Voting with Their Feet: Inferring User Preferences from App Management Activities

Huoran Li[1], Wei Ai[2], ‡Xuanzhe Liu[1], Jian Tang[3], ‡Gang Huang[1], Feng Feng[4], ‡Qiaozhu Mei[2]
[1] Key Lab of High-Confidence Software Technology, MoE, Peking University, {lihuoran, liuxuanzhe, hg}@pku.edu.cn
[2]University of Michigan, {aiwei, qmei}@umich.edu
[3]Microsoft Research, jiatang@microsoft.com, [4]Wandoujia Lab, jackfeng@wandoujia.com

ABSTRACT

Smartphone users have adopted an explosive number of mobile applications (a.k.a., apps) in the recent years. App marketplaces for iOS, Android and Windows Phone platforms host millions of apps which have been downloaded for more than 100 billion times. Investigating how people manage mobile apps in their everyday lives creates a unique opportunity to understand the behavior and preferences of mobile users, to infer the quality of apps, and to improve the user experience. Existing literature provides very limited knowledge about app management activities, due to the lack of user behavioral data at scale. This paper takes the initiative to analyze a very large app management log collected through a leading Android app marketplace. The data set covers five months of detailed downloading, updating, and uninstallation activities, involving 17 million anonymized users and one million apps. We present a surprising finding that the metrics commonly used by app stores to rank apps do not truly reflect the users' real attitudes towards the apps. We then identify useful patterns from the app management activities that much more accurately predict the user preferences of an app even when no user rating is available.

General Terms

Experimentation

Keywords

Mobile apps, App management activities, Behavior analysis

1. INTRODUCTION

The prevalence of smartphone applications (a.k.a., apps) has introduced an evolutionary change to people's everyday lives. By 2014, the Apple App Store and the Google Play have hosted more than a million apps and have collected over 100 billion downloads [4, 5]. Many apps are also hosted through other marketplaces such as the Amazon App Store, the Samsung App Store, the F-Droid [1], and the Fetch [2]. In regions where the native app marketplaces are inaccessible, these third party marketplaces play a major role in facilitating users to find, download, and manage mobile apps.

To help users find and explore high-quality apps, most app marketplaces gather user-input ratings of apps, either in the format of like/dislike votes, numerical ratings, or free-text comments. Much work has been done to analyze these ratings [26]. However, among the hundreds of millions of users who downloaded and used these apps, only a small proportion have left reviews [25]. While popular apps are rated heavily, the majority of apps, especially new apps receive very few or even no ratings, making them invisible from the users even if the quality is high. Biased, fake, paid off, and malicious reviews also commonly exist, which compromises the credibility of user-input ratings [29].

In this paper we study a different signal, the signal of users voting an app using their feet. We hypothesize that users who like an app manage it differently from those who don't. Users' preference towards an app can be revealed by the way they manage the app. These actions are just like votes made by the users. Indeed, it is reported that 80 to 90 percent of mobile apps were downloaded, used only once, and then deleted by a user*. These users, although few of them bother to leave a review, are clearly not in favor of the app. Instead, other users are much more engaged, who download an app, keep to date with the updates, and immediately install it back after they get a new device or update the operation system.

More generally, these app-managing activities can include searching, downloading, installing, updating, uninstalling, and rating an app. They provide much richer, and arguably less biased indicators of the actual preferences of the users about the apps. In other words, if one can infer user preferences from behavioral patterns in these app management activities, one could provide an accurate and unbiased indicator of the user preferences and the quality of apps *even if they are not rated by any users.* Indeed, many marketplaces count how many times an app has been downloaded (or how many users have downloaded the app), hoping to provide an evidence of the quality of the app when the ratings are rare. These simple counts, however, may not really reflect the users' preferences when they are interpreted apart from the context: the actual sequences of activities of the users (e.g.,

‡Corresponding author.
Huoran Li and Wei Ai contributed equally to the work.

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
http://dx.doi.org/10.1145/2872427.2874814 .

*http://www.digitaltrends.com/mobile/16-percent-of-mobile\-userstry-out-a-buggy-app-more-than-twice/

a user may download an app but discard it right away). It is also a metric that can be easily manipulated (e.g., through fake users or multiple downloads from the same user).

A deeper understanding of how the quality of apps can be inferred from user activities is critical. This requires analyzing the activity sequences of a large number of real users managing a large number of apps, a data set that does not exist in literature. Indeed, previous studies rely on data sets of user activities collected from small groups of volunteers, usually limited to hundreds of users and devices [33, 32] due to the inability to collect such large-scale behavioral data.

Recent developments of app marketplaces have opened the possibility to collect user activities at a much larger scale. Many marketplaces provide their own native smartphone apps through which their users can manage apps installed on their devices. Through such a tool, sequences of app management activities are collected from millions of users. Even though only particular types of user activities are recorded (e.g., downloading, updating, and uninstalling apps), such large-scale behavioral data sets already present brand-new opportunities to data miners. Knowledge discovered from these data sets provides not only a better understanding about the behavior of mobile users, but also insights for the app marketplaces to assess the quality of apps and to provide effective recommendations to their users.

In this paper, we present an analysis of a very large collection of app management activities collected from millions of Android users[†]. The data is collected through a leading app marketplace in China, called Wandoujia[‡]. Wandoujia now serves near half a billion registered users, 50 million of which are active on a daily basis. Developers can upload and publish their apps through Wandoujia, and consumers can search, download, update, install, uninstall, and rate apps through its own native app. The data studied in this paper covers timestamped activities of downloading, updating, and uninstalling over **1 million** apps by over **17 million** anonymized users, spanning **five months** (May 1, 2014 - September 30, 2014). Based on the largest data set to date, we present a comprehensive study of how users manage mobile apps and how their preferences can be inferred from these activities.

In summary, we present the following contributions:

- The first empirical analysis on app management activities collected from about **17 million** Android users. We characterize how smartphone users manage their apps.
- A descriptive analysis of how app management activities correlate with user ratings of these apps. Surprisingly, we find that more downloads do not correlate with higher ratings, especially when ratings are rare.
- We present some sequential patterns of user management activities that are actually correlated with online ratings of apps, which can be used as features to predict the ratings of apps.
- We present a systematic evaluation of how well the indicators from app management activities can be combined with the state-of-the-art machine learning algorithms and predict the quality of apps.

The rest of the paper is organized as follows. We first relate our study to existing literature in Section 2. Section 3 describes the data set and presents a descriptive analysis of app management activities. Section 4 identifies patterns of app management activities that are correlated with the user ratings of apps. Section 5 presents a prediction analysis to measure how much these activity patterns improve the ranking of apps. We conclude in Section 6.

2. RELATED WORK

To the best of our knowledge, we make the first empirical study on app management activities at the scale of millions of users and apps. Our work is in general related to the literature of analyzing user preferences and assessing app quality. Recently, analyzing behavioral data collected from smartphones or app marketplaces has attracted much attention. Considering the sources of user behavior data, existing studies can be categorized into three aspects:

• **User reviews**. Like other online reputation systems, app marketplaces such as Apple App Store and Google Play have accumulated a large volume of user reviews. Lin et al. presented the AR-Miner [12], which collects narrative user reviews and groups them through topic modeling. AR-Miner prioritizes the narrative reviews using a ranking scheme. Fu et al. presented the WisCom [18], a system that analyzes millions of user ratings and narrative comments from app marketplaces. Assessing quality of apps solely based on online ratings suffers from data sparseness (e.g., the average number of reviews per app is small and most apps are rarely rated) and various types of selection biases.

