SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

April 2025, Vol. 68, Iss. 4, 142101:1–142101:16
https://doi.org/10.1007/s11432-023-3973-1

Characterizing the app recommendation relationships in the iOS app store: a complex network's perspective

Gang HUANG[1,2], Fuqi LIN[1], Yun MA[1,3*], Haoyu WANG[4*], Qingxiang WANG[5], Gareth TYSON[6] & Xuanzhe LIU[1,2]

[1] *Key Lab of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871, China*
[2] *School of Computer Science, Peking University, Beijing 100871, China*
[3] *Institute for Artificial Intelligence, Peking University, Beijing 100871, China*
[4] *School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*
[5] *Alibaba Group, Beijing 100029, China*
[6] *Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511458, China*

Received 8 January 2023/Revised 2 August 2023/Accepted 13 October 2023/Published online 16 January 2025

Abstract Mobile apps have become widely adopted in our daily lives. To facilitate app discovery, most app markets provide recommendations for users, which may significantly impact how apps are accessed. However, little has been known about the underlying relationships and how they reflect (or affect) user behaviors. To fill this gap, we characterize the app recommendation relationships in the iOS app store from the perspective of the complex network. We collect a dataset containing over 1.3 million apps and 50 million app recommendations. This dataset enables us to construct a complex network that captures app recommendation relationships. Through this, we explore the recommendation relationships between mobile apps and how these relationships reflect or affect user behavior patterns. The insights gained from our research can be valuable for understanding typical user behaviors and identifying potential policy-violating apps.

Keywords mobile app, recommendation, complex network, user behavior, policy-violating app

Citation Huang G, Lin F Q, Ma Y, et al. Characterizing the app recommendation relationships in the iOS app store: a complex network's perspective. Sci China Inf Sci, 2025, 68(4): 142101, https://doi.org/10.1007/s11432-023-3973-1

1 Introduction

Mobile apps have become an important component of the Internet ecosystem. However, with millions of mobile apps available in the market, it is sometimes hard for users to find their desired apps. Conversely, it is also challenging for developers to get their apps discovered. To enhance user experience, app markets try to recommend relevant apps to users. Taking the iOS app store as an example, it recommends apps on the display page of each app, in the form of "You Might Also Like" lists. Figure 1 provides an example of an app recommendation in the iOS app store. This page shows detailed information on the `Instagram` app, where the iOS app store recommends a number of related apps that users might find interesting. By clicking the "See All" button, users are redirected to a new page[1)] where additional recommended apps are listed. These recommendation relationships form a graph, which we term the app recommendation network. We conjecture that analyzing this network provides a unique opportunity to investigate how these relationships reflect and affect user behavior patterns at a massive scale.

Only a handful of prior work [1] has touched upon app recommendation relationships. Our focus lies in contrasting the characteristics of the app recommendation network with more traditional product recommendation networks (e.g., Amazon) and other real-world complex networks (e.g., the World Wide Web). Researchers have applied various network analysis methods to characterize these networks [2–9]. Given that mobile apps can be considered a unique type of product, we believe it is valuable to apply these techniques to investigate the properties of the app recommendation network and gain insights into potential improvements in the recommendation process.

* Corresponding author (email: mayun@pku.edu.cn, haoyuwang@hust.edu.cn)

1) https://apps.apple.com/cn/app/instagram/id389801252?see-all=customers-also-bought-apps.

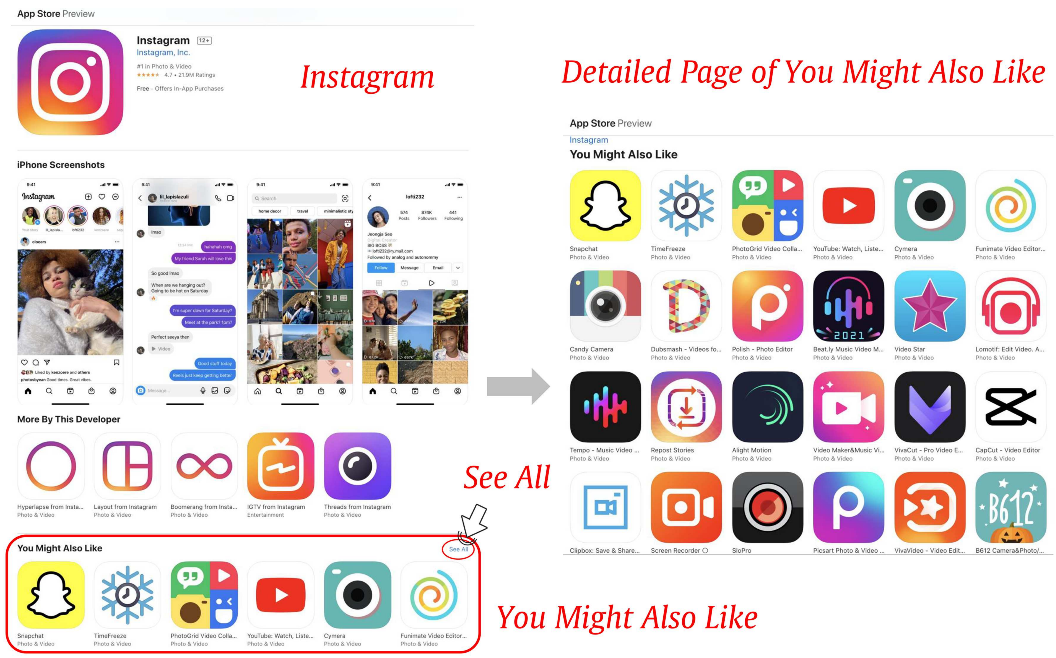

Figure 1 (Color online) Example of the "You Might Also Like" feature in the iOS app store (for Instagram).

To this end, we have collected a comprehensive snapshot of the iOS app store, encompassing over 1.3 million apps and more than 50 million recommendations. Using this dataset, we have reconstructed a large-scale app recommendation network (Section 3). We have thoroughly examined the properties of this network from various perspectives (Section 4). Our investigation concludes that the app recommendation network exhibits characteristics of a complex network, and we have identified both key similarities and differences between our app recommendation network and other real-world complex networks, especially traditional product co-purchasing networks. Driven by this observation, we further explore the relationships between app metadata and node properties, as well as the patterns of app recommendations (Section 5). Our findings highlight that certain types of apps are more likely to be recommended in conjunction with others, resulting in increased exposure. Drawing from the aforementioned insights, we discussed potential real-world applications of our data and findings (Section 6). Notably, the following results and findings stand out as particularly significant:

