A Cooperation Based Metric for Mobile Applications Recommendation

Fei Wang, Zhe Zhang, Hailong Sun, Richong Zhang, Xudong Liu
School of Computer Science and Engineering
Beihang University
Beijing,China
Email: $\{wangfei, zhangzhe, sunhl, zhangrc, liuxd\}$@act.buaa.edu.cn

***Abstract*—As the smartphone is popular recently, mobile applications have been more and more attractive to users. With the exponential growth of applications' number, many works have been done to recommender the interesting applications for users. To check the effectiveness of these methods, the researchers are all tend to choose the traditional accuracy metrics to evaluate their system. However, considering the particularity of the mobile application recommendation, the accuracy evaluation metrics are not enough. Aiming at this problem, the goal of this paper is towards propose a new metric to evaluate the results of the mobile recommender systems and present an associated recommendation methodology. Base on the real usage pattern of mobile applications, we suggest a new evaluation metric called Cooperation. That is, when users use one application, they would enjoy using another one, we say those two applications have Cooperation Relation. In addition to, we also design and implement a strategy, which could improve the Cooperation degree of the mobile application recommendation results.**

I. Introduction

With the dramatically increasing of smartphone using, the popularity and quantity of the mobile apps are accelerated noticeably. Various applications markets such as App marketplaces for iOS, Android, and Windows Phone platforms have made it more convenient for users to discover apps, but it is still hard for users to find an app that could meet their needs. Furthermore, in the academic area, many researches have been conducted to design and implement the recommender system to suggest desirable applications for users [1], [2], [3], [4], [5], [6]. Confronted with a variety of recommendations methods, how to evaluate their effectiveness becomes more and more challengeable.

In the application recommendation domain, existing works all adopt the accuracy metrics (such as MAE, Precision, Recall, F1, P@N, ect.) to measure a recommender system, which are the typical evaluation metrics for conventional recommendation. However, in practice, even though a recommender system has a higher accuracy, it still could not meet the users' needs[7]. In addition to, in the paper [1], by giving the difference between mobile applications recommendation and the conventional items recommendation, the researchers illustrate that the typical conventional methods, which target at the accuracy recommendation, fail to make satisfying recommendation in the mobile application domain, such as user-based collaborative filtering (UCF), item-based collaborative filtering (ICF), content-based recommendation (CBR) and so on. Therefore, there is a great necessary to search for other evaluation directions to improve algorithms of the mobile applications recommendation systems.

By analyzing the habit when people use mobile applications, we could observe a phenomenon that there is a usage correlation between a pair of apps. And a person would need different applications under a similar situation. For example, if a user has installed an app for running training, they are less likely to need another running app, but they probably want other apps for squat, abs or legs training. We assume, sometimes, there is a kind of relation that we called cooperation existing between the pair of the apps. These apps with relation cooperation could be used in the same situation. In another word, if people use one of them, they would like to want the other one. These apps target at the users' similar interest, but they could not replace with each other from the perspective of the functionality.

To the best of our knowledge, we do a survey about all the works about recommendation areas of the mobile applications. The study of [8] just confirm our motivation mentioned in last paragraph. In that paper, they find that some apps have a high likelihood of co-occurrence. That is, when people uses one app, they are likely to use another one. Unfortunately, the contribution of their paper is that presenting the mobile app usage from a large amount of true data. They do not conduct further works based on this knowledge.

In this paper, we leverage this information to evaluate and improve the mobile applications recommender systems. At the first place, we do a review of related work about the recommendation of smarphone applications. In the next section, we provide a detailed description and a concrete explanation for our idea about new metric cooperation. Afterwards, in the section 4, we design a recommendation strategy based on the Android platform that could improve the recommendation performance under the metric cooperation. Up to the last section, we conclude this work and discuss some our meaningful future works.

II. Related Work

In this section, we generally review mobile applications recommender systems and relevant evaluating metrics.

A. Applications Recommender Systems

Given the significant challenge of helping the users to find interesting mobile applications, plenty of research works on the mobile app recommendation have appeared in the industry and the literature.

With the development of hardware on mobile devices, it is easy to collect a large amount of data to improve the recommendation results, such as location (GPS) data and users' activity feedback. Thus, there has been an increase in the possibility to build the context-aware recommender systems. The works of [5] and [3] integrated the location information to implements context-aware recommender systems. Similarly, in the paper [9], the authors propose a context-aware recommendation methods for mobile applications. Different with the former studies, the context they refer to is the users' activity.

