude the unstable Apps before selecting the best app recommendation.

We have performed extensive empirical evaluations to verify the performance of our proposed scheme. The experimental results demonstrate that the proposed scheme achieves the best performance against comparable robust RS schemes such as principal component analysis (PCA) approach and Least Trimmed Squares Matrix Factorization (LTSMF) approach in terms of Hit Ratio (HR) over the various App attack strategies. The three algorithms compute similarity between Apps and return K Apps as a recommendation to the users using Eq. (6) of [4]. We simulated with 50 target Apps, then averaged the Hit Ratios (HRs).

Fig. 1 shows the fraction of targeted App to be included in K-top recommendation. Since the smaller value of the HR is low sensitivity between the correct K Apps and the recommended K Apps, we show that the proposed scheme is better than other schemes with the lowest HR.

In addition, we derived the error rate of the proposed scheme with other schemes as shown Table.1. Through the calculus of

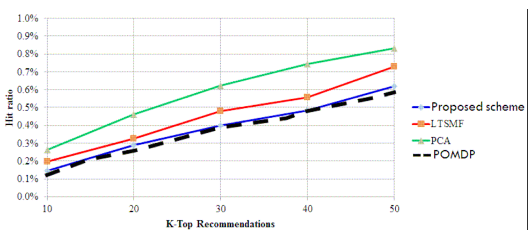

Fig. 1. Hit Ratio

Data set	LTSMF	PCA	POMDP
	0.9336	1.0123	0.272

the MAE (Mean Absolute Error), we verified the proposed scheme is the lowest error rate than other schemes.

V. Conclusion

Due to the huge number of mobile apps and malicious app possibility problems in Android, it becomes necessary to provide users a robust and satisfied app recommendation service. Previous work leveraged the user's application download history or static ratings. However, these are a weak indicator of the user's robust application taste. In light of these problems, in this paper, we first pursue a trusted mobile app recommendation based on the Kernel level and a series of logs sorted by the time stamp with inter-App communication as well as static information.

References

[1] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," Internet Computing, IEEE, pp. 76-80, vol. 7, no. 1, Jan. 2003.

[2] Zhibo Wang, Jilong Liao, Qing Cao, Hairong Qi and Zhi Wang, " Friendbook: A Semantic-Based Friend Recommendation System for Social Networks," Mobile Computing, IEEE Transactions on, pp. 538 - 551, vol. 14, no. 3, Mar. 2015.

[3] Sumedh Sawantand Gina Pai, "Yelp Food Recommendation System," Stanford University, pp. 1-5, Jan. 2013.

[4] G. Noh, Y.-m. Kang, H. Oh, and

C.-k. Kim, "Robust Sybil attack defense with information level in online Recommender Systems," Expert Systems with Applications, pp. 1781-1791, vol. 41, no. 4, Mar. 2014.
〔5〕 Paarijaat Aditya et al., "EnCore: Private, Context-based Communication for Mobile Social Apps," MobiSys '14 Proceedings of the 12th annual international conference on Mobile systems, applications, and services, pp. 135-148, June.2014.
〔6〕 Erika Chin et al., "Analyzing Inter-Application Communication in Android," MobiSys '11 Proceedings of the 9th international conference on Mobile systems, applications, and services, pp. 239-252, July. 2011.
〔7〕 Bo Yan and Guanling Chen, "AppJoy: Personalized Mobile Application Discovery," MobiSys '11 Proceedings of the 9th international conference on Mobile systems, applications, and services, pp. 113-126, July. 2011.
〔8〕 B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness," Journal ACM Transactions on Internet Technology (TOIT), pp. 1-41, vol. 7, no. 4, Oct. 2007.
〔9〕 Z. Cheng and N. Hurley, "Robust collaborative recommendation by least trimmed squares matrix factorization," in 2010 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 105-112, Oct. 2010.
〔10〕 Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol. 42, no. 8, pp. 3-37, Aug. 2009.

〈저자소개〉

오 하 영 (Hayoung Oh) 정회원
2002년 2월: 덕성여자대학교 컴퓨터공학과 졸업
2006년 2월: 이화여자대학교 컴퓨터공학과 석사
2013년 2월: 서울대학교 컴퓨터공학과 박사
2010년 4월~2010년 10월: U.C. Berkeley 방문연구원
2013년 3월~2013년 8월: 서울시립대학교 연구교수
2013년 9월~2016년 8월: 숭실대학교 전자정보공학부 조교수
2016년 9월~현재: 아주대학교 다산학부대학 조교수
〈관심분야〉 소셜 정보망, 추천시스템, 무선 네트워크 및 비디오 스트리밍

구 은 희 (EunHee Goo) 정회원
2002년 8월: 단국대학교 전자컴퓨터공학부(공학사)
2004년 8월: 단국대학교 전자컴퓨터공학과(공학석사)
2009년 8월: 단국대학교 전자컴퓨터공학과(공학박사)
2011년 3월~2013년 2월: 서일대학교 정보통신과 강의전담 교수
2013년 3월~2014년 9월: ㈜도넛시스템 LSI 이미징사업부 책임 연구원
2014년 10월~2016년 8월: ㈜이너트론 이동통신연구소 수석 연구원
2016년 9월~현재: 아주대학교 다산학부대학 조교수
〈관심분야〉 정보보호, 암호 알고리즘, 서비스로서의 보안(ASCaaS), 소프트웨어 교육