Explicit and Implicit Ratings for Mobile Applications

Andrea Girardello, Florian Michahelles

Information Management - ETH Zurich
{agirardello, fmichahelles}@ethz.ch

Abstract: Today most mobile operating systems provide an application portal (e.g. Android Market, AppStore) where users can search by keywords and explicitly rate applications published by third-party developers. In this paper we go beyond this approach and introduce an implicit rating mechanism for Android programs. Our approach, captures installation, update, and removal events, and allows to show them among users. Based on these measurements we calculate implicit ratings. As a result we compare these ratings with explicit ratings from the Android Market. For applications with less than 5,000 downloads implicit ratings provide more information than users explicitly enter on Android Market.

1 Introduction

Today, mobile phones are everyday companions reaching the attention of research, industry and consumers. The constant improvement of the hardware related to these devices has enhanced their capabilities, thus letting them run a huge variety of applications, which counts only on the Android Market roughly 47,000 programs. In order to distribute their works, developers publish applications on these centralized portals from which the consumer can directly search, download and then install the application onto her mobile device [HO09]. At present, these application portals let users rate applications by giving a rating on a one to five stars scale. Despite this being well understood and accurate, it interrupts the usual behavior of a user [Cl01]. Additionally, we suppose that users are more likely to review applications if they perceive them as either very good or very bad, as already found out for movies [DN06].

To overcome this limitation we let users implicitly rate applications and thus define their acceptance. Our approach, AppAware, assists users to find interesting applications with the help of other users, thus allowing individuals to make fortunate discoveries of fresh applications by accident – similar to sharing of bookmarks, e.g. Delicious[1]. To achieve this, an AppAware client running on an Android[2] device automatically shares online installations, updates and removals of applications. In this way a user becomes aware (App-Aware) of what other people are installing on their Android phones right now or in her proximity, learning from them [GM10]. Every time a user installs, updates or removes an application, these events will contribute to the implicit rating algorithm we designed.

In the following section, we examine the related work in the field of mobile application portals. This is used to deduce some concept principles that we describe in section 3. In section 4 we follow up with the evaluation for then summarizing the AppAware implicit ratings idea in section 5.

2 Related work

In this section, we briefly review the state of the art and related work that have informed our concept and indicate how AppAware differs from these.

At present, the official Android application portal can be accessed only from the Market mobile application and, in a limited way, from the related website. Here, mobile applications are divided into categories (i.e. Communication, Entertainment, Shopping, etc.) and for each application the user can look at its details, namely: number of downloads, average ratings and a list of recent comments form users. Moreover, users who have installed a certain application can rate it on a one to five stars scale and, eventually, provide a short review. AppAware does not aim at replacing the Android Market or providing a proxy, it is rather a companion to plan users' serendipity [Ea04] in applications finding and provide an alternative method for mobile application rating.

To overcome some limitations imposed by Google (i.e. full access to the Market only from an Android device), many third-party developers are launching new services to access applications' details from a personal computer. Good examples are AndroLib[3] and AppBrain[4]. The major difference between the two is that AppBrain provides a user with an applications shopping cart that can be synced with the device through an Android client application. However, the idea is not innovative since it is trying to port the concept of Apple's iTunes to the Android world. The mentioned websites provide statistics for applications, however they do not provide any other mechanism to let users rate Android programs and simply show the data from the official Android Market.

[1] http://delicious.com
[2] The Android platform has been chosen since its API allows to capture applications' events (i.e. installations, removals and updates).
[3] http://www.androlib.com
[4] http://www.appbrain.com

Another related work is Appazaar [BPB10], a recommender system for mobile applications developed at the Lab for Software Engineering at Münster University of Applied Sciences. Based on a user current and historical locations and applications usage, Appazaar recommends applications that might be of interest for her. Therefore, Appazaar applies different algorithms from the research field of context awareness to analyze all the input data and create profiles of different situations. Despite providing apps recommendations is an appealing feature, AppAware focuses towards an implicit rating approach which is then also used to suggest a list of applications that users can try.

3 Concept

AppAware is a mobile application that captures and shares installations, updates, and removals of Android programs in real time.

For each Android application a web page shows its description, the list of recent users' events (installations, updates or removals) and a meter representing its acceptance by the AppAware community (Figure 1b). The core idea behind this meter is that it takes installations, updates and removals of applications as input for the computation. When the gauge points toward the green range the acceptance is excellent, yellow range for good acceptance and red range if almost no AppAware user is keeping the application installed. This continuous stream of application events (installations/removals/updates, see Figure 1a) provides the basis for serendipity for other users [GM10].

