Mining Usage Data from Large-Scale Android Users: Challenges and Opportunities

Xuan Lu[1] Xuanzhe Liu[1†] Huoran Li[1] Tao Xie[2] Qiaozhu Mei[3] Dan Hao[1] Gang Huang[1] Feng Feng[4]
[1]Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, PRC
[2]University of Illinois at Urbana-Champaign, [3]University of Michigan,[4]Wandoujia Lab, Beijing, China
{luxuan, xzl, lihuoran}@pku.edu.cn, taoxie@illinois.edu
qmei@umich.edu, {haodan, hg}@pku.edu.cn, jackfeng@wandoujia.com

ABSTRACT

Mining usage data from a large number of Android users can assist various software engineering tasks. In collaboration with Wandoujia, a leading Android app marketplace in China, we have conducted a large empirical analysis based on mining app usage behaviors collected from millions of Android users. Our empirical findings can provide implications, challenges, and opportunities to app-centric software development, deployment, and maintenance.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics-complexity measures, performance measures

Keywords

Mobile apps, user behavior analysis

1. INTRODUCTION

As of 2015, millions of apps have been developed and made available in app marketplaces such as the Apple Store and Google Play, which have received billions of downloads. Numerous app developers have profited from the revenues generated by the downloads and usages of their apps. Thus, the app-centric software development, deployment, and maintenance are important software engineering tasks.

A number of important questions in different cycles of app development remain unanswered. For example, how do users select, manage, and use apps? How do apps perform in terms of traffic and energy consumption? How can developers find potential bugs in apps and better allocate their resources to improve the quality of the apps? Measuring and analyzing the user behavior data of apps can help address these questions.

We have conducted an empirical analysis [1] based on mining app usage behaviors collected from millions of users, with a real-world data set from Wandoujia, a leading Android app marketplace in China. We have measured the app popularity, usage patterns, network performance, and device-specific behaviors. Based on the data set, our recent work [2] has helped address prioritizing device models for individual apps to improve app developers' resource allocation, given the significant fragmentation of Android device models.

In this paper, we summarize our findings and discuss related challenges and opportunities for software engineering tasks.

†Corresponding author.

MobileSoft '16 May 16-17 2016, Austin, TX, USA

ACM ISBN 978-1-4503-4178-3/16/05.

DOI: http://dx.doi.org/10.1145/2897073.2897721

2. DATA SET

The app usage data under analysis are collected through a commercial Android app management tool developed by Wandoujia, a leading Android marketplace in China. Wandoujia was founded in 2009 and has grown into one of the largest Android app marketplaces in the world, with over 250 million users and 1.5 million free Android apps as of year 2015. Wandoujia provides a native management app, through which users can manage the apps on their devices, e.g., downloading, searching, updating, and uninstalling apps. Users can also rate/review apps via the Wandoujia management app. Beyond these basic features, the Wandoujia management app includes some optional features that can monitor and optimize system-wide activities. These features include network-statistic collection, permission monitoring, content recommendation, etc. All features are developed upon Android system APIs and do not require the "root" privilege. Users can opt in and out these features.

We focus on three months of app usage data collected from July 1st, 2014 to September 30th, 2014. The data cover 4,775,293 unique users, 16,602 device models, and 238,231 apps. A sample of the data set has been released in our previous work [1].

3. MINING RESULTS AND FINDINGS

We briefly summarize the mining tasks and findings as follows. We identify some implications for different stakeholders in software engineering, including app developers, appstore operators, and network service providers.

3.1 App Popularity

Motivation. We aim to figure out which apps are really popular and which ones may have potentially "fake" downloads. We measure the number of downloads of an app, the number of unique devices that ever subscribed an app, the traffic of an app, and the browsing time of an app. These metrics can offer comprehensive knowledge of app popularity.

Findings. Not very surprisingly, the distributions of all these measurement results comply with the Pareto principle, i.e., a small portion of apps account for substantial popularity. Thus, marketplace operators and network providers can identify which apps need to be prioritized in resource allocation, e.g., they can organize an efficient caching or prefetching mechanism to enable fast downloads and delivery for these apps. However, we also find that some apps can pursue fake downloads. For example, an app subscribed by only 4 unique devices gained more than 900 downloads in a week. Appstore operators need to be cautious about these apps.

3.2 App Selection and Abandonment

Motivation. Based on the records of downloading, updating, and uninstalling apps, we explore why some apps are installed and how an app is likely to be uninstalled. The measurement results can help developers know user attitudes towards their apps, improve the apps, and adapt to the user requirements accordingly.