In addition to, some other researchers focused on analyzing the characteristics of apps usage pattern, to provide new ideas for apps recommendations. The paper [2] argues that existing recommendation solutions in the industry could not effectively reflected users' needs in fact. Being different from the conventional domain, the users' downloading actions cannot indicate whether they like that application or not. Thus, AppJoy collects and analyzes the users' behaviors of using the applications, such as the using frequency, the using time and the recently using number for a personalized mobile app recommender system. Additionally, In the paper [1], the authors assume that a decision in download is greatly affected by the apps that had been installed by users. Thus, the authors propose a new mode called Actual-Tempting to reveal the contest between old apps and new apps, then using that information obtained to improve the apps recommendation.

B. The evaluation of app recommender systems

Confronted with various recommendations methods, how to appropriately evaluate them becomes more and more challengeable.

In the area of the applications recommendation, because of the lack of the massive true users data, most works fail to evaluate the recommendation algorithms. They just implement and deploy a prototype system, and recollect the response of the users. To prove that the recommender algorithm is convincing, the paper [3] realizes several different typical recommender algorithms and the context-aware one they proposed. And from the perspective of the paper, the particular recommendation result that is selected by users is the best one. Meanwhile, the works [9], [6] still do not evaluate the recommendation. They also implements and publishes a prototype. Actually, to create industrial value is the terminal goal of the academic study, so downloads of the recommendation app will show the public whether this recommendation method is good or not.

Even though there is a great difficulty to evaluate the applications recommender systems, after having collected a certain amount of data, there are still a few works to do some real evaluation. During the experiment in [1], the researchers choose relative precision and recall to measure the recommendation performance of the recommendation technique. Additionally, to evaluate the accuracy of the recommendation results of AppJoy[2], the authors select the precision metric.

From the perspective of [10], the degree of users satisfaction with recommendation results is not based on the accuracy in the recommendations. So we provide a new evaluation guideline for applications recommendation, in order to grasp the broader taste of users.

III. The definition of the metric cooperation

A. Preliminary

At the first place, we describe the Cooperation of two apps detailed as: That two apps app_i and app_j have cooperation relation means if a user uses app_i, there is a strong possibility that he or she would use app_j. In other words, these two applications have a great potential to be used under the same situation.

As the authors of [8] mentioned, there are many reasons that the pair of apps have highly correlated usage. Accordingly, we could definitely indicate that in which situations the apps could be used together. Based on those rules, persons could decide that which apps have cooperation relation.

1) Applications provide users the same content, but they are delivered in different forms. For example, several apps all focus on the local news, but they offer the information in different forms, such as the text news app, the picture news app and the video news app.
2) Applications serve the same target, but the functions they offer are not the same completely. For example, if a user would like to build his body, the app for training and the app for diets would probably be used by him at the same time.
3) Applications meet the users' same interests, but they are not the same. For example, if a user is a big fun of the NBA, he would like to install both the NBA news app and the NBA forum app with a great possibility.

Therefore, all apps available in app market could be sorted in different sets. And that apps are in the same set means these apps could be used in the same situation. Thus, each pair of apps in the same set has the cooperation relation. Inspired by what the former research [3], we implement and deploy an smartphone app to collect the usage pattern of correlated apps. As we erase the users and apps semantic information when collecting data, we would not violate users' privacy. So by this means, we could get the information about which apps could be used in the same situation.

B. Definition

Suppose that we have gotten all the situation sets that apps with cooperation could be used in. Each pairs of applications in the same set have the cooperation relation, meaning such applications could be used in the similar situation. The situation sets that apps could be used in like: $S_q = \{app_m, ..., app_n\}$.

According to our description of the cooperation relation between apps, we adopt the Jaccard Similarity Coefficient equation to quantify the cooperation relation between each pair of applications. We define the number of situations which both app_i and app_j could be used in as $S(app_i \bigcap app_j)$. And we count the number of situations which either app_i or app_j could be used in as $S(app_i \bigcup app_j)$. Thus, we have the quantified equation of the degree of the cooperation between two apps $C(app_i, app_j)$ as the following:

$$C(app_i, app_j) = \frac{S(app_i \bigcap app_j)}{S(app_i \bigcup app_j)} \tag{1}$$

Few of recommender system would give only one result, and they all would like to show the recommendation results as a ranked sets or an items package. Accordingly, we define the cooperation of a recommendation results set of applications, $app_1, ..., app_n$ to be the average cooperation between all pairs of apps in the results set.

$$C\{app_1, ...app_n\} = \frac{\sum_{i=1...n} \sum_{j=i+1...n} C(app_i, app_j)}{\frac{2}{n} \cdot (n-1)} \quad (2)$$

IV. Design and implement the recommendation algorithm

By focusing on users' need, our final goal is to find optimized methodologies to help users to find the applications they are interested in. Therefore, in this section, guided by the new evaluation direction we proposed, we design and implement a recommendation algorithm.