- The app recommendation network shares both significant similarities and notable differences with other real-world complex networks, especially traditional product co-purchasing networks. The properties have interesting implications in the context of app recommendations. Both networks exhibit small-world properties; i.e., most nodes are well connected. In contrast, although mobile apps are more interconnected than traditional product networks, users may find it harder to navigate from one app to another; i.e., the app recommendation network demonstrates a higher average degree and average path length compared to traditional product networks.
- Apps with certain features gain more exposure and are more likely to be recommended together. The nodes in the network exhibit heterophily, where apps in specific categories, highly-rated apps, and better-maintained apps gain more exposure. In contrast, the network exhibits homophily, where popular apps, highly-rated apps, apps with similar functionality, and apps catering to specific user demands are more likely to be recommended together. Developers can exploit these patterns to promote their apps effectively.
- The local structure of the app recommendation network serves as a reflection of typical user behaviors. We exploit the local structures of the app recommendation network to infer these user behavior patterns. By analyzing the motifs of the network, we reveal several interesting patterns. Notably, we observe the presence of high-exposure apps and app cliques within the same category. The results are a reflection of the app recommendation mechanism, which can provide valuable insights for developers to enhance their app promotion strategies.
- We propose a method to automatically detect potential policy-violating apps based on app relations. Experimental results demonstrate that our proposed method successfully identifies a significant portion

of policy-violating apps, 43.75% of which have been subsequently removed by the iOS app store. Furthermore, our method effectively identifies a substantial number of policy-violating apps that have not been removed by the iOS app store.

To boost research in this direction, we will release our dataset to the research community.

2 Related work

In this paper, we consider mobile apps as a special kind of product, which leads us to draw parallels between the app recommendation network and the product co-purchasing network. Therefore, in this section, we introduced some background knowledge of product co-purchasing networks and reviewed relevant studies on network analysis and app recommendations.

2.1 Product co-purchasing networks

The advent of online shopping platforms has facilitated the collection of large datasets on consumer co-purchasing patterns. These patterns are typically represented as networks, where nodes are products and edges indicate that two products have been purchased by the same consumer. Researchers have conducted analyses across various dimensions of these co-purchasing networks [2–9]. Understanding product co-purchasing behavior is crucial for vendors, leading researchers to investigate which products are more likely to be co-purchased and whether these trends can be predicted [3, 5]. Various properties and structures of product co-purchasing networks have been analyzed, including degree [3], connected components [2], clustering coefficient [4, 5], and motifs [6]. Communities also play a significant role in networks, and researchers have explored different structural definitions of communities [7], overlapping community detection [8], and community detection considering node attributes [9].

Inspired by these studies, we are particularly interested in a special type of product, i.e., mobile apps. Similar to traditional products, mobile apps can be consumed by users. For example, users are able to install, use, and uninstall apps, which is similar to consuming, using, and abandoning products. Given the similarity between apps and products, we seek to apply the above techniques to study the app recommendation network.

2.2 Network analysis

Numerous efforts have focused on analyzing real-world systems from a complex network perspective [1, 2, 10–16]. For example, Barabási et al. [12] analyzed the structure of the World Wide Web. They found that the vertex connectivity follows a scale-free power-law distribution. You et al. [13] analyzed neural networks in deep learning. They mapped neural networks to relational graphs, and found that the performance of neural networks is related to the clustering coefficient and average path length of their relational graphs. Thus, the "sweet spot" of the relational graphs can be identified efficiently. These efforts focused on analyzing different kinds of real-world networks. Few of them have explored app relations from a complex network perspective.

2.3 App recommendations

Limited research has specifically focused on studying app recommendation relationships, but there is a wealth of research on app usage in general [17–21]. For example, Lu et al. [17] proposed PRADO, a model to predict user adoption of Android apps. Baeza-Yates et al. [18] conducted a large-scale study on mobile app usage, extracting multiple features that represent user behaviors, and building a model to predict the next app users are likely to use.

There have also been studies proposing new recommendation systems [1, 19, 20, 22–24]. Chen et al. [20] took advantage of high-order connectivity and neural network collaborative filtering techniques to provide app recommendations for users. Furthermore, a thread of work focused on developing personalized app recommendation systems based on a variety of information. Liu et al. [19] balanced users' interests and their privacy preferences in their app recommendation systems. Peng et al. [23] leveraged both the hierarchical taxonomy of apps and competition among apps with the same category. Liu et al. [24] combined app functionalities and permissions in their recommendation system design.

These efforts mainly focus on app recommendations for users, which is tangential to our work. To the best of our knowledge, the most related work is a preliminary study of app relationships in Google

Play [1]. However, the recommendation relationships in Google Play are based on app similarity. In contrast, we study the iOS store, where user behavior plays a more important role [25]. Thus, we can take a glimpse at how the app recommendation network reflects and affects user behaviors, which is not touched upon by existing studies.

3 Study design

In this section, we first outlined our research questions and then provided details about the dataset used in our study.

3.1 Research questions

Our study is driven by the following research questions:

RQ1. What are the properties of the app recommendation network, and what implications do they have? How does the app recommendation network compare to other real-world complex networks? While product co-purchasing networks have been extensively studied, there is still a lack of knowledge about app recommendation relationships and how these relationships may reflect or affect user behavior patterns.

RQ2. What kinds of apps are more likely to be interconnected in the network? Since each node has different degrees, communities, and neighbors, we are curious as to what kind of nodes have a higher degree (i.e., higher rates of recommendation and exposure), and what kinds of nodes are more likely to connect to each other.

RQ3. Can the structure of the app recommendation network be exploited to assist developers and market maintainers? As the network structures may reflect the behavioral patterns of mobile users, we are interested in exploring how it can be utilized to support developers and market maintainers in their decision-making processes.

3.2 Dataset collection and network construction

To answer the above questions, we first must harvest a comprehensive dataset of apps and their recommendation relationships. In this paper, we focused on the "You Might Also Like" recommendations in the iOS app store. It is worth noting that "You Might Also Like" is a non-personalized recommendation list, which presents the same to all users in the same region. We have tested its consistency via different accounts in the iOS app store. Existing literature [25] has verified that the past user behavior of a large population (i.e., the installation behavior) is a crucial factor of the iOS recommendation engine. This makes it feasible for us to explore the relationships between the app recommendations and user behaviors.

However, obtaining such a dataset is not a trivial task, as it requires gathering a wide range of apps. Thus, we relied on the following strategy. Firstly, we collected a set of seed apps. We started by crawling the daily real-time app ranking lists and the latest-released app lists of the iOS app store in China from January 1, 2020, to March 31, 2021. Based on these apps, we extracted the corresponding app store optimization (ASO) keywords from a data intelligence service[2)] and applied a keyword-based searching approach to harvest more apps related to these keywords. These crawled apps formed the initial set of seed apps. After that, we performed a breadth-first-search using the aforementioned seed apps to collect a complete snapshot of the iOS app store. For each app, we collected its metadata, its recommended apps, and other apps developed by the same developers recursively until no new apps were available. As a result, we obtained a rather comprehensive snapshot of the iOS app store. It is worth noting that we only focus on the apps from the iOS app store in China in this study. Moreover, since the "You Might Also Like" lists in the iOS app store are non-personalized recommendations, the crawler did not need to log in and kept no profile or context during the crawling process. We have tested data acquisition using different devices and user accounts, and the recommendation lists remain consistent.

