Discovering More Mobile Apps with Fewer Jumps

1Xiao Xia, 1Xiaodong Wang, 2Wei Wei and 1Xingming Zhou
1School of Computer Science, National University of Defense Technology, Changsha, 410073, China
2School of Computer Science and Engineering, Xi’an University of Technology, 710048, China

Corresponding Author: Xiao Xia, School of Computer Science, National University of Defense Technology, Changsha, 410073, China

ABSTRACT
The explosive growth of mobile apps in recent years makes it much more difficult for users to find out interesting apps. For this reason, online app markets, e.g., the Google Play market, have employed recommender systems. Such systems construct recommending networks of mobile apps so that they alleviate the challenge of app discovery. However, research efforts on the recommender systems are mainly focusing on the improvement of recommending accuracy. Little attention has been paid to measure and optimize the navigating effects of the recommending networks. To be specified, rare works in the literature have focused on advancing the efficiency of helping users explore more apps while discovering them with fewer jumps. This study therefore initially addresses and formulates such a problem. It further proposes to reconstruct the recommending networks after they have been formed by the recommender systems. Since mobile apps in the online markets have constituted complex networks, this study designs reconstructing schemes leveraging the complex network metrics and methods. Particularly, based on specific complex network measurements, e.g., the number of SCCs (strongly connected components), the APL (Average path length) and the node centrality, this study proposes two reconstructing schemes. After all, real-data evaluations have verified the effectiveness of the schemes proposed by this study.

Key words: Mobile app discovery, recommender system, recommending network, complex network, node centrality

INTRODUCTION
With the development of mobile computing, recent years have undergone a tremendous expansion in the population of mobile apps. As reported by recent investigations, both Apple Store and Google Play have hit the milestone of 700,000 apps in their markets (Cnet, 2013). Since the long tail theory tells that it is non-less important to sell a large number of items with small quantities than selling fewer popular items in large quantities, this study introduces such a strategy into the mobile app markets. Therefore, in addition to recommend users the most popular or latest apps, it is also non-trivial for the online markets to exhibit other apps which are less popular. In a word, the markets need to inspire the users to explore and discover apps as much as possible.

To provide users with alternative choices of apps, most of the online app markets have employed the recommender systems. Recommendations made by such systems are presented as links to apps on the websites. The apps and the links eventually form a recommending network for the mobile users. Users thus can follow those links to explore and discover more apps in the online market.
Therefore, the recommending networks are of help to users on finding out more apps. However, studies of recommender systems nowadays are mainly focusing on how to provide accurate recommendations to form the recommending networks. Little attention has been paid to investigate the recommending networks on the network measurements after their formation which exerts direct influence on the behaviors of users. Especially, the state of art is short of measuring and enhancing the recommending networks on their efficiency of navigating users to explore more apps while discover them within fewer jumps in the online app markets.

To meet such a problem, this study firstly defines the metric of app discovery efficiency to measure the recommending networks. Then, based on such a definition it proposes two reconstructing schemes to elevate the efficiency of recommending networks. Both the metric and the schemes are in the purpose of helping users to explore more apps and discover them within fewer jumps on the webpages. Since the recommending network exhibits as a complex network of mobile apps, the metric and schemes above are both based on measurements from the scope of complex networks, such as the strongly connected component, the average path length and the node centrality. Main efforts and contributions of this work are as follows:

- It initially proposes the problem of measuring and reconstructing the recommending networks for mobile apps after their formation. Such recommending networks which are formed by recommender systems of online app markets have been rarely studied on their overall efficiency on navigating users to explore and discover apps, especially from the scope of complex networks.
- It defines the metric of app discovery efficiency as a new measurement of the recommending networks. Based on such a metric it formulates an optimization problem to measure and enhance the recommending networks. Specifically, to solve the optimization problem, this study originally proposes two reconstructing schemes for the recommending networks to elevate their efficiency.
- It crawls real data of 103348 apps from the Google Play market. This study measures the existing recommending network and evaluates the schemes. The results reveal the potential to enhance the existing market from the scope of network measurement and concurrently verify the effectiveness of the schemes.

To the best of the knowledge, for the problem of examining and elevating the app discovery efficiency of recommending networks of online app markets, this work serves as the first effort to define and propose it. Furthermore, this study originally introduces the methods of complex network measurement and optimization to address and solve it.

The reminder of the study is as follows. It firstly constructs the recommending network based on the data crawled from the Google Play market. It then defines the app discovery efficiency for recommending networks, formulate the optimization problem of network reconstruction and propose two reconstructing schemes. After that, this study conducts evaluations using the real data to verify the effectiveness of the schemes. It further presents some related works and draws some conclusions.