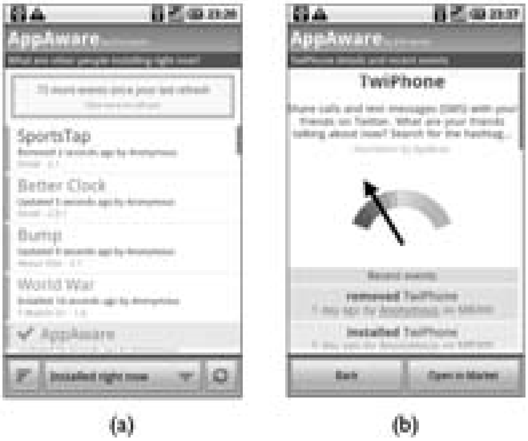

Figure 1. Real-time stream of installed, updated and removed applications (a) and an application's page with its average implicit rating represented by a meter (b).

Besides this new way to interact with an application portal, AppAware introduces an implicit rating mechanism for Android programs. Every time a user installs, updates or removes an application, these events will contribute to the rating algorithm we designed. The assumption behind this approach is that excellent/good applications are not removed once installed, whereas applications not liked tend to be removed from the device. To model what described so far, AppAware defines the acceptance rate *v* for an application *app* as the value going from 0 to 100 computed with the formula in (1), where *U* is the set of users having at least one event for *app*.

QuickTime™ and a
decompressor
are needed to see this picture. (1)

QuickTime™ and a
decompressor
are needed to see this picture. (2)

As (2) defines, while computing the acceptance rate we consider for each user her most recent event for a certain application *app*. In this process, an update is considered the highest valuable event, even more important than an installation. The belief is that an application's update brings to light the user's uninterrupted interest in that piece of software and, at the same time, the developers' effort in keeping their application up to date.

4 Evaluation

In this section we compare the implicit ratings generated by AppAware with explicit user ratings entered by users at the Android Market portal.

In February 2010 we have freely released AppAware on the Android Market and, at the time of writing, AppAware has been downloaded from more than 24,000 unique users, 10,500 of which are active in the last week. The users voluntarily installed AppAware on their mobile device and at present we have successfully collected more than 1,400,000 installation, update and removal events. While collecting these events we were also collecting information from the Android Market on the same applications being traced by AppAware clients, for a total of 18,740 Android apps. In this way we were able to retrieve the number of ratings, the average of ratings and the download category for each of the monitored Android applications. From these data we computed Table 1 that shows implicit and explicit rating statistics for the applications under study. As the Android Market does not provide the exact number of downloads for an application, we used the 9 download categories an application can be part of (these categories are provided by the Android Market itself). In order to compute the average percentage of users giving a rating to an application we took the median between the lower and upper limit for each download category (we took 25 for the category "<50" and 250,000 for the category ">250,000").

Table 1. Implicit and explicit rating statistics for the 9 download categories defined by Google.

		AppAware					Android Market				
Download categories	**Number of applications under study**	**Implicit ratings per application**		**Average % of users sending events**	**Acceptance per application[5]**		**Explicit ratings per application**		**Average % of users giving ratings**	**Average ratings per application (1-5)**	
		Avg	**St. dev**		**Avg**	**St. dev**	**Avg**	**St. dev**		**Avg**	**St. dev**
<50	906	5.71	40.55	0.228	3.48	0.69	1.39	1.66	0.056	2.4	2.10
50-100	609	2.33	2.98	0.031	2.79	0.87	3.10	2.80	0.041	3.24	1.67
100-500	3109	3.87	7.51	0.013	3.12	0.77	7.36	8.02	0.025	3.5	1.24
500-1,000	2194	6.02	8.55	0.008	3.22	0.74	13.12	15.22	0.017	3.48	0.93
1,000-5,000	5133	12.73	25.46	0.004	3.11	0.70	33.96	48.44	0.011	3.48	0.82
5,000-10,000	2060	25.99	54.06	0.003	3.04	0.71	88.14	115.63	0.012	3.53	0.72
10,000-50,000	3256	56.37	155.60	0.002	2.97	0.66	247.77	332.18	0.008	3.63	0.65
50,000-250,000	1101	172.99	311.28	0.001	3.03	0.63	1220.01	1370.21	0.008	3.89	0.52
>250,000	372	787.18	1046.34	0.003	3.32	0.64	9189.60	13434.27	0.037	4.12	0.40

Table 1 shows that only a little percentage of users give explicit ratings to applications, highlighting this uncommon activity among users. What is also evident is that the average of ratings is very high and the expected quality for an application on the Android Market is 3.47 out of 5 stars. This suggests that an application having an average of 3 stars is under the Market standards and, despite 3 stars can be considered a good average, the application might still not be a very good one. With implicit ratings AppAware tries to overcome both difficulties. For achieving this, our assumption is that an application not considered good or useful is removed (i.e. uninstalled) from the device. Despite the little penetration (24,000 users over 8 million people having an Android phone[6] and thus the Market application), we notice that AppAware performs reasonably well in terms of "average implicit ratings per application" for the low download categories up to 5,000 downloads. This can be explained since these categories contain relatively new applications that have been launched while AppAware was already installed on many devices. Therefore it makes sense to compare data from these categories, whereas is not significant for high-download categories and "old" applications. As a result, for extremely new applications, i.e. category "<50" downloads, AppAware has in average 4 times more (implicit) ratings than the Market.