In total, we collected a dataset consisting of 1346575 apps and 50162114 relationships between these apps. The number of crawled apps is close to the number of total apps monitored by a leading mobile app intelligence company[3)], which implies that our dataset is broadly representative. Finally, we induce the app recommendation network from our data, where each node refers to a specific app, and each edge

2) https://app.chandashi.com.

3) https://www.qimai.cn.

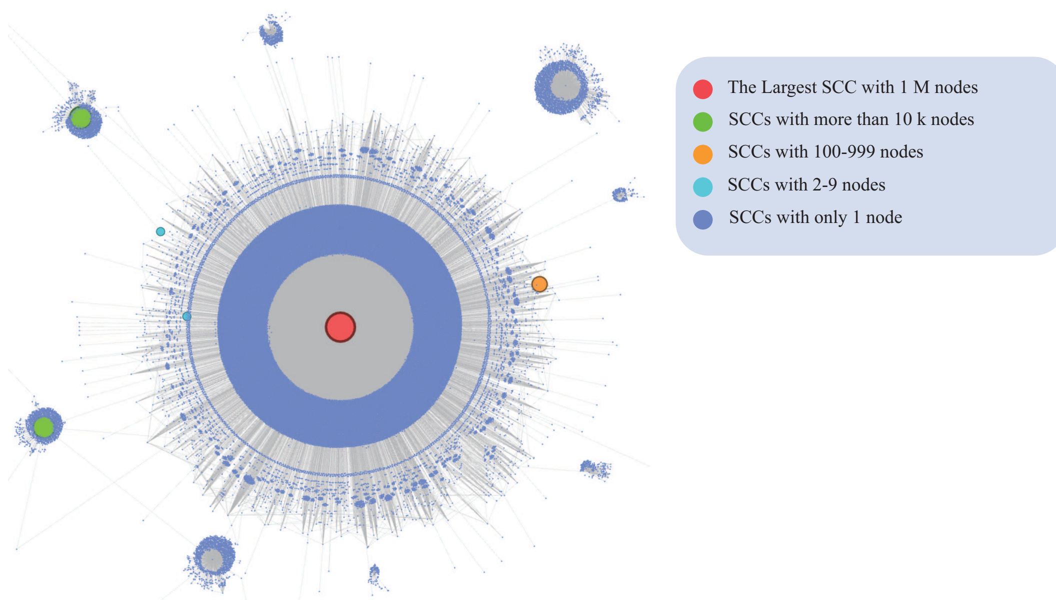

Figure 2 (Color online) Visualization of app recommendation network. Each strongly connected component is merged into one node.

refers to the recommendation relationship. Note that all app recommendation relationships in the iOS app store are directed.

Ethics consideration. All data used in this study is publicly accessible[4)]. Since the "You Might Also Like" recommendation is a non-personalized one, no patterns from individual users are collected.

4 Network properties

In this section, we presented a high-level analysis of the network properties of the constructed app recommendation network, compared it with other real-world complex networks, and discussed the implications of the network properties in the context of app recommendation.

4.1 Overview of the app recommendation network

Table 1 [2, 7, 26–30] summarizes key properties of the app recommendation network. We can observe the presence of giant strongly connected components (SCCs) and giant weakly connected components (WCCs). A strongly connected component is a directed subgraph with a path in each direction between each pair of nodes in the graph. Meanwhile, a weakly connected component is an undirected subgraph with a path between each pair of nodes when all edges are considered undirected [31]. The largest WCC in the app recommendation network covers 90.25% of the nodes, whereas the largest SCC covers 80.08%. This observation suggests that the majority of apps in the iOS app store are well connected. They are not "forgotten" by the store, i.e., these apps have a certain degree of exposure, and thus can be installed by users. To visually illustrate this observation, we provided a visualization of the app recommendation network (see Figure 2). As is shown, the majority of nodes in the network are well connected (see the central red core in Figure 2), which reflects the existence of a giant strongly connected component. Addtionally, there are smaller connected components surrounding the central core, connecting a few nodes in the network.

To explore this further, Figure 3(a) presents the distribution of the node degree. The node degree refers to the number of edges incident to a node. While in-degree is the number of edges that start from a node, out-degree is the number of edges that end with a node [32]. In the context of the app recommendation network, a node with a high out-degree indicates that the app recommends many other apps, whereas a high in-degree indicates that the app is frequently recommended by others. Through

4) For example, one can obtain the complete app recommendation list of Instagram in the iOS app store in China via https://apps.apple.com/cn/app/instagram/id389801252?see-all=customers-also-bought-apps.

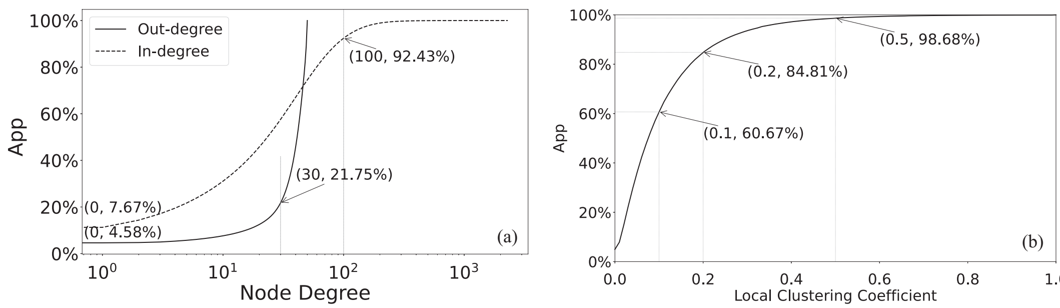

Figure 3 Distribution of node degrees and local clustering coefficients. (a) Node degree; (b) local clustering coefficient.

Figure 3(a), we observed that the iOS app store recommends up to 50 apps for a given app. We also noticed that the distribution of out-degree is left-skewed. More than 78% of nodes have an out-degree greater than 30. This indicates the iOS app store recommends many other apps related to each app. In contrast, the distribution of in-degree is right-skewed, which indicates that most apps in the iOS app store receive only a few recommendations. Only 7.56% of nodes have an in-degree greater than 100. This suggests that a small set of elite apps receive considerable recommendations and exposure in the iOS app store. It is worth noting that there are 4.58% of nodes whose out-degree equals 0, and 7.67% of nodes whose in-degree equals 0. Most of these apps have been released for a long time with hardly any user ratings or comments. This suggests that app quality might be a reason for their lack of exposure.

