Evolving Mobile App Recommender Systems: An Incremental Multi-objective Approach*

Xiao Xia, Xiaodong Wang, Xingming Zhou, and Bo Liu

School of Computer Science, National University of Defense Technology, Changsha, P.R. China, 410073

Abstract. Existing recommender systems for mobile apps mainly focus on single objective which only reflects monotonous app needs of users. Therefore, we evolve the existing mobile app recommender systems leveraging the *multi-objective* approach. Moreover, to avoid risks introduced by dramatic system vibration, we realize the system evolution in an *incremental* manner. To achieve these two goals, we model the recommendation generation of the evolved system as a multi-objective optimization problem and propose a new *rank aggregation based evolving scheme* to gently evolve the systems. Furthermore, we propose a new recommending scheme for mobile apps based on Latent Semantic Analysis and leverage it to evolve the existing system. Real data evaluations have verified the effectiveness of our approach.

Keywords: Mobile app, multi-objective, incremental, rank aggregation.

1 Introduction

The tremendous increase in population of mobile apps has given birth to the challenge of app discovery. To meet this challenge, online markets have employed recommender systems to provide users with app suggestions. For instance, AppJoy [1] filters out app choices based on personalized app usage patterns. AppBrain [2] generates recommendations of the same category with those have been installed by users while AppAware [3] exploits the context information for app recommendations.

Such existing mobile app recommender systems (MARS) are of help to users for app discovery. However, they mainly focus on the recommendations of a single objective, which only reflects the monotonous app needs of users. Specifically, Appjoy utilizes focuses on the similarity among apps with respect to their usage patterns. AppBrain exploits the category of apps to capture their similarity. Systems such as the AppAware and others pay their attention to discover apps that are of similar using contexts. Therefore, most of the existing MARSs are advancing their recommendations by solely taking the app similarity into consideration.

On the other side, recent studies have recognized that single-objective systems may be of little use or even negative [4] while other aspects of recommendation quality are of similar important to the similarity [5,6]. Thus the multi-objective recommender

* This work is supported by the projects of National Natural Science Foundation of China: *No. 61070201, No. 61170260* and *No. 61202486*.

James J. (Jong Hyuk) Park et al. (eds.), *Future Information Technology*,
Lecture Notes in Electrical Engineering 276,
DOI: 10.1007/978-3-642-40861-8_4, © Springer-Verlag Berlin Heidelberg 2014

systems are attracting increasing interests [7]. However, the study of multi-objective MARS is still missing in the literature. Therefore, we study the development of future MARS leveraging the multi-objective approach. Moreover, in the evolution of systems, severe system vibration may result in significant loss of customers. Therefore we utilize an incremental way to design the evolution for avoiding dramatic changes.

Main efforts and contributions of this paper are as follows:

- We propose a novel Latent Semantic Analysis (LSA) based scheme for mobile app recommendation, which overcome the user experience constraint.
- We model the recommendation generation of the evolved MARS as a multi-objective problem and propose a rank aggregation based evolving scheme, which realizes the incremental evolution of multi-objective MARS.
- Through real data evaluations, we verify the effectiveness and identify the potential of developing MARSs leveraging the incremental multi-objective approach.

2 LSA Based Recommending Scheme

Most online app markets generate app recommendations based on the behaviors of the users. For instance, the Google Play market provides users with apps that "users who installed this also installed". Such a method may experience a cognitive constraint since users are not able explore even a majority of apps in a population over 700,000.

To conquer such limitations of user experiences, we propose the novel LSA based recommending scheme for mobile apps, which is also used to define the multi-objective optimization problem and to realize the incremental evolution. The scheme compares the app descriptions by using the LSA method thus to measure the similarity among apps. Based on the similarity measurements, it then recommends users with apps that are of the most similar to those they have accessed. This scheme inspires the recommender system to make better use of the global information of apps, i.e., the app descriptions. By this way, our scheme conquers the limitation of user experiences.