• **Smartphone usage data**. A less biased way to evaluate an app is through observing how the app is actually used by real users. Researchers build auxiliary apps to monitor the activities and performance of other apps installed on the same device. Ravindranath et al. developed the AppInsight [34], a system that instruments mobile app binaries to automatically identify and characterize the critical paths in user transactions across asynchronous-call boundaries. It collects app usage data such as launching frequency, traffic volume, and access time. The data can provide sufficient information to infer user preferences and interests. However, such a "usage monitor" app is usually deployed as a system-level service [7], and not many users are willing to voluntarily install them on their devices. This is a major obstacle to conduct this type of research at scale. Although some widely-deployed commercial apps such as Flurry [3] and PreEmptive [6] can also be used to monitor app usage, little existing work has been reported using their data sets. One exception is Shi and Ali's work which analyzed app usage of a hundred thousand users collected by the GetJar system, in order to provide personalized recommendations [36].

• **Field studies**. Another common approach analyzes user behavior through field studies. A few field studies were conducted based on the LiveLab data sets [39, 33, 32]. Rahmati and Zhong conducted a longitudinal study to collect the network usage data of 24 users, which were used to analyze how users use the network on their smartphones. They also collected actual usage data from 14 teenage smartphone users from a four-month field study [33]. The same group of authors presented a study involving 34 iPhone 3GS users, reporting how users with different economic background use smartphones differently [32]. Many other studies have been done based on similar fields studies, reporting the diversity

[†]Our study has been approved by the research ethnics board of the Institute of Software, Peking University. The data is legally used without leaking any sensitive information. A sample of the data set has been released along with our IMC'15 work [24].

[‡]`http://www.wandoujia.com/`

of cellular usage from different user groups, different app categories, and different OS platforms [22, 13, 9, 10, 15]. For example, similar apps (e.g., weather apps) may have a significant difference in the consumption of network traffic [35]. In most of these field studies, smartphones must be provided to human subjects. Similar to collecting usage data from instrumented devices, these field studies normally only involve a small group of subjects and are hard to scale.

Compared to existing approaches, we study the actual activities of millions of users. Such behavioral data have a much higher coverage and less bias than user ratings and involve a significantly larger user population than field studies. The availability of such a data set provides a unique perspective of understanding user preferences on smartphone apps.

3. A DESCRIPTIVE ANALYSIS

We start with describing the data used in this study. For the completeness of presentation, we include some results that are already reported in our recent poster paper [23].

3.1 The Wandoujia Data Set

Founded in 2009, Wandoujia has grown to be a leading Android app marketplace in China. Up to 2014, Wandoujia has hosted more than 1.5 million apps and has been accessed by more than 200 million Android devices. Users can access Wandoujia via its Website, its Windows client, and its native management app. The Wandoujia management app works as a system service and functions just like Apple App Store and Google Play. Through the app a user can conduct various activities to manage the apps on their devices, i.e., browsing and searching apps, installing and uninstalling apps, as well as checking and installing updates of apps. Upon the permission of users, the Wandoujia management app logs these management activities and uploads the logs on its own private and secure server.

In our study, we collected app management activities of Wandoujia users spanning five months from May to September, 2014, covering more than 17 million users (actually devices) and 1 million apps. A timestamped record is logged whenever a user downloads, updates or uninstalls an app via the Wandoujia management app. The total numbers of users, apps, and activity log entries involved in our study are reported in Table 1. There are on average 227.6 management activities collected per app, which is 13.9 per device. To protect user privacy, all devices are de-identified. Besides app management activities, we also collected the complete set of online ratings users have ever posted on Wandoujia, with an average of 4.01 per app.

Table 1: Summary statistics of the data.

# of Users (Devices)	17,303,122
# of Apps	1,054,969
# of Activities	240,108,930
# of Online Ratings	4,225,153

Note that similar to Apple App Store and Google Play, the Wandoujia management app records the activities of only the apps that are actually installed, and only the activities that are conducted through the Wandoujia app. We therefore do not distinguish the activities of "downloads" and "installations." When an app is directly downloaded from the Web and installed (i.e., not through the Wandoujia app), this downloading activity is not logged. Similarly, only updating and uninstallation actions conducted through the Wandoujia app are logged. Therefore, our data set may miss some apps or management activities of certain apps if they are not conducted through the Wandoujia management app.

It is also possible that a user's activity sequence spans longer than 5 months. There may be incomplete activity sequences of certain users and certain apps in our data set. Nevertheless, given the longitudinal collection and the very large volume of activities, we anticipate that these possible limitations would not have a significant effect on our analyses. Below we present some characteristics of app management activities in this data set.

Indeed, the usage information of the apps can also be very important, such as how often is the app used, as it provides insight into an app's quality. However, such information is not captured by the Wandoujia management app. In addition, the goal of this paper is to explore the correlation between **management activities** and app qualities, so the usage information is not included in our work.

3.2 Distributions of Management Activities

We first report a series of distributions that may help validate the representativeness of the data. We start with the distributions of the popularity of apps. The popularity of an app can be measured using either the total number of downloads, the most recently downloads (e.g., in the past week or month), or the total number of users. When an app is rarely rated, a marketplace usually provide such a popularity measure as an indicator of the quality of the app. Figure 1 plots the distribution of the number of users per app (i.e., number of devices on which at least one activity of that app has been logged). The popularity of apps follows a typical *Power Law* distribution on this log-log plot. More than 95% apps are downloaded by less than 1,000 devices. A few apps are downloaded by millions of devices. This distribution is consistent with observations in many other behavior data at the population level [8], which usually indicate an effect of "rich get richer." Indeed, the highly-ranked apps may attract even more users while barely-rated apps are buried in the long tail. Similarly, the number of ratings an app receives also presents a typical power law distribution, which we omit here for the sake of space.

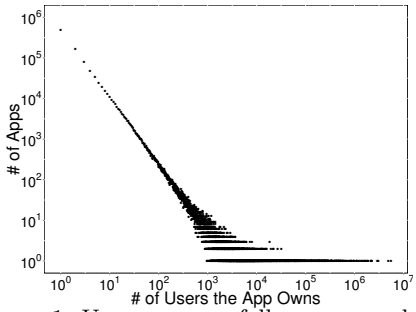

Figure 1: Users per app follow a power law.