Apart from the previous observations, we also found that bidirectional edges account for a significant proportion of edges (40.79%) in the network. A bidirectional edge exists between two nodes when they connect with each other in both directions in a directed graph. Therefore, our observation suggests that many recommendations in the app recommendation network are reciprocal. It is worth noting that the iOS recommendation network, unlike the Play Store, is partly driven by co-installation patterns [1, 25]. Hence, this observation suggests that these apps are likely to be commonly installed by the same user. To better identify these clusters of co-installation, we calculate the local clustering coefficient. The local clustering coefficient quantifies the proximity of its neighbors to forming a clique [33]. It equals the proportion of the number of edges between nodes within its neighborhood divided by the number of edges that could potentially exist between them. Figure 3(b) depicts the local clustering coefficient of each node in the app recommendation network. Contrary to expectations, we see that the average is only 0.11. This suggests that there are only a few clearly defined clusters of apps in the network.

4.2 Comparison with other networks

Table 1 includes a comparison of the app recommendation network with several other real-world complex networks. We observed that the app recommendation network exhibits similarities to these networks. Most of them have a giant weakly connected component and a giant strongly connected component, while the average path length is within 20. Considering that the average path length of a network refers to the average number of steps along the shortest paths between all possible pairs of network nodes [32], our observation indicates that nodes in these real-world networks are well connected, and demonstrate a small-world effect [33]. In addition, the clustering coefficients of these networks are around 0.15. This suggests that these real-world networks have clearly defined clusters instead of giant ones. However, there are also differences between the app recommendation network and other real-world networks. The app recommendation network has the largest average degree among these networks. Surprisingly, the average degree of the app recommendation network is 4.4× that of the Amazon network. This suggests that mobile apps, as a special type of product, interact (i.e., recommend) more frequently. Additionally, the app recommendation network has the largest network diameter and the second-largest average path length. The diameter of the app recommendation network is 1.8× that of the Amazon network. Since network diameter refers to the shortest distance between the two most distant nodes in the network, a larger diameter suggests that although mobile apps interact more frequently, it is harder for users to navigate from one specific app to another via recommendations.

Answer to RQ1: As a real-world complex network, the app recommendation network exhibits both similarities and differences in comparison to other networks, which holds significant implications for app recommendations. The presence of a giant connected component indicates that most apps in the store

Table 1 Basic statistics of some real-world networks. Deg refers to average degree, C refers to clustering coefficient, and L refers to average path length.

Network	# Nodes	# Edges	Deg	SCC (%)	WCC (%)	C	Diameter	L
App recommendation network	1346575	50162114	74.50	80.07	90.26	0.11	37	14
Amazon product network [2]	403394	3387388	16.79	97.98	100	0.19	21	–
E-mail network [26]	59812	86300	2.88	95.25	–	0.003	–	4.95
E-mail address network [27]	16881	57029	6.76	–	–	0.17	–	5.22
WWW (nd.edu) [28]	325729	1469680	9.02	–	–	–	–	11.20
WWW (Altavista) [29]	203549046	1466 million	14.40	27.74	91.76	–	28 (SCC)	16.18
LiveJournal [7,30]	4847571	68993773	28.47	78.98	99.93	0.13	16	–

have gained a certain amount of exposure, while the small clustering coefficient suggests that apps often recommend a wide range of other apps rather than being limited to fixed recommendations. Therefore, we conclude that most apps in the iOS app store acquire exposure through recommendations. Moreover, the large average degree suggests that the iOS app store establishes numerous connections among apps, thereby expanding user choices during app store browsing. Additionally, the large diameter and average path length indicate that the app recommendation mechanism in the iOS app store tends expose users to a variety of apps.

5 Demystifying app recommendation

Our previous network analysis suggests that nodes in the app recommendation network are well connected (i.e., the existence of a giant strongly connected component), and there exist tight relations between connected nodes (i.e., bidirectional edges). This motivates us to delve into the characteristics of frequently recommended apps (i.e., important nodes and edges in the complex network).

5.1 Recommendation and exposure frequency

We first look at factors that may relate to the number of recommendations an app receives (in-degree).

5.1.1 *App category*

We calculate the number of apps whose in-degree ranks in the top-10k in each category. Interestingly, we observed that apps in `Business`, `Utilities`, `Education`, and `Game` account for the majority. These particular categories demonstrate higher rates of recommendations and exposure. We speculated that apps in these categories are therefore more popular among users. On the other hand, there exist quite a few rarely recommended apps, i.e., zero in-degree apps. The proportion of zero in-degree apps is extremely high in `Developer Tools` and `Graphics & Design` (over 30%). This suggests that most apps in these categories have fewer recommendations and exposure. These apps cater for certain users, and have a smaller user base. This might be the reason why they are less likely to be recommended.

5.1.2 *Developers*

Next, we focused on developers. Figure 4(a) presents the sum of in-degree for all apps built by each developer. It can be observed that there are about 80% of developers who have a sum in-degree below 100, while the sum in-degree of about 4% of developers is even 0. In contrast, there are only 0.86% of developers having a sum in-degree over 1000. The results indicate that the rates of app recommendations from the same developers are quite low. The majority of developers and their apps do not get sufficient exposure in the iOS app store, while only a handful of developers and their apps dominate the recommendations.

Furthermore, we analyzed the group of dominant developers, defined as those whose sum in-degree of apps surpasses predefined thresholds. Inspired by the existing literature on app developers [34], we predefined two different thresholds (500 and 1000) to separate developers. Multiple thresholds were selected to distinguish developers with a significant sum in-degree (i.e., within the top 2% of developers) and to avoid random error. We observed that the number of apps from dominant developers is larger than that of other developers under both thresholds (see Figure 4(b)). When the threshold was set at 1000, the average in-degree of apps from dominant developers was 78.62, whereas for other developers, it was 37.02. Similarly, when the threshold was set at 500, the average in-degree was 85.85 compared to

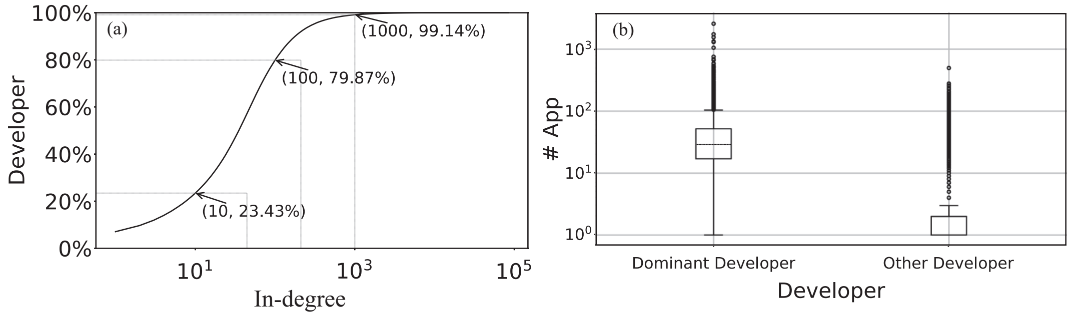

Figure 4 Relations between nodes (apps) and developers. (a) In-degree distribution; (b) app number.