**RELATED WORK**

To promote mobile apps to users, both the commercial and the research community have made various efforts, in which the commercial venues mainly concentrate on the recommender systems of apps. From the research community, people have studied various aspects of the mobile app
discovery, such as from the personalized scope (Yan and Chen, 2011) or even the best time for release (Henze and Boll, 2011). The methods leveraged vary from semantic relations (Lim et al., 2011) to implicit feedback analysis (Davidson and Moritz, 2011). However, little of them have focused on measuring the overall efficiency of the online market from the view of complex networks.

The optimization of complex networks have been applied successfully to solve problems in cascade resilience, internet design, power grid, traffic optimization, social networks and etc. (Thai and Pardalos, 2011). However, this is the first effort to introduce the methods of it, especially the topology optimization (Souza et al., 2009) to elevate the discovery of mobile apps.

**NETWORK CONSTRUCTION**

This section constructs the recommending networks of mobile apps based on data crawled from the Google Play market. To this end, it has gathered information of 103348 apps from the webpages of the online market. After that it parses a wide range of features for each app which includes the id, category and those apps recommended by the market.

Using the data, two kinds of networks are constructed. They are the app relationship networks which capture the recommending relationships among apps of the market and the app recommending network which navigates users to discover apps. The sizes of them are listed in Table 1.

**App relationship network:** For each app on the market, Google Play recommends several apps to users based on it. They are tagged with “user viewed this app also viewed” and “user installed this app also installed”. Obviously, such recommendations are based on the relationships among apps initiating from the viewing and installing behaviors of the market users. Those relationships are computed by the recommender system of the online market and are used to generate specific app recommendations for users.

Since the relationships are crucial to provide appropriate recommendations, it needs to keep them while reconstructing the recommending network. For this reason, this study constructs two app relationship networks named the “alsoview” network \( G_v \) and the “alsoinstall” network \( G_i \) for each relationship, respectively. As the relationships are symmetrical by semantic understanding, this study constructs both networks using undirected graphs. In these two networks, each node denotes an app in the market. Each edge between two apps indicates that one of them is recommended to the other.

**App recommending network:** The app recommending network \( G_r \) is constructed to capture the recommending links of the market websites, following which the users are navigated to discover more apps. Such a network exerts direct influence on users to explore and discover apps when they are surfing in the online market. Since the jumps among websites are directional, this study constructs the app recommending network leveraging the directed graph.

In the app recommending network, nodes denote apps from the market. Edges are linked from one app to the others which are recommended based on it by the market. To keep the relationships

<table>
<thead>
<tr>
<th>Table 1: Size of the app relationship networks and the app recommending networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>( G_v )</td>
</tr>
<tr>
<td>( G_i )</td>
</tr>
<tr>
<td>( G_r )</td>
</tr>
</tbody>
</table>
among apps in the recommending network, this study defines the type of the edge to be the “alsoview” or the “alsoinstall” which respectively indicates the relationship featured with “user viewed this app also viewed” or “user installed this app also installed”.

**NETWORK RECONSTRUCTION**

To elevate the efficiency of the online market on navigating users to explore apps as much as possible and discover them as quickly as possible, this section originally presents the method of reconstructing the recommending network after they have been formed by the recommender systems of online app markets. To be specifically, it formulates an optimization problem of network reconstruction and proposes two heuristic schemes for solutions.

**Program formulation**

**App discovery efficiency:** To measure the efficiency of the online market, this study defines a metric, i.e., the app discover efficiency (APE) for the recommending network of the online market. Since this study aims to navigate users to explore as much apps as possible and discover them as quickly as possible, the APE is defined to capture both the scope of apps that can be discovered and the jumps across webpages to reach them. To this end, two network measurements are introduced from the study of complex networks, i.e., the number of SCC (strongly connected component) and the APL (Average path length).

The SCC denotes the part of network in which each node can be reached by another. The number of SCCs is used to capture the scope of apps which can be discovered. To be detailed, cutting down this number may result in better connectivity among apps, thus from one app the users may discover more others. The APL of the network denotes the average of the shortest path length between all connected nodes. It is used to capture the average jumps from one app to another by webpages. Thus the shortening of the APL suggests that from one app the users may reach another by fewer clicks. Therefore, the app discovery efficiency is defined as:

\[
e = \frac{1}{\left\{\frac{\log n + 1}{\alpha + \beta}\right\}}
\]

where, \( n \) denotes the number of SCCs and \( I \) denotes the APL in the recommending network. The \( \log \) operation on the number of SCCs is to normalize its scale so that two parts exhibit approximate scales. The parameters \( \alpha \) and \( \beta \) are weighting parameters of each part which are set to 1 in evaluations of this study.

**Reconstruction constraints:** As mentioned in Section “App Relationship Network”, to capture the relationships among apps for generating appropriate app recommendations, the app relationship networks have been constructed. To make use of such relationships which are generated by the recommender system of the online market, constraints of the network reconstruction process are defined based on those app relationship networks. After all, two constraints are defined as below.