A. Recommendation

We implement two recommendation algorithms. One is the content-based similarity algorithm, which is as the baseline. The other is the Relation-Topic (R-T) algorithm, which is the improvement to increase the degree of cooperation of the recommendation results. We just illustrate the later one in detail.

Based on the definition we suggest in the last section, the pairs of apps with cooperation may be suitable for three situations: apps with same content, but in different type; apps for similar purpose, and apps reveal the same interests. With concluding those reasons, we could find two significant points. At the first place, that pair of apps must focus on the same area, or have the same topic. Secondly, those two apps cannot replace with each other, and they would offer users different functionalities. As a consequence, to reach our goal, we need keep focusing on those two conclusions.

1) Similarity of the Topic: To obtain the similarity of the topic between apps, we use the web page description of the application as the raw data. We choose the term frequency-inverse document frequency (tf-idf) method to make the text information computable. Accordingly, Each app is now represented by a real-valued vector of tf-idf weights. We use the cosine function to calculate the distance between every each pair of apps vectors, which is as their similarity value.

As a consequence, by quantifying the page description of the mobile applications and calculating the similarity value of each pair of apps, we could get a topic similarity matrix. In this matrix, each pair of the application has a value of topic similarity.

2) Difference of the Functionality: In order to show the difference of the functionality, we need more information than just the description details. As we choose the Android platform, apart from the page information of app, we can get the source code and related configure files of the app.

Android has a mechanism that called Intent. As said in [11], An Intent provides a facility for performing late runtime binding between the code in different applications. In a word, Android platform provides a mechanism that allows one application to invoke another application in runtime, to complete a task together. Therefore, we can assume that the former one aims at one purpose, but it cannot fulfill it independently, so it needs to launch the second app to complete this task. Therefore, we use the intent relation between apps to describe the functionality dissimilarity of a pair of apps.

A kind of intent may exist between more than one pair of apps, so we donate the intent frequency (if) as the number of times that the intent occurs to connect two apps. For the reason that the rarer the intent occurs, the more representative this intent would be, we adopt the inverse intent frequency (iif) to measure the weight of the intent relation between apps. As the following equation, the intent occurs rarer; its iif value would be larger; and the relation between apps it shows would be more important.

$$iif(intent_i) = \frac{1}{\log_{if(intent_i)} + 1} \quad (3)$$

What's more, for the reason that there may be several kinds of intent relation existing between a certain pair of apps, we sum up all the intents' iif value between two apps as the Dependency value, which indicates the functionality dissimilarity of the pair of apps.

$$Dependency(app_i, app_j) = \sum intent_i \quad (4)$$

Thus, by analyzing intent relations between apps, we also could get a matrix, which is used to reflect the difference of functionality between apps.

3) Combine the Topic Similarity and Intent Relation: We use the F-measure equation to combine those two elements in our quantifying standard.

$$CombineValue(app_i, app_j) = \frac{1}{\frac{1}{Similarity\{app_i, app_j\}} + \frac{1}{Dependency\{app_i, app_j\}}} \quad (5)$$

B. Relation-Topic Recommendation Strategy

1) Input: The query app, the similarity matrix, the dependency matrix.
2) Output: A package containing top-N mobile app.
3) Strategy details: With the content-based similarity algorithm, the normal approach is to select k most similar applications with query app. During our experiment, according to query app, we sort all the apps by the value combining the topic similarity and functionality diversity. And we retrieve the top k apps from this set as our recommendation results.