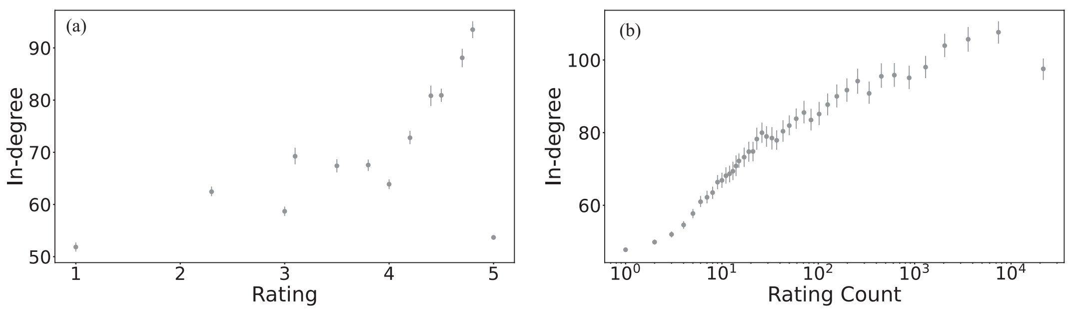

Figure 5 Relations between node in-degree and ratings. (a) User ratings; (b) rating counts.

36.18. The results indicate that dominant developers tend to create more apps in the iOS app store, and their apps have larger recommendation and exposure rates.

5.1.3 *User ratings*

User ratings serve as a proxy for the general quality of an app. Users can give each app a rating from 1 to 5 stars. A high-quality app is more likely to have a high user rating. Figure 5(a) shows the relationship between in-degree and user ratings. We observed that highly-rated apps tend to have higher in-degrees. Surprisingly, however, when the user rating equals 5 stars, the in-degree decreases sharply. This indicates that, apart from 5-star apps, highly rated apps are more likely to receive more recommendations and exposure. For context, we inspected 5-star apps. There are 106997 5-star apps in total. The majority of them have a relatively small number of ratings (90.90% have fewer than 10 ratings). As for the remaining 5-star apps, we speculate that they might manipulate their ratings, since it is almost impossible for all users to be fully satisfied with an app. The number of ratings and manipulations might be the reason for the last outlier in Figure 5(a).

The number of ratings submitted further reflects the popularity of an app. In general, a frequently downloaded app is more likely to have a larger number of ratings. As shown in Figure 5(b), apps with a large number of ratings tend to have a large in-degree. This suggests that popular apps tend to have a higher recommendation rate, and get more exposure in the app store. Notably, we also observed that there exists a drop-off when the count hits above 10000. Upon manual inspection of the 1389 apps with a large number of ratings (above 10000) and low in-degree (below 50), the majority of these apps are already popular and well-known, including apps like Douyin and WeChat. 914 apps (65.8%) rank in the top 200 in their categories. It is possible that the iOS app store reduces the exposure of these popular apps as they have already gained sufficient exposure.

5.1.4 *App maintenance*

Lastly, we examined app maintenance, which includes app age and app updates. App age refers to the number of days since the app was released in the store. Figure 6(a) presents the relationship between in-degree and app age, showing an increase in in-degrees as the app age increments. Although the correlation is weak (Pearson correlation coefficient of 0.23), it is noteworthy that app age plays a role in app recommendations, in contrast to the World Wide Web where no correlation exists between site age and the number of links [11].

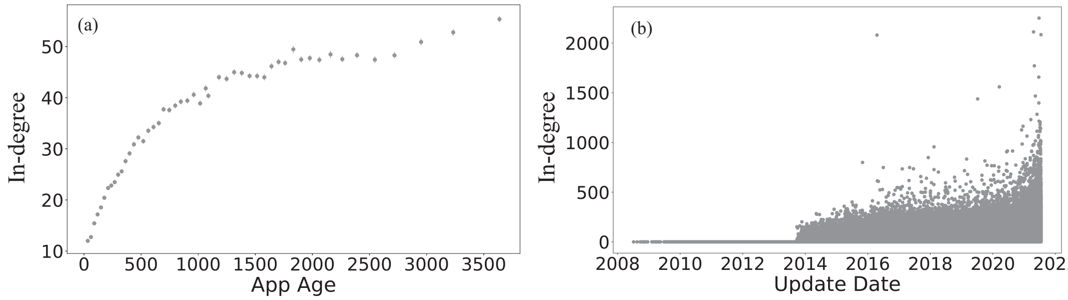

Figure 6 Relations between node in-degree and app life-cycle. (a) App age; (b) update date.

Table 2 Dominant app categories in the communities of app recommendation network.

Community ID	App category
1	`Weather`, `News`, `Social Networking`, `Sports`, `Shopping`, `Food & Drink`, `Lifestyle`
2	`Graphics & Design`, `Reference`, `Photo & Video`, `Medical`, `Entertainment`, `Productivity`, `Books`, `Games`, `Finance`, `Health & Fitness`, `Utilities`, `Business`, `Education`, `Music`
3	`Navigation`, `Travel`
4	`Stickers`
5	`Magazines & Newspapers`
6	`Developer Tools`
7	`Social Networking`

App updates refer to the recent update date of apps. Figure 6(b) presents the relation between in-degree and app updates. We observed that apps with a large in-degree tend to have released updates recently. Specifically, for apps whose in-degree is larger than 500, 73.56% of them were updated in 2021. The results indicate that apps with better maintenance and regular updates are more likely to gain increased exposure and recommendations.

5.2 Bidirectional recommendation frequency

In this subsection, we examined the frequency of bidirectional recommendations, focusing on the types of apps that are frequently recommended together.

5.2.1 *App category*

We started by focusing on the category of each app. We would like to figure out which categories of apps are more likely to recommend each other. To answer this question, we analyzed the community of the network. Communities refer to a subset of nodes that are densely connected internally [35]. Since the app recommendation network is a giant complex network, we used Infomap [36], a tool dedicated to complex networks, to perform community detection. As a result, we obtained 7 different communities (containing 871229, 344163, 64038, 13033, 1485, 713, and 2 nodes, respectively).