Figure 2 plots the distribution of the number of apps that a user installs on her device. Intuitively, many users use only a few apps, while a few others try out a large number of apps. The distribution obeys a power law in its tail distribution, which is also a typical phenomenon of user behaviors [28]. We'd like to point out that the number of apps installed on a device is likely to be underestimated as lots of devices have pre-loaded apps and users might install apps through channels other than Wandoujia. This might have made the

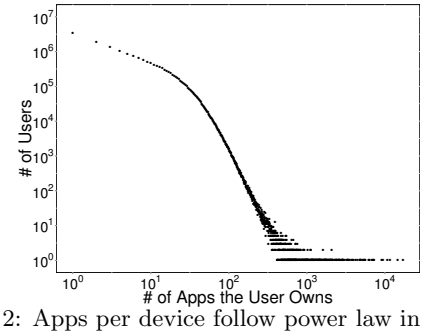

Figure 2: Apps per device follow power law in the tail.

distribution not follow one power law distribution over the entire range.

We then look at how the users manage their apps throughout a day. Figure 3 plots the percentages of user activities over 24 hours of a day. The common sense assumes that a download and an updating activity reflect a user's preference towards an app and an uninstallation indicates that the user is not in favor of the app. We thus aggregate downloading and updating activities (solid, blue line) in this plot and compare them with uninstallations (dashed, red line). All timestamps are converted to the Beijing Time (UTC+8) as most of Wandoujia users are in China.

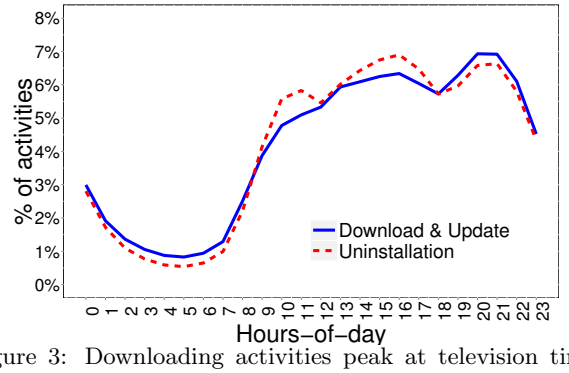

Figure 3: Downloading activities peak at television time; uninstallations present three peaks during the day.

As one may anticipate, app management activities start to increase sharply from 7 a.m. and stay high through out the day. Both downloading/updating and uninstallation activities fall around lunch and dinner. It is interesting to observe clear peaks of both types of activities, indicating that most users may have a routine schedule for managing their apps.

To further investigate whether the users do have a routine schedule of app management, we plot the distribution of the **time intervals** between any two consecutive activities of the same user (Figure 4). Most consecutive activities are conducted within less than a hour (the leftmost data point), which is not surprising. These are likely to be activities conducted in the same session (e.g., updating a batch of apps). However, when the intervals are larger we observe rather surprising patterns. There is a peak at every 24 hours, and between two peaks the time interval distributes as a reversed bell-shaped (either Normal or Poisson) distribution. This suggests that a user does have a routine time period of a day for housekeeping - she may not do it every day, but whenever she does it is likely to be close to the same time of the day. If we plot the intervals by days instead of by hours, it follows an exponential distribution (the red line).

Many of the above results are consistent with the common sense, indicating the validity and representativeness of our

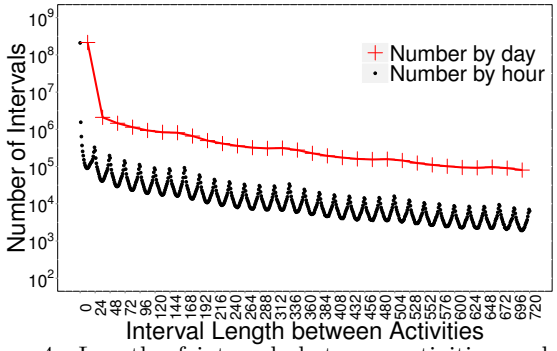

Figure 4: Length of intervals between activities peaks at every 24 hours. Intervals in days follow an exponential distribution (note the y-axis in log scale).

data set. Some of them are quite interesting, providing insights to both our following analyses and the marketplaces and app developers.

4. INFERRING USER PREFERENCES

We learned from Table 1 that the average number of management activities per app in 5 months is 50 times larger than the average number of ratings in 5 years. This motivates us to explore whether the abundant user activities could be utilized to infer the preferences of users and the quality of apps, either as a complement or a surrogate of the scarce user ratings. We start by demonstrating the limitations of user ratings.

4.1 Limitations of User Ratings

Intuitively, an app is considered to be of high quality if it is reviewed highly by its users. In practice, however, there are quite a few problems of directly using such a straightforward metric to assess app quality, as it may suffer from data sparseness and biases.

Most app marketplaces allow users to explicitly rate the apps they installed. For example, Google Play allows users to rate apps in a 1 to 5 scale where 1-star refers to the lowest quality and 5-star refers to the highest. Wandoujia allows users to simply tag an app with "*like*" or "*dislike*."

For comparison purposes, we crawled user ratings of all apps from both Wandoujia and Google Play. In this way, given a specific app, we can synthesize its average ratings on both marketplaces. Figure 5 presents a Venn diagram of the apps rated at the two marketplaces. Surprisingly, among the 1 million apps, only a small portion (<2%) are rated at both two marketplaces. Others do not receive any rating on either one or two of the sites.

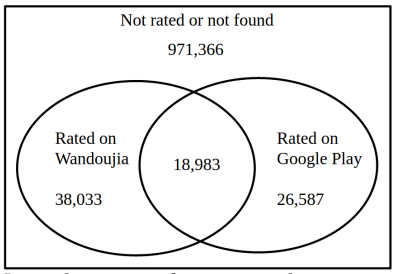

Figure 5: Venn diagram of user rated apps on Wandoujia and Google Play. Only a few are rated on both sites.

For the 18,983 apps that are rated by the users at both marketplaces, we are able to compare their ratings. We

explore the correlation of the numbers of ratings an app receives on the two sites, which is plotted in Figure 6. Generally, the numbers of ratings per app at the two marketplaces are positively correlated. However, there are considerable biases, appearing as the noticeable vertical lines on the left and horizontal lines at the bottom of the plot. These data points refer to the apps that have many ratings on Google Play but very few ratings on Wandoujia, or the opposite. Some of such biases are due to language or cultural differences, and some of these are introduced by regional barriers. For example, the `Facebook` app has 25,169,686 ratings on Google Play but only 1,644 on Wandoujia.

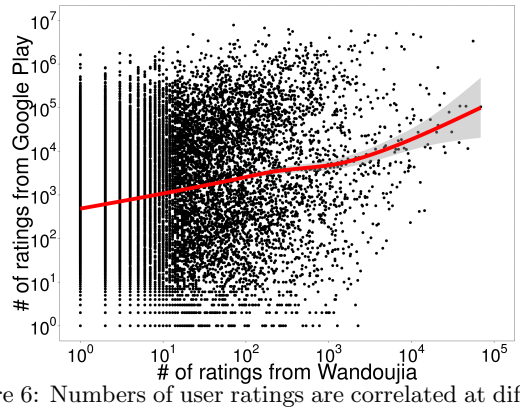

Figure 6: Numbers of user ratings are correlated at different marketplaces, but significant biases exist.

