ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

Personalized Recommender System for Smartphones based on Application Usage

Chirag Hingorani[1], Reshma Bhatia[2], Omesh Pathai[3], Tarun Mirani[4]

B.E., Computer Engineering, Vivekanand Education Society's Institute of Technology, Mumbai, India [1]

B.E., Computer Engineering, Vivekanand Education Society's Institute of Technology, Mumbai, India [2]

B.E., Computer Engineering, Vivekanand Education Society's Institute of Technology, Mumbai, India [3]

B.E., Computer Engineering, Thadomal Shahani Engineering College, Mumbai, India [4]

Abstract: The rebellious growth of the mobile application market has made it a significant challenge for the users to find relevant applications in crowded Application Stores. To diminish this problem, existing solutions often use the user's application-download history or user-rating to recommend applications that might interest them. However, the user downloading an application does not indicate that the user likes that application. Using user-ratings, on the other hand, suffers from tedious manual input and potential data insufficiency problems. In this paper, we present a system that makes personalized application recommendations by analyzing how the user actually uses his installed applications. Based on all user's application usage records, our system employs an item-based collective filtering algorithm for individualized recommendations.

Keywords: mobile application market, user rating, recommend applications, item-based collective filtering algorithm.

I. INTRODUCTION

The mobile application market has seen tremendous growth in recent years, with Apple's App Store boasting more than 1,100,000 applications and Google's Android Market also having well above 1.3 million applications. The mobile application market is estimated to reach $25 billion by 2015, which was $6.8 billion in 2011[1]. While the application stores allow the users to browse top applications in different categories, it is still difficult for the users to find interesting applications that they may like. To solve this problem, many solutions have been proposed that mostly leverage the user's application download history or their ratings of other applications as the basis for personalized recommendations. Installing an application doesn't indicate that the user liked that application. The user may simply want to try out that application, and may never use it again or may have uninstalled it. On the other hand, asking the users to rate each application they use can provide a better picture of their application taste. This approach, however, requires manual input and not many people are willing to or can remember to consistently provide their input. In this paper, we present a system that makes personalized mobile application recommendations. The novel feature of our system is that it measures how the user actually uses his applications, and the usage scores are then used by a Collaborative Filter (CF) algorithm to make personalized recommendations. Our solution is completely automatic without requiring manual input and also is adaptive to the changes of the user's application taste.

II. RECOMMENDER SYSTEM

Recommender systems are best known for their use on E-Commerce web sites, which recommend products to the visiting users based on their past purchase history or their ratings on other items [2]. Collaborative Filtering (CF) techniques are widely used to make automatic predictions (filtering) about the interests of a user by collaborating many users' interests information. The intuition behind CF is that those showed similar tastes in the past tend to agree again in the future. These recommendations are specific to the user, but leverage data collected from many other users. In the context of mobile applications, we expect that CF algorithms will work in a similar fashion by clustering the applications into related groups based on similar personal tastes. Presumably people with similar taste will like a similar set of applications. While not directly exposing the users' personal tastes, CF algorithms do implicitly link related applications driven by user similarities to provide a foundation for personalized recommendations.

III. USAGE SCORE

The input to CF algorithms is the score of the (user, item) pairs. A score can be binary 0 or 1, indicating whether the user has liked the item or he hasn't. In the context of mobile applications, we may assign 1 as the score to the (user, app) pair if the user has installed that application, or assign 0 if otherwise. Our system chooses to passively observe how the applications are being used with an assumption that the more an application is being used suggests that the more the user likes it. The users continuing to use similar applications can be considered as like-minded users. We have applied a RFD (Recency, Frequency, and Duration) model. Recency means how recently a user has interacted with the application. Frequency means how frequently the user interacted with the application in a given time period. Duration is to measure how long the user actually interacted with the application. By collaborating these three values, RFD can provide a good estimate of how much a user "likes" an application. We define Recency R as the time elapsed

since the last use of the application a by the user u, frequency F as the number of times u interacted with the application a within a certain period, and duration D as the total duration time that u interacted with the application during that period.
The usage score is thus represented as
Usage of a by u = W_R*R + W_F*F + W_D*D
where W_R, W_F , W_D are the weights based on their relative importance. The combination of these three measurements reflects the user's application taste. The applications that have been used more recently, more frequently and more time are likely to be favoured more by the user.