**C1 edge constraint:** \( R(ij) = 1 \) only if \( V(ij) = 1 \) or \( I(ij) = 1 \), where \( R \), \( V \) and \( I \) denote the adjacency matrix of recommending network, “alsoview” and “alsoinstall” app relationship networks, respectively.
C2 degree constraint: \( \Omega_i(i) | (P(ij) = \text{alsoview}) \leq 4 \), \( \Omega_i(i) | (P(ij) = \text{alsoinstall}) \leq 4 \), and \( j \in N(i) \), where \( N(i) \) is the neighbor sets of node \( i \), \( P(ij) \) is the type of edge \( e_{ij} \) and \( \Omega_i(i) \) denotes the out-degree of node \( i \) in the recommending network.

Essentially, the edge constraint indicates that only if apps are connected in the app relationship networks, they can be linked to each other in the reconstructed recommending network. The degree constraint suggests to provide recommendations of a limited number, i.e., 4, for each app which is corresponding with the design of the Google Play.

Program formulation: To design reconstructing schemes for the recommending network, this study formulates the network reconstruction as an optimization problem based on the definition of APE and the constraints generated above. The problem is presented as below:

\[
\begin{align*}
\text{Min} & (\log n+1)^{2}\eta_l^p \\
\text{s.t.} & \quad r_{ij} \leq 1 \\
& \quad r_{ij} \leq n_{ij} + v_{ij} \\
& \quad \sum_{\nu \in \eta} t_{ij} \leq 4 \\
& \quad r_{ij} \in R, v_{ij} \in V, i_j \in I \\
& \quad r_{ij} \in R, v_{ij} \in V, i_j \in I
\end{align*}
\]

where \( R, V \) and \( I \) denotes the adjacency matrix of the recommending network, the alsoview network and the alsoinstall network, respectively. The objective function of the problem is the variant of the app discovery efficiency while constraints of the problem are derived from the edge and the degree constraints.

Reconstructing schemes: To solve the optimization problem of reconstructing the recommending networks which are constituted of a large population of mobile apps, this study presents two heuristic schemes. Since there are two crucial metrics in the objective function, i.e., the number of SCCs and the APL, each of the two schemes is designed to focus mainly on one of the metrics. To be detailed, the SCC-APL scheme mainly focuses on reducing the number of SCCs while lessening the APL. The APL-SCC scheme concentrates on shortening the APL while cutting down the number of SCCs.

SCC-APL scheme: As mentioned in the above section, the SCC-APL scheme focuses mainly on reducing the number of SCCs. To this end, this study connects different SCCs in the network by adding edges among them. The original SCCs are then merged to bigger ones and the number of SCCs is reduced. However, due to the problem constraints, adding edges leads to removing edges which may results in the increasing of the APL. To meet such a problem, the solution of this study
is to connect specific nodes in different components which may make the nodes closer, thus to shorten the APL as well. To this end, this study introduces the *closeness centrality* of nodes in the network:

\[
cc(i) = \frac{1}{d_{a}(i)}
\]

where, \(d_{a}(i)\) is the average distance of node \(i\) to all other nodes in its SCC. The definition tells that nodes with large closeness centrality are generally closer to other nodes in the graph that those with small closeness centrality. Therefore, the strategy that connecting nodes with large closeness centrality in separate SCCs may also shorten the APL while merging the SCCs.

Based on the discussion above, the SCC-APL scheme is designed as a closeness centrality based SCC merging scheme. Basic steps of the scheme are presented as follows which serve as a preliminary framework:

**Step 1**: Finding SCCs in the network  
**Step 2**: Computing \(cc\) of nodes in each SCC  
**Step 3**: Connecting nodes of large \(cc\) between two different SCCs  
**Step 4**: Going to step 3 while the SCCs are not merged

Above framework describes only the rough steps, as in the implementation various algorithms can be utilized. For instance, to find SCCs in the network, the classic Tarjan’s algorithm (Tarjan, 1972) can be introduced with the Nuutila’s modifications (Nuutila and Soisalon-Soininen, 1994). To implement the whole scheme, it can utilize either the dynamic programming or the divide-and-conquer algorithm, depending on the number and the distribution of SCCs in the network.