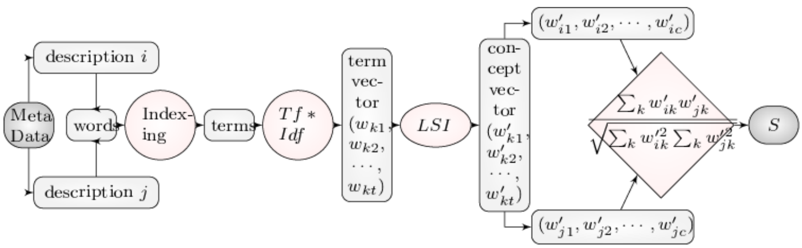

Fig. 1. Process of metadata similarity measurement using LSA

The process of applying LSA to measure the similarities among apps is illustrated in Figure 1. The LSA represents app descriptions by vectors of weighting terms. It then projects the term-description matrix to a lower-dimensional space, through which it mines the meanings and the variability of terms underlying the descriptions. After all, the term-description space is projected to the semantic space, which represents semantic concepts instead of raw terms. By this way, the similarity measurement which is based on the concepts comparison is expected to gain a better understanding.

3 Multi-objective Recommendation

To provide multi-objective app recommendations, we model the recommendation generation of the evolved system as a multi-objective problem in this section.

3.1 Objectives of Evolved System

We capture not only the needs of the users, but also the expectations of the developers and the online market. Therefore, we denote the following evolution objectives. The notations to be used are listed in Table 1.

Ranking. As users may want to find out and compare similar apps as those they have accessed, we define the objective "Ranking" to recommend the most similar apps to users. It is denoted as the average similarity between i and all its recommendations:

$$Ranking(i) = r_i^T * Lsa * e_i / N_R. \tag{1}$$

Range. As users also want to find novel apps while developers need to promote new apps, recommending similar apps alone is not sufficient. Therefore, we define the objective "Range" to recommend novel or even serendipitous apps. To define the *Range* objective, we capture the *category diversity* and *item diversity* of the recommendations. The former metric helps to improve the novelty and the scope of app discovery. The later avoids that the recommended apps are too similar to each other.

We define the *category diversity* based on both the number of categories and the proportion of apps of different categories:

$$D_c(i) = (C_d * (1 - e_i) / C_d * 1) * C_n * 1, \tag{2}$$

where $C_d = r_i^T * C$, which indicates how many apps of each category are recommended. The C_n is derived from the C_d, where $C_n(k)=1$ if $C_d(k) \geq 1$ and $C_n(k)=0$ otherwise. The C_n denotes the categories that the recommendations have covered. We define the *item diversity* as the average of the intra-list dissimilarity:

$$D_i(i) = 1 - r_i^T * Lsa * r_i / N_R(N_R - 1). \tag{3}$$

Therefore, we can derive the *Range* objective by:

$$Range(i) = D_c(i) * D_i(i) \tag{4}$$

Revenue. While the recommending services are provided by online markets, it is rational to cover the profit expectations of them when evolving the existing systems. Therefore, we define the objective "Revenue". To define the *Revenue* objective, we leverage the price and installations of apps to capture their profit potentials. That is,

$$Revenue(i) = lg(r_i^T * diag(P) * I + 1), \tag{5}$$

where the *lg* operation is introduced because the number of app installations varies across large scales.

Robustness. Since the preferences of users, developers and online markets vary over time, the recommender systems should be designed to be adaptive. To this end, we integrate the Robustness to our evolved system. to achieve better *Robustness* of the system, we define the category diversity parameter $\theta_c(i)$ and the price diversity parameter $\theta_p(i)$to tune the performance of the system. They are defined to determine the upper bounds of recommended apps in different categories and those are not free.