Findings. By using the metric of *Jaccard Similarity Coefficient*, we find that two apps from the same vendor and the same category are more likely to be co-selected. We also find that apps providing related functionalities are more likely to be selected together. These findings can help appstore operators improve their recommendation systems. In addition, developers can seek opportunities to assemble several relevant apps to complete a task.

We study the uninstallation patterns of apps by devising a new metric called the *installation/uninstallation ratio*, denoted as *I/U ratio*. This metric represents the ratio of #downloads to #uninstalls, directly informing how much an app is abandoned by users. Combining the temporal information with the *I/U ratio*, we find that about 40% abandoned apps can "survive" for less than a day, and about 93% can survive for less than a week. In addition, a positive correlation exists between *I/U ratio* and lifecycle of abandoned apps, and thus *I/U ratio* can be an indicator of how soon an app will be uninstalled. Such measurements can be used to predict how a new app can be adopted by the users based on the *I/U ratio* of other similar apps, so that the app developers and appstore operators can negotiate the investments on ranking the app.

3.3 App Performance in Network Activities

Motivation. Network activity patterns can identify network-intensive apps that are concerned by different stakeholders. Based on the records of network activities in a fine granularity, we can investigate patterns in terms of the data traffic and access time under Wi-Fi/cellular in foreground/background, respectively.

Findings. Besides identifying some "network-intensive" apps, we find some apps that may "suspiciously" keep continuous network connections or consume traffic at the background. We are also surprised to find that a large number of apps keep long-lived TCP connections at the background after they are launched. These findings can help appstore operators locate some "*problematic*" apps, and help users choose alternative apps to reduce potential loss of traffic and energy caused by these apps. However, currently we cannot precisely locate where the background activities of an app come from and whether they are justified, either by the ad networks or the normal functionalities of the app itself. We need better profiling techniques to detect an app's abnormal background network activities and remove them from the app.

3.4 Device-specific App Usage

Motivation. With the heavy fragmentation of Android devices, the diversity of users can be identified by the device models. We hypothesize that a device model used by a user can reflect the level of hardware specifications and the user's financial background. We then explore how the selection of device models can impact user behaviors on app usage.

Findings. We find that users of higher-end device models tend to use more apps and consume more Wi-Fi traffic. Users from different groups of device models have quite significantly various patterns in consuming cellular traffic and selecting apps providing similar functionalities. For example, users of higher-end device models prefer `Chrome` browsers while users of lower-end device models prefer `UCWeb`. Based on such findings, appstore operators can make more-accurate app recommendations. App developers can target specific user groups to optimize user experiences.

3.5 Device Model Prioritization

Motivation. Selecting and prioritizing major device models are critical for app developers to select testbeds and optimize resources such as marketing and quality assurance. Based on the hypothesis that the usage of an app on a specific device model reflects the importance of that device model for this app, we propose a collaborative filtering approach [2] to predict the usage of an app on different device models, even if the app is entirely new (without its actual usage in the market yet), based on the usage data of a large collection of apps.

Findings. We predict the rank of a device model according to the *browsing time* that the device model holds for the target app based on "known" similar top apps in the same app category. Comparing with top device models reported by market-share reports such as AppBrain[1], our approach performs far better in terms of device model hits, browsing time coverage, and average precision of prediction.

Acknowledgment

This work was supported by the High-Tech Research and Development Program of China under Grant No.2015AA01A202, the Natural Science Foundation of China (Grant No. 61370020, 61421091, 61572051, 61528201). Tao Xie's work was supported in part by National Science Foundation under grants no. CCF-1349666, CCF-1409423, CNS-1434582, CCF-1434596, and CNS-1513939, and a Google Faculty Research Award. Qiaozhu Mei's work was supported in part by the National Science Foundation under grant No. IIS-1054199.

4. REFERENCES

[1] H. Li, X. Lu, X. Liu, T. Xie, K. Bian, F. X. Lin, Q. Mei, and F. Feng. Characterizing smartphone usage patterns from millions of Android users. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement, IMC 15*, pages 459–472, 2015.

[2] X. Lu, X. Liu, H. Li, T. Xie, Q. Mei, D. Hao, G. Huang, and F. Feng. PRADA: Prioritizing android devices for apps by mining large-scale usage data. In *Proceedings of the International Conference on Software Engineering, ICSE 16*, 2016.

[1] http://http://www.appbrain.com/