We then investigate whether the same app receives consistent ratings at the two marketplaces. We compute the average ratings of apps in both marketplaces and correlate them in a scatter plot. The average rating on Google Play is denoted as the *score* and the average rating on Wandoujia is denoted as the *likerate*, computed as follows. These two notations will be used through the rest of the paper.

$$\text{score} = \frac{\sum stars}{\text{number of ratings}} \quad (1)$$

$$\text{likerate} = \frac{\text{number of } likes}{\text{number of } likes + \text{number of } dislikes} \quad (2)$$

Figure 7 presents a positive correlation between the *scores* on Google Play and the *likerates* on Wandoujia, for apps with at least five ratings on Wandoujia (7,513 in total). This indicates that user ratings are overall consistent at the two marketplaces. However, one can also easily identify many different or even contradictory ratings. The Pearson's coefficient r of this correlation is 0.35, which becomes much smaller if we include apps with fewer ratings (0.30 for apps with at least 3 ratings, and 0.19 for all rated apps).

The comparison of user ratings at two marketplaces indicates that user ratings may be a trustful measurement of the quality of apps that have received many ratings, but it suffers from severe problems of sparseness and biases for apps that are rarely rated. Other types of review biases have been discussed in literature. For example, Asian users are reported to be less likely to provide their ratings [25]. There has been a debate whether users who like or dislike an app are more likely to leave a review. There is also a prevalence of fake reviews [30] or review spams [20], as developers can manipulate the ratings by hiring the crowd to rate their apps (known as *water army* in China). These problems have limited the effectiveness of users ratings as the solo indicators of app quality, especially for those with fewer ratings. For

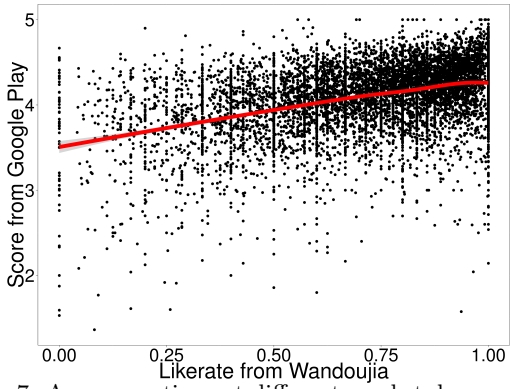

Figure 7: Average ratings at different marketplaces are correlated if there are abundant of ratings. Showing only apps with at least 5 ratings on Wandoujia.

these apps, it is critical to find alternative signals that better indicate their quality.

4.2 Activity Indicators

Most app marketplaces have recognized the limitation of user ratings. When the number of ratings is small, many marketplaces (e.g., Google Play) also report the total number of times an app is downloaded/installed, with the assumption that more downloads indicate a better quality of the app. Some variants are also adopted, such as the number of users/devices that have downloaded the app (e.g., Wandoujia), or the number of most recent activities (e.g., Apple App Store). These metrics are generally extracted from the actual user activities instead of online reviews.

4.2.1 *Popularity Indicators*

While how the users use an app intuitively imply how they like the app, can simple metrics such as the number of downloads reflect this preference? To verify this, we correlate the number of downloading activities to the likerate of an app, assuming that the likerate is a more objective measure of app quality when the ratings are abundant. Motivated by the findings in Figure 7, we rank all apps that have received at least 5 ratings based on how many times they are downloaded in the five months, split them into equal sized bins, and plot the means and the standard deviations of each bin. As the number of downloads follows a power law distribution, we plot it at the *log* scale (X-axis). A few observations can be made from Figure 8.

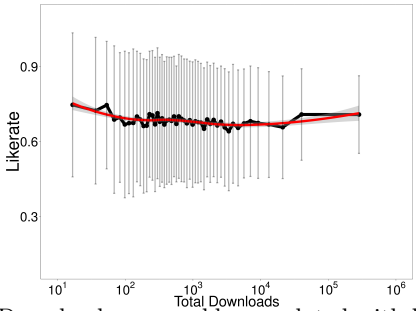

Figure 8: Downloads are weakly correlated with likerate for popular apps and negatively correlated for unpopular apps.

When the number of downloads is small (e.g., less than 1,000), it is negatively correlated with the average likerate. In other words, the more times an app is downloaded, the

more likely it is *disliked* by users. This is rather counter-intuitive. There might be several reasons. Apps which are not frequently downloaded may be sensitive to fake "*like*" ratings, either artificially boosted by paid-off reviewers or maliciously rated down by their competitors. When the number of downloads exceeds 10,000, the correlation becomes positive, but still quite weak. In either case, this indicates that the number of downloads alone is at best a weak indicator of app quality, which may even be invalid for unpopular and new apps.

Would the number of users/devices be more reliable? Unfortunately no. From Figure 9, we see that the number of users (devices) installing an app is even negatively correlated with its likerate, although the correlation is quite weak.

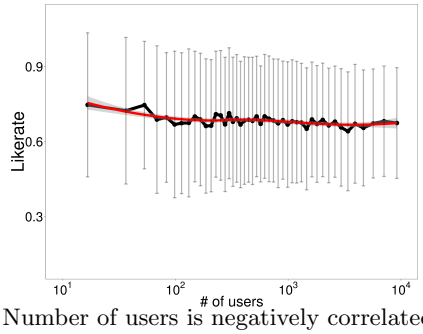

Figure 9: Number of users is negatively correlated with likerate.

Although marketplaces have explored simple activity indicators such as the number of downloads or the number of users to complement user ratings of apps, unfortunately they are at best weakly correlated with app quality. This sounds discouraging, but can we find better indicators from app management activities?

4.2.2 *Sequential Indicators*

An important understanding about user activities is that they are not independent but always appear as sequences of events. Indeed, when search engines utilize users activities as an implicit feedback about the relevance of documents, the sequences of actions are usually more indicative than single clicks. For example, Joachims et al. found that the sequence of actions where a user skips a higher ranked search result but clicks on one ranked below emits a signal that the user finds the lower ranked document more relevant than the higher ranked one [21]. Analogically, we anticipate that some sequential patterns of the app management activities may be better indicators of app quality than downloading actions alone. For example, we find that the sequence of "Uninstallation-Downloading (UD)" activities, i.e., a user uninstalls an app and later on re-installs it, may be a good indicator of the user liking the app (that's why he installed it back). To verify this, we plot the proportion of these "UD" patterns among all users of an app and correlate it with the likerate of the app. Since not all apps have an "UD" activity sequence, we only plot those who have.

Interestingly, the average number of "UD" sequences per user is in general positively correlated with the likerate of the app (see Figure 10). This is promising, indicating that some user activity sequences may be good indicators of user preferences and app quality. Unfortunately, the "UD" sequences are relatively rare. Among 30,614,327 user activity sequences that have at least two actions, only 4 percent contain a subsequence of "UD." If a marketplace uses this metric directly, it may yield an even smaller coverage than the online ratings. Are there other activity patterns that are both indicative and rich? Although generally positive, the correlation in Figure 10 is still quite weak. How much better can sequential indicators from app management activities assess app quality? We are motivated to find all such sequential indicators and combine them in a systematic way.