Table 1. Notation definitions

Notation	*Definition*
A	the set of all apps
N_A	the size of A, i.e., the number of all apps
$R_{i,}$	app recommendations for app i
N_R	the size of R_i, i.e., the number of recommended apps
r_i	N_Ax1 vector, $r_i(k)=1$ if $k\in R_i$, else $r_i(k)=0$
Lsa	N_AxN_A matrix, $Lsa(i,j)$ is the similarity between i and j
C	N_CxN_C matrix, $C(i,j)=1$ if app i is in the category j, else $C(i,j)=0$
c_i	category of app i
P	N_Ax1 vector, P_i is the price of app
I	N_Ax1 vector, I_i is the installations of app i
e_i	$e(i)=1$, $e(j)=0$ for any j that $j\neq i$
1	$1(i)=1$ for all i

3.2 Problem Formulation

Based on the definitions above, we denote the R^3 metric, to measure the fitness of recommendations. Given i and the recommendation R_i for it:

$$R^3(R_i) = R^3(r_i) = Ranking(r_i)^{\delta_1} * Range(r_i)^{\delta_2} * Revenue(r_i)^{\delta_3}. \tag{6}$$

where the δ_x weights each kind of objectives so that the system can obtain better robustness. Based on the R^3 metric, we derive the objective of the evolving process to be the overall R^3 of all the apps $\sum_{i\in A}R^3(i)$. Furthermore, given the constraints of the limited space on web pages and the control parameters, we model the evolution process as a constrained optimization problem as follows, where P is the price matrix P and $P(i,j)=1$ denotes that app i has the price j.

$$Max \sum_{i\in A} R^3(r_i) \tag{7}$$

$$\text{s.t.} \quad r_i^T * 1 = N_R, \tag{8}$$

$$r_i(i) \in \{0,1\}\ \forall i = 1,\dots,N_A, \tag{9}$$

$$r_i^T * C * (1 - e_{ci}) / r_i^T * C * 1 \le \theta_c(i), \tag{10}$$

$$r_i^T * P * (1 - e_{ci}) / r_i^T * P * 1 \le \theta_p(i). \tag{11}$$

4 Incremental Revolution

To achieve the incremental revolution, we introduce the method of *rank aggregation*, which is denoted as deriving a "consensus" ranking of the alternatives, given the diverse ranking preferences of various criteria. The rank aggregation has been applied in many areas, such as web search [8]. Furthermore, for the purpose of generating multi-objective recommendations, we design our evolving scheme following the optimization problem presented in Section 3.2. To be formal, the evolving scheme is defined as a problem of finding the rank aggregation method Ra, which satisfies:

$$R_o(i) = Ra(R_b(i), R_m(i)), \tag{12}$$

$$R^3(R_o(i)) \geq R^3(R_m(i)) \wedge R^3(R_o(i)) \geq R^3(R_b(i)), \tag{13}$$

where the $R_b(i)$ is the set of recommended apps provided by the Google Play market, the $R_m(i)$ is the set of apps recommended by our LSA based method and the $R_o(i)$ is the app recommendations generated by the evolved system.

There are $C_{|R_m|+|R_b|}^{|R_o|}$ recommendation candidates for each app, thus the global optimization could by computational expensive. We therefore propose a heuristic evolving scheme which is described in Algorithm 1. The basic idea of our heuristic scheme is to generate two ranks for further aggregation based on the sets R_b and R_m. We firstly weight them by the app similarity/dissimilarity, price and installations values. We then filter apps out to generate the R_o following the heuristic policy in the scheme.

Algorithm 1. The Evolving Recommending Scheme

```
Require: the number of recommended apps N_R
For i in A do
  initialize k with 0, initialize R_o^k(i) with ∅
  While k<N_R and R_b(i)∪R_m(i)≠∅ do
  find the app j in R_b∪R_m which maximizes
     Lsa(j,i)*D_i(R_o^k(i)+j)*P(j)*I(j)
    If {R_o^k(i),j}satisfies the category and price di-
versity parameters then
    let R_o^{k+1}(i)= R_o^k(i)+j, k=k+1
    End if
  delete j from R_b∪R_m
  End while
End for
Return  R_o^k(i)
```

5 Evaluation

To conduct the evaluations and verify the effectiveness our methods, we implement both the LSA based recommending scheme and the rank aggregation based evolving scheme. We then compare the R^3 metrics of the three kinds of recommendations, i.e., the existing recommendations R_b, the LSA based recommendations R_m and the evolved recommendations R_o. For clear illustration, we normalize the values of all recommendations by that of the R_b, i.e., $R^3_{norm} = R^3(R_x(i)/ R^3(R_h(i))$. We further measure the similarity, the intra-list item diversity and the average profit of the three recommendations to better understand the incremental realization of the scheme.