C. Illustrating the recommendation effectiveness by case study

At the time of this writing, as we have not gotten the enough data of the cooperation sets of apps until now, we carry out a case studies to illustrate effectiveness of our R-T algorithm under the cooperation metric. We select 7000 Android applications from AppChina, which is one of the most popular Chinese Android App market. We display two contrast groups of recommendation results: content-based similarity recommendation VS R-T recommendation. The recommendation results for target app Google Maps is showed in the figure1. The left table is the result of content-based

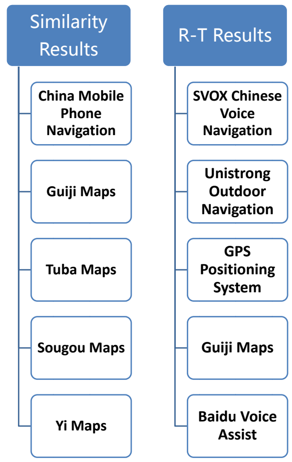

Fig. 1. Recommendation Results

similarity recommendation, while the right result is from our R-T algorithm. We could see that, in the left side, most of the recommendation results are maps apps, which all have the same functions with the target app, so users with Google Map are unlikely to select another maps app that may not performance better than Google Maps. Meanwhile, the right recommendation results are more various and meaningful. Both SVOX Chinese Voice Navigation app and Unistrong Outdoor Navigation app are navigation apps, which need a maps app to fulfill a complete navigating task. Thus, there is a great possibility for users to use them under the same situation. Similarly, the app GPS Positioning System need the target map to show the accuracy position. As to the last app Baidu Voice Assist, it helps users launch the Google Maps through the voice instruction. Therefore, these two apps would cooperate to finish a users' task. In conclusion, the right recommendation results have more cooperation apps with the query app than the left list.

V. Conclusion

Accuracy metrics have promoted the development of the recommendation systems to a great extend. Beyond it, we provide another perspective to evaluate the recommender system additional. On our opinion, the recommender systems should focus on the users.

In this paper, based on the usage patterns of smartphone applications. We propose a new user-centred metric cooperation to judge the quality of recommendation results. The meaning of the cooperation metric is the probability of the pair of the apps is used in the same situation. We give the both intuitive and formalized definition.

By conducting several experiments, we show how to improve the conventional recommendation algorithms could achieve a better performance under the cooperation metric. As a consequence, we find that the R-T recommendation algorithm we designed show a significant better cooperation degree, comparing with the concise content-based similarity recommendation.

To our best knowledge, we are first to investigate the recommendation measures beyond the accuracy metrics. In this research, we make the following contributions:

1) We offer a new direction for the future researchers in smartphone application recommendation.
2) We are first to exactly define the new metric aiming at evaluating the recommendation results.
3) We are first to adopt a new kind of raw data intent relation that would improve the apps recommendation.

After we publish this paper, and make our idea recognized in the academic areas, we could conduct cooperation with the big telecom companies and the famous smartphone apps market. On one hand, we could do our best to get the true and huge amount of the usage patterns data to make our metric comprehensive. On the other hand, based on the rules we will learn from the data above, we design an optimized recommendation algorithm to provide the users much better experience than existing works.

Acknowledgment

This work was supported by China 863 program (No. 2012AA011203), and National Natural Science Foundation of China (No. 61103031).

References

[1] P. Yin, P. Luo, W.-C. Lee, and M. Wang, "App recommendation: a contest between satisfaction and temptation," in *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 2013, pp. 395–404.

[2] B. Yan and G. Chen, "Appjoy: personalized mobile application discovery," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 2011, pp. 113–126.

[3] W. Woerndl, C. Schueller, and R. Wojtech, "A hybrid recommender system for context-aware recommendations of mobile applications," in *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*. IEEE, 2007, pp. 871–878.

[4] R. Burke, "A case-based reasoning approach to collaborative filtering," in *Advances in Case-Based Reasoning*. Springer, 2000, pp. 370–379.

[5] C. Davidsson and S. Moritz, "Utilizing implicit feedback and context to recommend mobile applications from first use," in *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*. ACM, 2011, pp. 19–22.

[6] M. Böhmer, G. Bauer, and A. Krüger, "Exploring the design space of context-aware recommender systems that suggest mobile applications," in *2nd Workshop on Context-Aware Recommender Systems*, 2010.

[7] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[8] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman, "Identifying diverse usage behaviors of smartphone apps," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 329–344.

[9] E. Costa-Montenegro, A. B. Barragáns-Martínez, and M. Rey-López, "Which app? a recommender system of applications in markets: Implementation of the service for monitoring users interaction," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9367–9375, 2012.

[10] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web*. ACM, 2005, pp. 22–32.

[11] Google, "Android developers," 2013. [Online]. Available: http://developer.android.com/reference/android/content/Intent.html