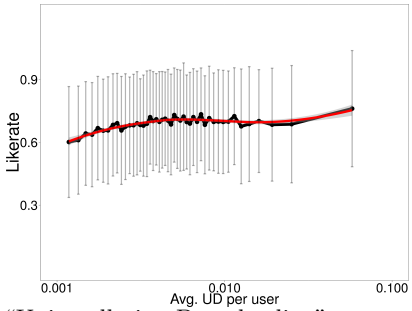

Figure 10: "Uninstallation-Downloading" sequences are positively correlated with Likerate.

5. A COMPARATIVE ANALYSIS

The correlation between the "UD" sequence and the likerate seems promising, which inspires us to find all sequential indicators like those and study them in a systematic way. We focus on answering an important question: how much better can these indicators measure user preferences and app quality comparing to the existing measures used by the app marketplaces. The answer to this question may motivate the marketplaces to adopt these indicators and improve their current systems. To find this answer, we need a formal task and evaluation metric(s) with which any combination of activity indicators can be fairly compared with the existing indicators currently used by the app marketplaces.

5.1 The Ranking Task

To help users find and explore high quality apps, most marketplaces provide rankings of apps, based on either user ratings or other indicators. Ranking by the number of downloads or the number of users is particularly useful when the ratings are rare. To fairly compare new and existing indicators, we adopt this practical scenario and rank the same set of apps based on one indicator or a set of indicators combined through an algorithm. A method is better if it ranks high quality apps higher and low quality apps lower.

Obtaining the ground-truth of app quality is difficult, if there exists one at all. In this analysis, we use the ratings of apps as a surrogate of app quality. In particular, we use the likerate of an app as the gold standard of its quality, as defined in Equation 2. Although ratings suffer from sparseness and biases, they are reasonable when there are many of them and we have to rely on them when nothing better is available. Readers could reproduce the results when a more reliable gold standard becomes available.

To make sure the likerates are reliable, we only look at apps with sufficient ratings. How many is sufficient? As in Section 4.2, we could use a threshold cutoff such as 5 ratings. In this analysis we use a more rigorous definition, which will be described below. Note although in the evaluation we only look at apps with sufficient ratings, the activity indicators assessed to be effective can be easily applied to unrated or barely rated apps and provide a meaningful ranking of those.

5.2 Evaluation Metrics

KENDALL'S TAU. When using ratings as the gold standard, the goodness of a ranking method (either by a single indicator or by a combination of indicators) can be measured by how much the ranking of apps it produces agrees with the ranking of apps by the likerates. Statistically, this agreement can be measured by *Kendall's Tau*, which is widely used to compare two ranking lists [27]. It measures the difference between the number of concordant pairs in two ranked lists and the number of discordant pairs, normalized by the total number of ordered pairs:

$$\tau = \frac{\sum_{i,j}(\mathbb{I}[(x_i - x_j)\cdot(y_i - y_j) > 0] - \mathbb{I}[(x_i - x_j)\cdot(y_i - y_j) < 0])}{\sum_{i,j}(\mathbb{I}[(x_i - x_j)\cdot(y_i - y_j) > 0] + \mathbb{I}[(x_i - x_j)\cdot(y_i - y_j) < 0])}, \quad (3)$$

where i, j are two apps, x_i, y_i are the orders of the two apps in two ranking lists, and $\mathbb{I}[\cdot]$ is an indicator function which values 1 if the expression is true and 0 otherwise. The value of τ approaches 1 if the two rankings match perfectly, -1 if they are completely opposite, and 0 if they are independent.

Although Kendall's Tau has a solid statistical foundation, there are two limitations of applying it to practice. It takes the whole list into consideration where top-ranked items contribute equally to items ranked at the bottom. In reality, the users normally browse apps from the top, and therefore having high-quality apps ranked to the top is more important than having low-quality apps ranked at the bottom. Moreover, like other correlation scores, τ is hard to be interpreted intuitively, which is a critical issue for the marketplaces.

MEAN AVERAGE PRECISION. To complement Kendall's Tau, we introduce another evaluation metric, the *mean average precision (MAP)*, which is widely used in information retrieval literature to measure the relevance of a ranked list of documents [27]. The advantage of MAP is that it puts more weight on the top-ranked items and as a "precision" its value is somewhat interpretable. However, MAP takes a binary value (good or bad) as the ground-truth of every item (unlike Kendall's Tau which takes a ranking of items as the truth). In our data, such a binary decision is not directly available, which must be transformed from the likerates.

A straightforward way of doing this is to treat all likerates that are greater than 0.5 as "good" and those less than 0.5 as "bad." This is equivalent to summarizing a voting system with majority votes, a commonly adopted strategy in election and crowd-sourcing [14]. When the votes are rare, this simple strategy becomes problematic. Even when an app has abundant ratings, one single rating could have flipped the sign of the gold standard. We therefore filter the apps using a statistical test, and only trust those with a likerate significantly higher or lower than 0.5 (there are significantly more likes or dislikes in the ratings). We used a two-tail proportion test with confidence level 0.05. Apps that pass this test would have many ratings and a considerable difference between positive and negative ratings. Specifically, we sample 7,016 apps from the set of rated apps, the rest of which will be used for training purposes that will be described later. After filtering with the proportion test, there are 1,423 apps left in the labeled set. For fair comparison, all ranking methods will be evaluated and all evaluation metrics will be computed using this set of 1,423 apps.

5.3 Baseline: Ranking by Popularity

As described, most marketplaces are already using popularity indicators, such as the number of downloads or the number of users to rank and recommend apps. How effective are these indicators?

We use Kendall's Tau to measure the effectiveness of the rankings of apps by the two popularity indicators. Ranking by the number of users yields a τ of -0.2436 and ranking by the number of downloads yields a τ of -0.2372 on the test set of 1,423 apps. This is disappointing, showing that the current indicators used by the marketplaces do not accurately reflect the app quality, even negatively correlated. It is surprising though, as the perceived quality of popularity ranking does not seem that bad. One possibility is that although the complete ranking is ineffective, it is probably doing a good job on top ranked apps (and the users always browse from the top). Indeed, the MAP scores of ranking using the number of users and the number of downloads are 0.8574 and 0.8629, which look reasonable.

This contradiction between two metrics is actually meaningful: apps with many good ratings are more likely to be used and therefore rise to the top of the popularity list - yet another process of "rich-get-richer." A reasonable MAP and a miserable τ presents good news to the popular apps and bad news to the long tail and new apps.

Having the understanding of the baselines, we are eager to know how much the sequential indicators from app management activities could improve the ranking of apps. Will they improve the ranking overall, thus providing an opportunity to less rated apps? If so, will the improvement be at an expense of compromising the top-ranked apps? Can multiple indicators be combined as a joint force? Below we present a comprehensive evaluation of the predictive power of activity indicators.

5.4 Predictive Power of Activity Indicators

We believe there are multiple patterns of app management activities that are good indicators of user preferences and app quality, in addition to the "UD" pattern studied in Section 4. Although one can test the effectiveness of such patterns one by one, a more interesting question is whether the combination of these patterns can much better rank the apps. There could be many ways to combine multiple indicators. We cast this as a machine learning problem. In other words, we treat each potential activity indicator as a feature and learn a model from data that combines these features and makes prediction of the quality of an app.