Figure 2(1) shows that the evolving scheme shows off an advanced performance to achieve multi-objective recommendations, comparing to each single method. Moreover, from Figures 2(2), 2(3) and 2(4), we can see that the evolving scheme realizes the incremental evolution of recommender systems by conducting tradeoffs between the existing system and the new method, which avoids severe system vibration.

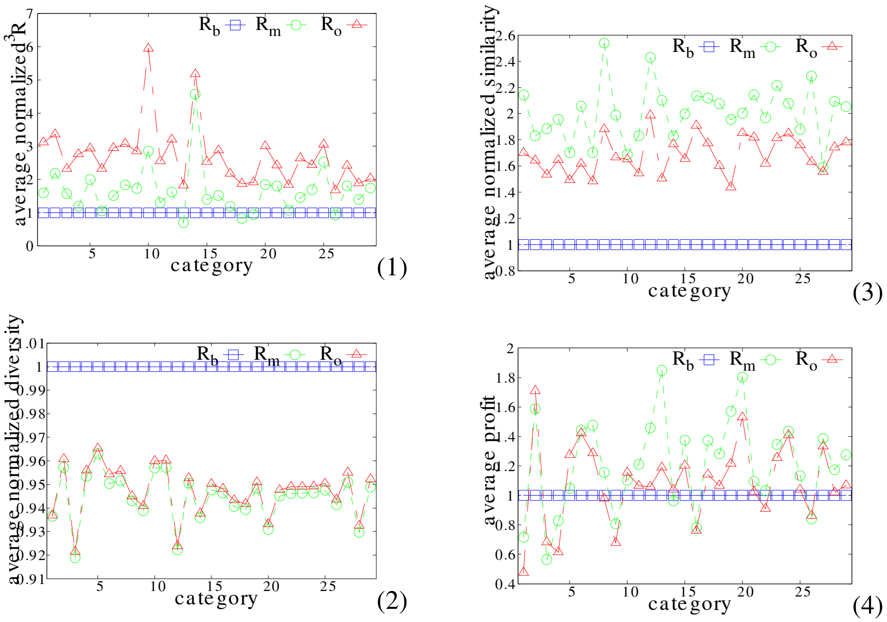

Fig. 2. The R^3 metric(1), intra-list diversity(2), similarity(3) and profit (4) of recommendations, illustrated by each category of apps

6 Conclusion

To evolve the MARSs, we propose a LSA based recommending method, model an optimization problem and design an evolving scheme for incremental evolution. By this way, we verify the effectiveness of the multi-objective and incremental approach.

References

1. Yan, B., Chen, G.: Appjoy: personalized mobile application discovery. In: MobiSys 2011, pp. 113–126. ACM, New York (2011)
2. Appbrain: Appbrain (April 15, 2013), `http://www.appbrain.com`

3. Girardello, A., Michahelles, F.: Appaware: which mobile applications are hot? In: MobileHCI 2010, pp. 431–434. ACM (2010)
4. McNee, S.M., Riedl, J., Konstan, J.: Accurate is not always good: How accuracy metrics have hurt recommender systems. In: Extended Abstracts of the 2006 ACM Conference on Human Factors in Computing Systems, CHI 2006 (2006)
5. Shi, Y., Zhao, X., Wang, J., Larson, M., Hanjalic, A.: Adaptive diversification of recommendation results via latent factor portfolio. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2012, pp. 175–184. ACM, New York (2012)
6. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. IEEE Trans. on Knowl. and Data Eng. 24(5), 896–911 (2012)
7. Rodriguez, M., Posse, C., Zhang, E.: Multiple objective optimization in recommender systems. In: Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys 2012, pp. 11–18. ACM, New York (2012)
8. Li, L., Yang, Z., Kitsuregawa, M.: Aggregating user-centered rankings to improve web search. In: AAAI 2007, pp. 1884–1885. AAAI Press (2007)