We referred to app categories that occur most frequently within each community as the dominant categories. Table 2 presents the dominant app categories in each community. Apart from some outliers, nodes in the same categories are clustered into the same community, indicating that apps from the same category frequently recommend each other. In addition, we observed that apps of certain categories are clustered into the same community, while apps of a certain category form a separate community. There are three communities composed of apps from different categories and four communities consisting of apps within a single category.

Community 1 encompasses seven different app categories, primarily related to users' daily basic needs, such as food and weather. This suggests that apps fulfilling basic needs are closely connected in the network and are more likely to recommend each other. In contrast, Community 2 includes app categories associated with more advanced needs. These app categories are more task-specific, such as `Photo & Video`, indicating that apps fulfilling advanced needs are closely connected and distinct from apps fulfilling basic needs. Community 3 comprises apps in `Navigation` and `Travel`, possibly due to their complementary functions. For example, users may use travel-related apps while making plans and simultaneously rely on navigation apps. The remaining communities consist of apps from a single category.

Table 3 Average proportion of high-score apps.

Condition	High-score apps (%)	Other apps (%)
$M = 4.0$, $N = 100$	50.83	1.30
$M = 4.5$, $N = 100$	44.23	1.19

Table 4 Average proportion of popular apps.

Condition	Popular apps (%)	Other apps (%)
Threshold = 1000	51.70	0.60
Threshold = 10000	34.90	0.21

5.2.2 *Developers*

We further investigated the developer of each app. We took the advantage of the developer identifier to distinguish them. Among all bidirectionally connected app pairs, only 150 pairs are created by the same developer. These apps belong to 19 different developers. It is worth noting that the majority of apps created by the same developer do not have bidirectional edges with each other. This indicates that apps created by the same developer are less likely to be recommended together.

5.2.3 *User ratings*

Next, we revisited the user ratings of each app. We would like to discover if highly rated apps are frequently recommended together. To answer this question, we separated apps into groups. Since some apps have a limited number of ratings in the iOS app store, their overall scores may not be representative. Thus, in our analysis, we considered apps that are rated higher than M stars and have more than N ratings as high-score apps. Considering the inflection points of ratings among most Android app markets found in existing literature [37], we flagged highly rated apps under two different conditions: (1) M equals 4.0 and N equals 100; and (2) M equals 4.5 and N equals 100. This results in 30952 and 25011 highly rated apps, respectively, under the two conditions. After filtering high-score apps, we calculated the proportion of high-score apps among the bidirectionally connected neighbors for each app. Table 3 presents the results. We observed that under both conditions, the average proportion of highly rated apps in the bidirectionally connected neighbors of high-score apps is much larger than that of other apps. This confirms our expectation that high-score apps are frequently recommended together to users.

5.2.4 *App popularity*

We shifted our focus to the popularity of each app. Since the iOS app store does not provide the number of downloads per-app, we utilized rating counts as a proxy for popularity. Again, following the observations of previous research efforts [37, 38], we set two different thresholds to identify popular apps (i.e., within the top 1% and top 0.3%), corresponding to 1000 and 10000 rating counts, respectively. Apps with ratings higher than the threshold are considered popular. After filtering popular apps, we calculated the proportion of popular apps among the bidirectionally connected neighbors of each app within different groups. Table 4 shows the results. When the threshold was set at 10000, the average proportion of bidirectionally connected popular apps was 34.90% for popular apps, whereas 0.21% for other apps. When we further lowered the threshold to 1000, the average proportion was 51.70% and 0.60%. This suggests that popular apps are more likely to be bidirectionally connected in the app recommendation network, which may cause a typical pattern that popular apps are more likely to be consumed by users.

5.2.5 *App functionality*

Finally, we examined the functionality of each app to explore whether apps with similar functions are more likely to be recommended together to users. Generally, developers will introduce the details about their app functions in the app descriptions. Thus, we measured the similarities of descriptions between different apps. We started by calculating the word embeddings of the app descriptions. We took advantage of SentenceTransformers [39] to obtain the embeddings. Since there exist different languages in the app description text, we chose one of the multilingual models, `paraphrase-multilingual-MiniLM-L12-v2` [40], provided by SentenceTransformers. We followed standard procedures to preprocess the raw app description text. After that, we fed the preprocessed text to the pre-trained model. As a result, we obtained a 384-dimension embedding for the description of each app. Once we obtained the embeddings, we were

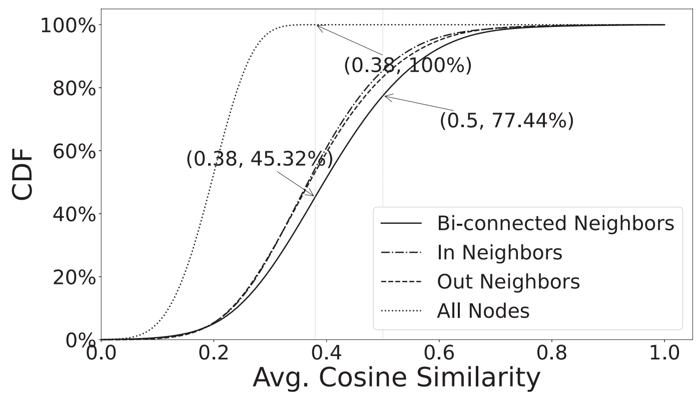

Figure 7 Average cosine similarity of app descriptions.

able to calculate the cosine similarities between different apps. A higher cosine similarity indicates that the corresponding apps have more similar functions.

Figure 7 presents the distribution of average cosine similarity of descriptions between different apps. For each mobile app, we calculated the cosine similarity between the description of a given app and the descriptions of all other apps, and took the average as the baseline. As is shown, apps will always have the highest similarity of descriptions with their bidirectional connected neighbors, while the similarity with their one-way neighbors is lower. However, the description similarity between apps and their neighbors is consistently higher than the baseline. On average, the cosine similarity among descriptions of different apps is 0.19. In comparison, there are more than 95% of all apps, whose average cosine similarities with bi-connected neighbors are higher than 0.19, while there are more than 50% of all apps, whose average cosine similarities with bi-connected neighbors are higher than 0.38 (twice the average cosine similarity among the descriptions of different apps). Further, over 20% of all apps are higher than 0.5. This indicates that apps with similar functions are more likely to be recommended together to users. The reason behind this tendency may be that, although these apps provide similar functionality, they still offer exclusive content that sets them apart. We will give a detailed case study in Subsection 6.1. We argue that this might be an effective methodology for enhancing recommendations, allowing users to fine-tune the metrics used (e.g., preferring similar apps vs. apps that have been co-installed by others). Besides, developers can utilize function (or description) similarity to establish connections with popular apps.