5.4.1 Experiment Setup

Specifically, given any set of activity indicators (features), we train a regression model that predicts the likerate of an app. We explore several standard, state-of-the-art machine learning algorithms for this purpose, including the Ridge Regression (Ridge) [19], the Lasso Regression (Lasso) [38], the Random Forrest Regression (RF) [11], and the Gradient Boosted Regression Tree (GBRT) [17]. These algorithms provide a representative coverage of methods that combine features linearly and non-linearly. [§]

To evaluate the predictors, we randomly select 50,000 apps for training and hold out the rest 7,016 apps as test set (the same split in Section 5.2 and 1,423 after filtering using the proportion test). The hyper-parameters of every algorithm is selected using a 5-fold cross validation on the training set (by optimizing either Kendall's Tau or MAP), including the regularization parameter for Ridge and Lasso and the number of trees and number of nodes for RF and GBRT.

[§]We use the glmnet package [16] in R for Lasso and Ridge, and scikit-learn package in Python [31] for RF and GBRT.

We then train a model with the best-performing parameters on the entire training set and evaluate it on the test set.

5.4.2 Feature Extraction

We start with a set of simple indicators and then move to two other sets that make use of the sequential patterns and the time interval information. In the rest of this paper, we will use D to represent *Download*, P for *Update* and U for *Uninstallation* actions.

Unit features.

We first extract a set of features for every app that includes the number of devices (denoted as #Dev), the average number of actions per device (denoted as Avg.Act), and the number of unit activities (D|P|U) per device. These simple features can be easily adopted by the production systems of marketplaces. Note that a subset of the features are already considered by the marketplaces (i.e., #Dev and D).

Table 2: Performance of unit features (5 features). Parameter selected using either Tau or MAP. Best performance under two metrics are highlighted.

Model	Metrics	Parameter Selection	
		Tau	**MAP**
Ridge	Tau	0.0440	0.0440
	MAP	0.9271	0.9271
Lasso	Tau	0.0440	0.0440
	MAP	0.9271	0.9271
RF	Tau	0.0571	0.0572
	MAP	0.9294	0.9298
GBRT	Tau	**0.0923**	**0.0834**
	MAP	**0.9333**	**0.9309**

The performance of models with unit features is presented in Table 2. Comparing to the ranking methods currently used by the marketplaces, even just adding other unit features will largely improve the performance. The improvements of MAP from 0.8574, 0.8629 to 0.9333 and Tau from -0.2436, -0.2372 to 0.0923 over the two baselines are statistically significant with p-value $\ll 0.01$. Significance level of MAP improvement is tested with two-side paired t-test and Tau improvement with randomization test [37]. The same tests are used for the rest of the paper. This improvement not only appears in the entire ranking list (Kendall's Tau becomes positive), but also appear among the top-ranked apps (a significant improvement of MAP). This is promising, which indicates that a combination of activity indicators better predicts the quality of both popular apps and the long tail. Among the four algorithms, GBRT achieves better results under both metrics, and tree-based methods outperform the linear regressions overall.

Sequential features.

The success of unit features motivates us to further explore more complicated activity indicators, patterns like the "Uninstallation-Downloading (UD)" studied in Section 4. Comparing to unit indicators that treat management activities independent to each other, the order between consecutive actions are likely to be a good indicator of user preference. For example, the behavior of downloading an app and then uninstalling it sends an opposite message to the behavior of uninstalling an app and then downloading it again. In general, if we consider the actions of every user managing every app as an ordered sequence, then such ordered activities appear to be a subsequence of length two.

To systematically study these sequential indicators, for every user of every app that have at least two actions, we construct a sequence of five possible symbols: three actions "Downloading (D)," "Updating (P)" and "Uninstallation," and two auxiliary symbols "Start (S)" and "End (E)" which indicates the start and end of a sequence. For example, a sequence "SDPPUDE" indicates that the user first downloaded the app, updated it twice, uninstalled it and then decided to re-install it, with no further actions recorded. The unit indicators (D|P|U) can now be seen as the subsequences of length one (or unigrams) and the "UD" and "DU" patterns can be seen as bigrams. Given any n, we can enumerate all subsequences with n or fewer *consecutive* actions. The average number of these ngrams per device, together with #Dev and Avg.Act, constitute the set of "sequence-n" features. Notice that some sequential features like the "UP" (uninstallation followed by updating) might sound confusing as in reality one cannot update an app without downloading it first. This is because only activities that are conducted through the Wandoujia app are logged, and the user may have downloaded the app from other sources. Also, a sequence does not necessarily begin with an "D," as the app may be preloaded or installed before the five-month period. We vary the n from 1 to 5 in our experiments, as a larger n may make some features very sparse and lead to over-fitting.

The performance of sequential features is presented in Table 3. Note that sequence-(n-1) features are always a subset of sequence-n features. Most of the best results are observed when length-3 or length-4 features are used. When the length of subsequences is greater than 4, the performance begins to drop. Comparing to unit features, the best performing sequential features increases Tau from 0.0923 to 0.1180 (p-value = 0.16) and MAP from 0.9333 to 0.9423 (p-value $\ll 0.01$).

With an investigation of the importance of individual features (e.g., the coefficients in Ridge), we notice that there are quite a few incomplete patterns such as "UU" and "UP." Some actions must have been omitted in these patterns simply because they were not captured by the Wandoujia app. App management activities only represent a small subset of activities that a user does with an app. Between consecutive "D|P|U" activities there may be other actions (such as clicking, browsing, and grouping) that are not logged but quite indicative. This suggests that the predictive power of the activity patterns may still be underestimated. Without observing these activities, we are not able to verify the hypothesis. Nevertheless, the time interval between two consecutive app management activities may be informative, even if we do not know what actually happened during the interval.

Time intervals.

The length of time intervals between consecutive activities can be indicative. Intuitively, an uninstallation immediately following a download indicates a stronger negative preference than an uninstallation that happens weeks or months after downloading. Incorporating time intervals into the sequential activity patterns is not easy, as the length of the intervals is continuous. However, recall from Figure 4 that the intervals between consecutive activities follow a cyclic pattern in every 24 hours and an exponential trend overall. We therefore decide to discretize the intervals into days. That is, we insert a time symbol T into the activity sequences, each of its appearance indicates a complete day (24 hours) between two consecutive activities. Two consecutive activities within a day will be inserted an "-" in the new sequence.

Table 3: Performance of sequential features (20, 68, 212, 644 features, respectively) with model parameters selected using Tau and MAP (columns labeled). For every algorithm and every metric, best performing sequence-n features are highlighted.