**APL-SCC scheme**: The APL-SCC scheme mainly concentrates on shortening the APL of the network. Small world network models such as WS (Watts and Strogatz, 1998) and NW (Newman and Watts, 1999) have shown the power of the random rewiring on shortening the APL of complex networks. Therefore, this study leverages such a method to reconstruct the recommending network. While it also needs to keep the connectivity of the network to cut down the number of SCCs, this study adopts the preferential attachment which is introduced from the scale free network model (Barabasi and Albert, 1999). To make use of those two methods, this study inspires the edges to rewire to specific nodes which would build connections among separate parts of the network. To find such specific nodes, this study introduces the degree centrality of nodes in the recommending network. It is defined to be the node degree in app relationship networks, i.e., the network \(G_{r}\) or the network \(G_{c}\):

\[
dc(k) = d_{c_{r}}(k) \text{ or } d_{c_{c}}(k)
\]

which depends on the type of edge being rewired. Based on the definition, this study proposes that through preferential rewiring to nodes with large degree centrality, it keeps the connectivity of the network while shortening the APL.
After all, the APL-SCC scheme is presented as a degree centrality based preferential rewiring scheme. It repeats following steps for each node in the recommending network:

**Step 1:** Picking up a node \( i \) with the probability \( p_{\text{rewire}} \) which determines the proportion of nodes that will participate in the rewiring

**Step 2:** Generating the neighbor set \( N(i) \) of \( i \) in the app relationship network according to the type of rewiring edge, e.g., when rewiring “alsoview” edge, gathering the neighbors in the alsoview network

**Step 3:** Computing the \( \text{dc} \) of nodes in \( N(i) \)

**Step 4:** Rewiring \( i \) to the node with largest \( \text{dc} \) in \( N(i) \), discarding if the edge exists already

**EVALUATION**

To verify the feasibility and effectiveness of the two reconstructing schemes, this study implements and evaluates them based on the networks constructed from the real data crawled from the Google Play market. It examines the measurements of the existing recommending network \( G_r \), the SCC-APL network \( G_{\text{SCC-APL}} \) and the APL-SCC network \( G_{\text{APL-SCC}} \), in which the latter two denote the recommending networks after reconstruction by using the SCC-APL and the APL-SCC schemes.

Numerical results are listed in Table 2. For detailed understanding, this study also captures the distributions of the shortest path length between all pairs of nodes in the three networks and illustrate them in Fig. 1. Figure 2 depicts the number of SCCS, the APL and the app discovery efficiency. To illustrate the improvements clearly, this study has normalized the app discovery efficiency of all networks by that of the existing recommending network \( G_r \).

![Graph showing the distribution of shortest path length between pairs of nodes](image)

**Fig. 1:** Distributions of shortest path length in recommending networks \( G_r \), \( G_{\text{SCC-APL}} \) and \( G_{\text{APL-SCC}} \)

<table>
<thead>
<tr>
<th>Network</th>
<th>No. of SCCs</th>
<th>APL</th>
</tr>
</thead>
<tbody>
<tr>
<td>( G_r )</td>
<td>176</td>
<td>8.117</td>
</tr>
<tr>
<td>( G_{\text{APL-SCC}} )</td>
<td>69</td>
<td>6.211</td>
</tr>
<tr>
<td>( G_{\text{SCC-APL}} )</td>
<td>1</td>
<td>7.147</td>
</tr>
</tbody>
</table>
From the results it can be seen that the number of SCCs and the APL are both cutting down in all the reconstructed networks. Especially, the APL-SCC scheme contributes mainly on the shortening of the distance between nodes. Meanwhile, the SCC-APL scheme achieves better performance on merging SCCs. Both schemes improve a lot on the app discovery efficiency compared to the existing recommending network, i.e., 49.4% by APL-SCC and 288.6% by SCC-APL. Such evaluation results eventually verify the feasibility and the effectiveness of the schemes. Meanwhile they reveal the potential to improve the online markets from the view of complex network measurement and optimization.

CONCLUSION

To elevate the discovery of mobile apps, this study focus on the measurement of the recommending networks of the online markets. It defines the app discovery efficiency to measure them and proposes two schemes to improve the efficiency.

The two schemes which are focusing on different metrics introduce various complex network measurements. Both of them exhibit good performance in the real data evaluations. The reason they are not combined to one is that they are designed from different point of view and can be applied to different scenarios. However, in reality they can cooperate to provide robust applications. For example, future systems can leverage the SCC-APL scheme in the formation of the recommending network and the APL-SCC scheme when new apps are published.

This study originally addresses the problem of measuring and advancing the recommending networks after they have been formed by the recommender systems. It also reveals the potential of improving the online app markets using the methods of complex network. Furthermore, it presents effective schemes which are verified by real data evaluations.

The problem of this study is originally addressed and the schemes are preliminary designed. Therefore there are lots of efforts serving as future works. For instance, further studies could involve the computation of SCCs in the network, the cooperation of two different schemes and designing more optimization schemes.
ACKNOWLEDGMENTS
We would like to thank the anonymous reviewers for their valuable comments. This project is supported by the National Natural Science Foundation of China projects (Program No. 61202483, No. 61170280, No. 61070201 and No. 61172018). This work is also supported by Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No.2013JK1139).

REFERENCES