Answer to RQ2: Through our analysis of the meta information of each app, we have identified several factors that contribute to gaining more recommendations. Specifically, apps belonging to specific categories, highly-rated apps, and well-maintained apps tend to receive more recommendations. We also observe homophily, where complimentary apps, highly-rated apps, popular apps, and apps that provide similar functions are more likely to be recommended together. These findings provide valuable insights for developers. Developers should focus on app categories and prioritize maintenance efforts to increase their app's exposure and likelihood of receiving recommendations. Furthermore, tailoring app descriptions can effectively stimulate more recommendations, as we have found that similarity serves as a reliable indicator for the presence of connections between apps.

6 Application of the app recommendation network

The preceding analysis served as a motivation for further exploration of common patterns within app recommendations. Through the analysis of common motif structures, we strived to measure archetypal patterns of recommendations (see Subsection 6.1) and how these can be used to identify policy-violating apps (Subsection 6.2).

6.1 Motif analysis

Network motifs are recurring patterns of interconnections that appear significantly more times in a given network than in a randomized network [41]. Motif analysis has become an important method to uncover the local structures of complex networks. In this subsection, we employed network motifs to understand

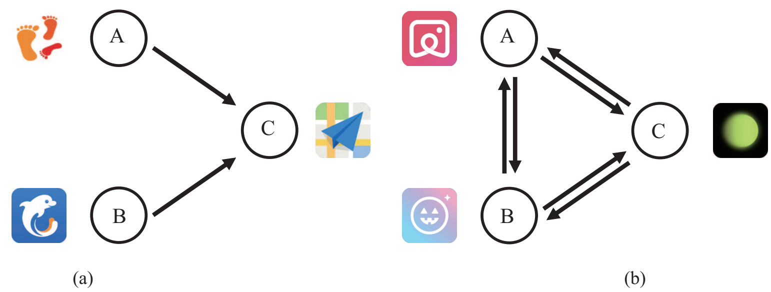

Figure 8 (Color online) Selected motifs in app recommendation network. (a) Motif 1; (b) motif 2.

the prevalent recommendation patterns of apps. There are 13 different types of 3-node subgraphs in total. 11 of them are motifs. For further analysis, we selected two motifs (see Figure 8). We believed that these motifs can help developers figure out different user patterns, and provide more personalized services.

6.1.1 *Motif 1*

The most frequently occurring motif is shown in Figure 8(a). The motif suggests that there is a subset of apps that are more likely to be recommended to users compared to others. For instance, different users may browse different app pages (nodes A and B), but they are still likely to encounter the same recommendations (node C). About 68% of apps belong to this subset in the app recommendation network. Commonly, the three apps belong to the same category, accounting for 79.49% of all motifs. Another prevalent pattern is that two apps from the same category recommend another app from a different category (5.73% of all motifs). For example, this might be two `Travel` apps recommending a `Navigation` app, or two `Health & Fitness` apps recommending a `Medical` app. The results can reflect two primary user patterns. On the one hand, there are clearly certain dominant apps in each category. Through more exposure given by the app store, users are more likely to view and even install these apps. On the other hand, apps in some categories possess functionality complementary to apps in other categories. Overall, we find that apps in `Productivity` and `Business`, `Finance` and `Business`, `Reference` and `Education`, `Medical` and `Health & Fitness`, and `Navigation` and `Travel` are all more likely to be complementary apps. These categories account for over 98% of motifs that align with the second pattern. This pattern implies that the recommendation system of the iOS app store tends to recommend functionally complementary apps together, making it convenient for users to install these apps simultaneously. Motif 1 indicates that a particular group of apps has a higher recommendation probability in the iOS app store. Based on this observation, we suggest that developers can benefit from motif 1 by gaining valuable insights into app development. By studying these highly visible apps, they can learn about their features and functionalities and customize their own apps accordingly.

6.1.2 *Motif 2*

Another important motif worth attention is shown in Figure 8(b). As is shown, all nodes are interconnected, forming a 3-node clique. No matter which app the user is browsing (e.g., node A), the remaining two apps (nodes B and C) will be recommended to the user. Surprisingly, more than 90% of such motifs consist of apps from the same category. This observation reflects another archetypal pattern of recommendations in the iOS app store, where apps from the same category are more likely to be recommended together and connected intensely. After manual inspection, we found out that the results align with a common phenomenon in the app store, i.e., app complementation. While apps in the same category provide broadly similar functionality, there still exist complementary functions or exclusive contents among certain apps (e.g., multiple social networks). This prompts users to install different apps from the same category, resulting in a higher likelihood of being recommended together by the app store. It is difficult to quantify this exactly, as users may perceive 'complementary' differently. However, Figure 8(b) provides an intuitive example, where three different `Photo & Video` apps form this clique. Although their main function is to help users take photos, there still exist differences between these apps. For example, they can provide users with different kinds of filters, which are attractive to users who enjoy taking photos. Therefore, to facilitate the target audience, the app store will recommend these apps

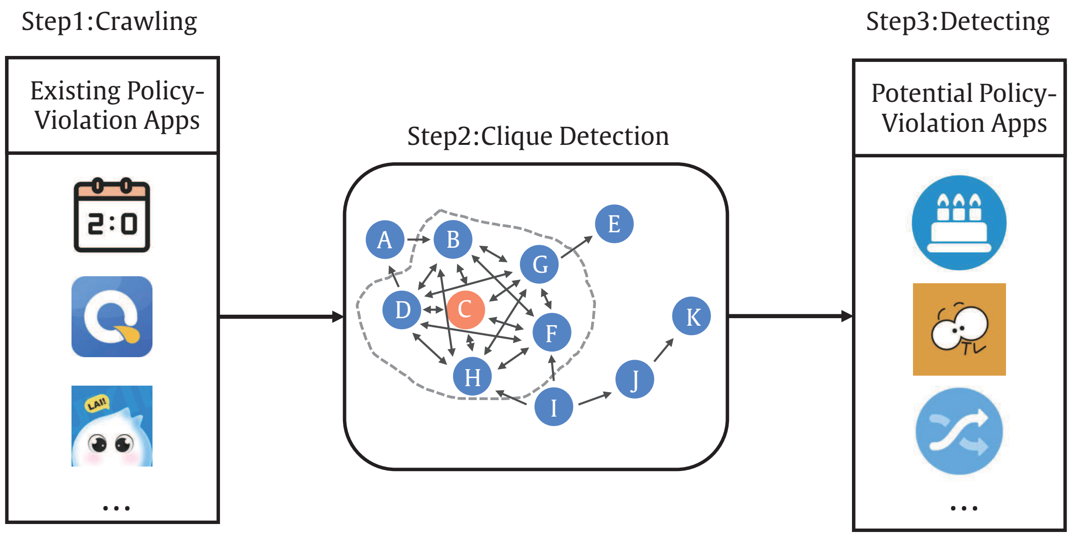

Figure 9 (Color online) Framework of policy-violating app detection method.

together to users. Motif 2 indicates that some apps are connected intensely internally. Their functionalities or contents might be complementary. Considering that policy-violation apps have the characteristics mentioned above (e.g., providing multiple illegal services such as gambling), we speculated that there might be tightly connected cliques among these policy-violation apps, which motivated us to conduct an in-depth analysis (see Subsection 6.2).