IV. Slope One Prediction

Given an application a that the user u has not used before, our system uses the Slope One algorithm to predict the RFD score U_a reflecting how u will like a as follows.
Let S(u) be the set of applications u has used, and let $R_{u,a}$ be the set of applications u has used and is relevant to i (meaning some other user(s) used the application in $R_{u,a}$ together with j). So
$R_{u,a} = \{i|i \in S(u),i\ 6= a,card(S_{a,i}) > 0\}$
where $S_{a,i}$ is the set of the users who have used both i and a and card($S_{a,i}$) is the number of the users in that set. The prediction of U_a thus is
P(U_a) =
1 card($R_{u,a}$)
X
$i \in R_{u,a}$
($dev_{a,i} + U_i$)
where
$dev_{a,i}$ = X $w \in S_{a,i}$
vw`j − vw`i card($S_{a,i}$)
.
Basically the $dev_{a,i}$ is the average score difference between a and i from all the users who have used both of them. Then our system predicts U_a based on U_i by adding $dev_{a,i}$ to U_i and taking an average for all relevant application i. The predictor is in the form of y = x + b, thus the name of Slope One. The simplicity of this approach also makes it easy to implement, and its predication accuracy is comparable to more sophisticated and computationally expensive algorithms [3].

Algorithm 1 Similarity Matrix
for ordered vector of app ids T_u by user u $\in$ U do
while size(T_u) > 1 do
find an application p $\in$ T_u with score $V_{u`p}$
$T_u = T_u - \{p\}$
for each application q $\in$ T_u with score $V_{u`q}$ do
Diff(p,q) = $V_{u`p} - V_{u`q}$
Count(p,q) = Count(p,q) + 1
end for
end while
end for
for each pair application p and q do
Diff(p,q) = Diff(p,q)/Count(p,q)
end for
return Diff

One of the drawbacks of simple Slope One is that the number of scores observed is not taken into consideration. Assuming that we are given the scores of user u on applications i and k to predict the score of user u on application j. If 5000 users have used i and j whereas only 50 users have used k and j, the score of user u on i is likely to be a much better predictor for j than the score of user u on k is. Taking this into account, the prediction can be changed to
Pw(uj) =
P $i \in R_{u,a}$ ($dev_{a,i} + U_i$)card($S_{a,i}$) P $i \in R_{u,a}$ card($S_{a,i}$) **-(i)**

This approach is called the Weighted Slope One and the Algorithm 1 shows how to compute the similarity matrix of the applications using this approach. Suppose there are N applications and M users, the computation of the similarity matrix table can be time intensive, with $O(N^2M)$ as the worst case. In practice, however, the complexity is closer to O(NM), as most users use only a few applications. Once the applications' similarity matrix is computed, our system makes recommendations for the user u as follows. For an application a not used by the user u, we can calculate its weighted slope one score Pw(U_a) using Equation (i), while the $dev_{a,i}$ can be looked up in the similarity matrix. Our system computes the scores for all applications a not in S(u), if there is a Diff(j,i) entry for any i in S(u), and returns the top N applications with the highest scores.

V. Conclusion And Future Work

In this paper we present a system that uses Collaborative Filtering to make personalized mobile application recommendations based on the user's actual application usage patterns. Unlike other approaches that uses the user's application download history or ratings, our system is completely automatic without requiring manual input and is adaptive to the potential changes of the user's application taste.
In the future work, we plan to improve the recommendation algorithm, such as by integrating the user context[4]. Finally, the application usage records collected when used will allow us to perform detailed analysis of the real world mobile application usage patterns at a much larger scale not done before.

References

[1] S. Perez. Mobile app marketplace. ReadWriteWeb.com, Jan. 2011.
[2] J. B. Schafer, J. Konstan, and J. Riedi. Recommender systems in E-commerce. In Proceedings of the 1st ACM Conference on Electronic Commerce (ACM-EC), 1999.
[3] D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. In Proceedings of SIAM on Data Mining, 2005.
[4] W. Woerndl, C. Schueller, and R. Wojtech. A hybrid recommender system for context-aware recommendations of mobile applications. In Proceedings of the IEEE ICDE, 2007.