Model	Metric	Sequence-2		Sequence-3		Sequence-4		Sequence-5	
		Tau	Map	Tau	Map	Tau	Map	Tau	Map
Ridge	Tau	0.0744	0.0744	0.0779	0.0779	**0.0801**	0.0793	0.0772	0.0757
	MAP	0.9211	0.9211	0.9229	0.9229	0.9239	**0.9240**	0.9217	0.9234
Lasso	Tau	0.0996	0.0744	**0.1027**	0.0779	0.1027	0.0801	0.1027	0.0775
	MAP	0.9048	0.9211	0.9032	0.9229	0.9032	**0.9239**	0.9032	0.9200
RF	Tau	0.0667	0.0679	0.0694	**0.0712**	0.0708	0.0690	0.0695	0.0695
	MAP	0.9325	0.9343	0.9364	**0.9376**	0.9368	0.9363	0.9365	0.9365
GBRT	Tau	0.1173	0.1154	**0.1180**	0.1099	0.1090	0.1090	0.1072	0.1072
	MAP	**0.9423**	0.9419	0.9406	0.9407	0.9392	0.9392	0.9400	0.9400

Table 4: Performance of T-Sequential features. Time intervals are inserted into sequence-3 features: 824 features before feature selection and 153 features after.

Methods	Parameter Tuning	Tau		MAP	
	Feature Selection	No	Yes	No	Yes
Ridge	Tau	0.0943	0.1033	0.0936	0.1032
	MAP	0.9293	0.9346	0.9299	0.9345
Lasso	Tau	**0.1716**	0.1716	0.0952	0.1033
	MAP	0.9209	0.9209	0.9278	0.9346
RF	Tau	0.0804	0.0749	0.0804	0.0744
	MAP	0.9439	0.9429	0.9439	0.9429
GBRT	Tau	0.1399	0.1356	**0.1399**	0.1356
	MAP	**0.9512**	0.9499	**0.9512**	0.9499

We also consider an interval longer than 4 days to be long enough and substitute 4 or more consecutive Ts with a "*" symbol. An activity sequence therefore becomes something like "SDTTP*P-UTDE." We then fill in the sequential features with time intervals. For example, the feature "DPU" in previous feature set is extended into "DTP-U," "DTTP-U," "D*PTTU," and so on. These extended sequential features, together with #Dev and Avg.Act, constitute the set of time-interval inserted sequential (T-sequential) features.

The consideration of time-intervals has significantly increased the number of features, exposing the models to potential threat of overfitting. We consider a feature selection strategy that filters out features not significantly correlated with the likerates. Specifically, we compute the Pearson's correlation between every feature and the likerates of apps based on the training data, and only keep those with a p-value smaller or equal to 0.05. The performance of T-sequential features are presented in Table 4, with and without feature selection. We see that by inserting time intervals into sequential features, the best result MAP increased from 0.9419 to 0.9512 (p-value $\ll 0.01$) and the best result of Tau increased from 0.1180 to 0.1716 (p-value $\ll 0.01$), surprisingly achieved by Lasso. Feature selection in general has improved the linear methods but does not improve the tree-based methods, suggesting that the tree-based methods are more robust to over-fitting.

Putting all together.

The last exploration we do is to put all features together and test whether it further improves the ranking performance. As unit features are subset of sequential features, we essentially combine sequence-3 features and their time interval inserted extensions. The same feature selection is used to reduce the number of features. The results are shown in Table 5. We see that GBRT still achieves the highest MAP, bringing the metric to above 0.95. This MAP improvement over the best result of T-sequential features is significant with p-value $\ll 0.01$. The best result of Tau does not improve.

Table 5: Performance of combination of T-Sequential features and sequential features: 890 features before feature selection and 207 features after.

Model	Metric	Parameter Tuning Tau	Parameter Tuning MAP
Ridge	Tau	0.1032	0.1031
	MAP	0.9346	0.9346
Lasso	Tau	**0.1716**	0.1023
	MAP	0.9209	0.9343
RF	Tau	0.0770	0.0768
	MAP	0.9433	0.9431
GBRT	Tau	0.1395	**0.1405**
	MAP	**0.9542**	**0.9523**

5.5 Discussion

Overall, combining multiple indicators extracted from app management activities has significantly outperformed the ranking methods used by the marketplaces, increasing Kendall's Tau from -0.24 to 0.17 and the mean average precision from 0.86 to 0.95. The result is encouraging, showing that these activity indicators improve the accuracy of both the complete ranking of apps (good news to unpopular and new apps) and the top ranked apps (good news to popular apps). We believe there is room to further improve the prediction performance, but decide to leave it as future work. Indeed, during the experiments we observed some rather interesting behaviors of the predictors, which may provide insights for the marketplaces on how to better rank the apps and how to explain the ranking.

5.5.1 Feature Analysis

One surprising observation is that the highest Tau score (0.1716) is achieved by Lasso (when the hyper-parameter is tuned using Tau). In most other cases, the performance of Lasso is inferior to GBRT and RF. We found that with Lasso, only one feature has a non-zero coefficient, which is the pattern "SD-U" from the T-Sequential feature set. This means the first action we observed from a user is download-

ing the app, and then he uninstalled the app within 24 hours. Intuitively, this suggests that the user downloaded the app, tried a few times, found it disappointing or even annoying, and then uninstalled it without hesitating. The time interval is quite sensitive. Lasso also identified the corresponding "SDU" from the sequential patterns, but its predictive power is much weaker (0.10 vs. 0.17). The coefficient of the feature is negative, meaning that the smaller fraction of users who did this, the higher the quality of the app. This is rather interesting, suggesting that if we care about the accuracy of the orders between all apps, especially the apps that are not ranked to the top (unpopular/new apps or low quality apps), one indicator is better than many. A marketplace could immediately adopt this finding in their system and provide a ranking of apps in the reversed order of this indicator, with an easy explanation of the top-ranked apps: "everyone who installs it keeps it." We anticipate this would be particularly useful for promoting new and high-quality apps.

It is much harder if our goal is to optimize the top-ranked apps. To achieve a high MAP, every method has utilized a combination of many features. We investigate the best performer, achieved by GBRT with both the T-sequential features and the sequence-n features. In particular, we look at the importance of each feature in the GBRT model and the Pearson's r correlation between the feature and likerate. We observe that many of the important features are variants of "DU," such as "SD-U", "D-UE", "D-U", and they all have negative correlations with likerate ($r < -0.18$). Interestingly, "D*UE" is also among the most important features, but with a positively correlated with likerate ($r = 0.02$). Notice that the "D*UE" requires a long interval ($\geq$ 72 hours), so the users must have kept the app for some time. Other important features include the variants of "DD" and the variants of "UU." The "UD" indicator we presented in Section 4 is ranked at the 48th by the importance in the best performing GBRT, among the 207 features in total.

5.5.2 Error Analysis

Both metrics show an overall agreement between the predicted result and the observed likerate. There are still many mistakes made in the predictions. Admittedly, all algorithms have their limits. Yet we are curious about whether the "gold standard" is well grounded. That is, do the observed likerates reflect the the true preference of the users? Table 6 shows the details about the most over-predicted and under-predicted apps, apps with a predicted likerate much higher or lower than observed. Compared to the statistics of all test apps, both over-predicted and under-predicted apps have less reviews and less users, implying the curse of data sparseness. Among the most under-predicted apps, there are generally more ratings than users. Although the popularity of apps might rise and fall, and the 5-month data might not include all users, the unexpected high ratio of ratings over users still make us suspicious of rating manipulations.