[5] Values originally in the 0-100 range have been scaled to a 1 to 5 range for better comparing them with explicit ratings from the Android Market.

[6] According to a report published by the market research firm Canalys on February 8, 2010.

We further have analyzed the correlation of implicit ratings from AppAware with the explicit ratings from Android Market. We considered only applications which had at the same time more than 20 users' ratings and more than 20 AppAware events from distinct users, thus a total of 5,618 applications were included in the computation. The correlation coefficient is 0.46 that suggests a weak correlation of the two datasets. We were expecting some correlation since objectively good applications must appear to be good on both datasets. Similarly, the same holds also for objectively bad applications.

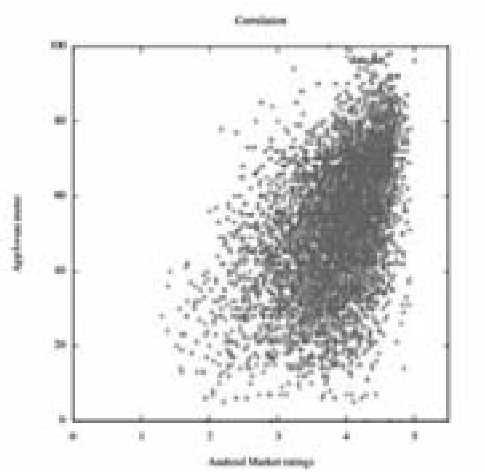

Figure 2. Correlation between AppAware acceptance rates (meter values) and Market average ratings for Android applications.

On the other hand, as also Table 1 suggests, the majority of applications have very good ratings on the Android Market but a significant fraction of them have worse acceptance rate from AppAware users (lower-right part of the chart in Figure 2), thus leading to the weak correlation we found. From Figure 2 it can be also noticed that there are almost no applications with a high acceptance rate from AppAware and low explicit ratings.

4 Conclusion

This paper has described the current status of AppAware, a mobile application that captures and shares installations, updates, and removals of Android programs in real time. AppAware introduces an implicit rating mechanism for Android programs where these events contribute to the rating algorithm described in Section 3. We assume that applications not liked by users tend to be uninstalled, however users might not always remove bad applications, or they could even uninstall AppAware thus not submitting any removals of previously installed programs. To accommodate this issue, we are considering ruling out inactive users from the computation.

We showed the low ratings activity on the Android Market and how these ratings are highly skewed towards the range 3 to 5 (as also shown in Figure 2). AppAware appears to be superior in term of number of ratings for the download categories up to 5,000 downloads, as we can assume these categories to have new applications and thus data that can be compared with significance between the two platforms.

As future work we plan to further improve our implicit rating mechanism by considering the existing explicit ratings on the Android Market, and including time spans between installations and removals of applications, thus giving a bonus to programs that remain installed for a long time. Additionally, we can check for possible systematic biases due to the nature of certain applications. For example, games might have a limited life based on the number of available levels, or lite (i.e. demo) version of certain applications might be uninstalled while moving to the full version. Mining these behaviors from users' activities could further develop the application meter presented in this paper.

Reference

[HO09] Holzer, A.; Ondrus, J.: Trends in Mobile Application Development. In Proceedings of the Mobile Wireless Middleware, Operating Systems, and Applications Workshops, 2009

[BPB10] Bohmer, M.; Prinz, M.; Bauer, G.: Contextualizing Mobile Applications for Context-aware Recommendation. In Adjunct Proceedings of Pervasive 2010, Helsinki, Finland, 2010.

[Cl01] Claypool, M.; Brown, D.; Le, P.; Waseda, M.: Inferring user interest. Internet Computing, IEEE, vol.5, no.6, pp.32-39, Nov/Dec 2001.

[DN06] Dellarocas, C.; Narayan, R.: What motivates consumers to review a product online? A study of the product-specific antecedents of online movie reviews. In WISE 2006, Evanston, Illinois, USA.

[Ea04] Eagle, N.: Can serendipity be planned? MIT Sloan Management Review. Fall 2004, vol.46, no.1, pp. 9-15.

[GM10] Girardello, A.; Michahelles F.: AppAware: Which Mobile Applications Are Hot? (submitted as demonstration paper for MobileHCI 2010, Lisboa)