Table 6: Summary statistics of the 100 most over/under-predicted apps, compared with all 1,423 in the test set

	All	over-pred	under-pred
Avg. #users	2379.18	357.33	26.25
Avg. #reviews	366.64	67.09	122.28
% of Games	36.61%	44.00%	21.00% ***
% of Tools	8.22%	8.00%	15.00% ***

Among the 1,423 apps in the test set. 37% of them are games, constituting the largest category. However, the ratio of games is only 21% among the 100 most under-predicted apps, which differs significantly from the overall ratio of 37% (p-value = 0.002, χ^2 test). Games consist a larger proportion among over-predicted apps (44%), although not statistically significant. One possible explanation is that users might have different expectations and attitudes to different categories of apps. Another possibility is that game users are less likely to leave positive reviews and more likely to leave negative reviews, probably due to the difference of user demographics. Similarly, we see the proportion of Tools is also higher in the under-predicted apps (p-value = 0.03, χ^2 test). In either case, this suggests biases in online ratings, which limit their effectiveness as the gold standard of app quality. This also suggests that incorporating additional information (e.g. app category) to user behaviors may further improve the performance of prediction. In this paper, we choose not to use information other than app management activities as the goal is to fairly measure the effectiveness of the activity indicators. Additional information such as app profiles, user demographics, and textual reviews may be explored in the future task to optimize the prediction of app ratings.

6. CONCLUSION AND FUTURE WORK

In this paper, we present an empirical analysis of a very large collection of app managing activities of smartphone users. The data is collected through a leading Android marketplace in China. We show that users download, update, and uninstall apps differently when they like or dislike the apps. These app management activities indicate the users "**vote with their feet**," which effectively supplement the biases and sparsity of online ratings of apps. We identify behavioral patterns that serve as indicators of the user's preference on apps, which can be integrated by machine learning algorithms that predict the ratio of positive ratings of the apps. With these activity indicators, the prediction can be improved significantly.

Some surprising findings from our analysis may be directly useful for app marketplaces or app developers. For example, we demonstrated that the number of downloads of an app is not a good indicator of the users' preference or the quality of the app. We also notice that users have a routine schedule to manage the apps on their mobile devices. A single pattern of the activities provides a general ranking of apps that is surprisingly accurate, which may be used to effectively promote new and high quality apps. Multiple time-aware sequential patterns can be combined with a machine learning algorithm and significantly improve the accuracy of top-ranked apps. Since Wandoujia is a marketplace most popular in China, it is interesting to explore the uniquely local characteristics of Wandoujia and compare them with other marketplaces such as Google Play. It is also a natural extension of this work to correlate the app management activities with the frequency on how users use the apps, or with the narrative reviews of apps. We plan to integrate the knowledge derived from the management behaviors into the recommender systems of apps.

Acknowledgment

This work was supported by the High-Tech Research and Development Program of China under Grant No.2015AA01A203, the Natural Science Foundation of China (Grant No. 61370020, 61421091, 61222203, 61528201). Qiaozhu Mei's work was supported in part by the National Science Foundation under grant No. IIS-1054199.

7. REFERENCES

[1] F-droid. http://f-droid.org/.

[2] Fetch. http://www.androidtapp.com/fetch/.

[3] Flurry. http://www.flurry.com/.

[4] Number of available apps on apple applestore. http://ipod.about.com/od/iphonesoftwareterms/qt/apps-in-app-store.htm.

[5] Number of available apps on google play. http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/.

[6] Preemptive. http://www.preemptive.com/.

[7] S. Agarwal, R. Mahajan, A. Zheng, and V. Bahl. Diagnosing mobile applications in the wild. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, page 22. ACM, 2010.

[8] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[9] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer. Falling asleep with angry birds, facebook and kindle: a large scale study on mobile application usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 47–56. ACM, 2011.

[10] M. Böhmer and A. Krüger. A study on icon arrangement by smartphone users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2137–2146. ACM, 2013.

[11] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[12] N. Chen, J. Lin, S. C. Hoi, X. Xiao, and B. Zhang. Ar-miner: mining informative reviews for developers from mobile app marketplace. In *ICSE*, pages 767–778, 2014.

[13] G. Chittaranjan, J. Blom, and D. Gatica-Perez. Mining large-scale smartphone data for personality studies. *Personal and Ubiquitous Computing*, 17(3):433–450, 2013.

[14] D. Easley and J. Kleinberg. *Network, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[15] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin. A first look at traffic on smartphones. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 281–287. ACM, 2010.

[16] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[17] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38:367–378, 2002.

[18] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh. Why people hate your app: Making sense of user feedback in a mobile app store. In *Proceedings of KDD '13*, pages 1276–1284. ACM, 2013.

[19] A. E. Hoerl and R. W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, Feb. 1970.

[20] N. Jindal and B. Liu. Opinion spam and analysis. In *Proceedings of WSDM '08*, pages 219–230. ACM, 2008.

[21] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of SIGIR '05*, pages 154–161. ACM, 2005.

[22] J. Jung, S. Han, and D. Wetherall. Short paper: enhancing mobile application permissions with runtime feedback and constraints. In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, pages 45–50. ACM, 2012.

[23] H. Li, X. Liu, W. Ai, Q. Mei, and F. Feng. A descriptive analysis of a large-scale collection of app management activities. In *Proceedings of WWW' 15 Companion Volume*, pages 61–62, 2015.

[24] 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 2015 ACM Conference on Internet Measurement Conference*, pages 459–472, 2015.

[25] S. Lim, P. Bentley, N. Kanakam, F. Ishikawa, and S. Honiden. Investigating country differences in mobile app user behavior and challenges for software engineering. *IEEE Transactions on Software Engineering*, 2014.

[26] B. Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, May 2012.

[27] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[28] M. E. Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[29] M. Ott, C. Cardie, and J. Hancock. Estimating the prevalence of deception in online review communities. page 201. ACM Press, 2012.

[30] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319, 2011.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[32] A. Rahmati, C. Tossell, C. Shepard, P. Kortum, and L. Zhong. Exploring iphone usage: the influence of socioeconomic differences on smartphone adoption, usage and usability. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pages 11–20. ACM, 2012.

[33] A. Rahmati and L. Zhong. Studying smartphone usage: Lessons from a four-month field study. *Mobile Computing, IEEE Transactions on*, 12(7):1417–1427, 2013.

[34] L. Ravindranath, J. Padhye, S. Agarwal, R. Mahajan, I. Obermiller, and S. Shayandeh. Appinsight: Mobile app performance monitoring in the wild. In *OSDI*, pages 107–120, 2012.

[35] A. A. Sani, Z. Tan, P. Washington, M. Chen, S. Agarwal, L. Zhong, and M. Zhang. The wireless data drain of users, apps, & platforms. *ACM SIGMOBILE Mobile Computing and Communications Review*, 17(4):15–28, 2013.

[36] K. Shi and K. Ali. Getjar mobile application recommendations with very sparse datasets. In *Proceedings of KDD '12*, pages 204–212, 2012.

[37] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *the sixteenth ACM conference*, page 623. ACM Press, 2007.

[38] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, Jan. 1996.

[39] C. Tossell, P. Kortum, A. Rahmati, C. Shepard, and L. Zhong. Characterizing web use on smartphones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2769–2778. ACM, 2012.