6.2 Policy-violating app detection

Previous research efforts [42–44] have shown that app quality remains a major issue across app stores, and that many policy-violating apps exist. These apps abuse a number of ASO techniques to manipulate their metadata, including reviews, descriptions, and ASO keywords, to boost their rankings. It is likely that the recommendation relations are influenced as well. This motivates us to explore whether policy-violating apps influence the recommendation relations and how they can be detected.

6.2.1 *Motivating example*

We started by providing a motivating example. `ZQ Counter`, a policy-violating app that has been removed from the iOS app store, served as our focal point. We identified the potential removal reason based on the taxonomy of previous work [43], and found that `ZQ Counter` was removed because of its fake description (it is a video app, which contradicts its description). By examining the maximum clique containing `ZQ Counter`, we observed that 5 out of the 13 apps in the clique have also been removed by the iOS app store. For further exploration, we checked the metadata of these apps. We found that most of the apps in the clique have similar functionality to `ZQ Counter`. They are actually video apps, but disguise themselves as other types of apps. While some policy-violating apps have already been removed, not all of them are detected by the current app vetting process. The example indicates that policy-violating apps tend to interconnect in the app recommendation network. However, the current app vetting process is not able to identify all policy-violating apps. We argue that leveraging the network structure could assist market maintainers in detecting potential policy-violating apps.

6.2.2 *Policy-violating detection*

Based on the insights gained from the motivating example, we proposed a method for automatically detecting policy-violating apps. Figure 9 presents the framework of our proposed method. We start by collecting the existing policy-violating apps in the iOS app store as seed apps. After that, we extract the maximum cliques for each seed app. The occurrence of a specific app in the maximum cliques formed by the existing policy-violating apps serves as an indicator of potential policy violation. To evaluate the effectiveness of our proposed method, we manually collected the removed apps in the iOS app store from July 19, 2021 to July 31, 2021 (13 days), monitored by a third-party app intelligence company[5]. A total of 17463 apps were removed during this period, which we considered as proxies for existing policy-violating apps. Based on these seed apps, our proposed method successfully labeled 33404 new policy-violating apps, which accounts for 43.75% of apps in the maximum cliques. These policy-violating apps are afterward removed by the iOS app store.

To explore the possibility of potential policy-violating apps that have not been removed by the iOS app store, we manually inspected the apps detected by our proposed method. Specifically, we examined apps identified by our proposed method based on the removed apps on July 19, 2021. Based on 913 seed apps, our proposed method labeled 655 potential policy-violation apps. Out of these apps, 276 were later removed, while 243 apps (37.10%) clearly violated the app store policy. Among these apps, 224 have quality issues such as app crushes, 13 apps manipulate user reviews to boost their ranking, 5 apps deceive users into purchasing premium services, and one provides an app description inconsistent with its functions. Considering both the already removed apps and those still present, our method detects 79.24% of policy-violation apps. A considerable proportion of them have not been noticed and removed by the app store. The results further prove the effectiveness of our proposed method. We suggest that market maintainers should pay more attention to apps that have tight relationships with removed apps (i.e., apps in the cliques of removed apps).

Answer to RQ3: The local structure of the app recommendation network provides valuable insights for various tasks. Developers can identify different behavior patterns through 3-node motifs. Additionally, we propose a method to automatically detect potential policy-violating apps based on cliques. Experimental results verify that market maintainers should pay more attention to cliques formed by existing policy-violating apps.

7 Discussion

In this section, we discussed the findings across platforms, insights and implications of the study, as well as the limitations.

7.1 iOS app store vs. Google Play

Our study focused on the iOS app store, one of the most well-known app markets. A previous study [1] has explored the network of Google Play, known as AppNet. It is important to note that the app recommendation network and AppNet exhibit distinct behaviors. The largest strongly connected component of AppNet forms a small central core, which contains 7.07% of all nodes, and 97% of all nodes. In contrast, the largest strongly connected component of the app recommendation network contains 80.08% of all nodes and 84.39% of all edges. Most nodes in the app recommendation network are well connected, and its central core is significantly larger. Both recommendation mechanisms have their own advantages and drawbacks. Google Play tends to recommend the most similar apps to users [1], aligning with their current functional requirements. However, most apps cannot be reached by users through the app recommendation by Google Play, since the majority of apps are never recommended by Google Play. Users can discover these apps only by searching their names. In contrast, the iOS app store exposes most apps through recommendations. Therefore, users can reach the majority of apps in the iOS app store through app recommendations. However, considering the diversity of apps that are recommended, users are more likely to explore the app store, rather than look for apps that fulfill their current demand.

7.2 Implications

Our study reveals novel findings about the app recommendation network. These are useful for developers, market maintainers, and researchers. We summarize the implications as follows.

First, through the analysis of key properties of the app recommendation network, our study presented the similarities and differences between the app recommendation network and other real-world networks. We observed that the app recommendation network can be regarded as a complex network, whereas it owns some unique characteristics. Our findings can serve as a valuable addition to existing research on complex network analysis, which sheds light on applying network theory to app analysis.

5) https://app.chandashi.com.

Second, by analyzing the relations between node connectivity and the corresponding app meta-data, our study demystified what kinds of apps receive higher exposure in the market, and what kinds of apps are frequently recommended together. The results are beneficial for different developers in multiple tasks. For new developers, we suggest that they should perhaps focus on specific hot categories (e.g., `Business`), since apps in these categories get more exposure. For developers who would like to enhance their app's visibility, they should maintain their apps regularly to keep them highly rated. Developers could also try to establish connections with popular apps through learning from these apps (e.g., providing similar functionalities or tailoring descriptions).

Third, with the help of motif analysis, our study showed that the local structures of the network are related to the behavioral patterns of users. Users are frequently exposed to a subset of apps, and their usage behaviors will affect the recommendation mechanism. Developers are able to infer user behavior patterns through different motifs and apply corresponding strategies to promote their apps.

Finally, through clique detection, we proposed a novel method that can efficiently detect potential policy-violating apps. Given the experimental results, we suggest that market maintainers should pay more attention to the cliques formed by policy-violating apps. Our observations show that there are many potential policy-violating apps in the cliques formed by removed apps. Some of these apps have been removed by the app store, while a considerable amount of them have not. These apps disguise themselves as normal ones. Our findings and proposed method shed light on the detection of such apps.

7.3 Limitations

Our study is subject to several limitations. First, we do not have full vantage on the specifics of the recommendation algorithm. Thus, the results may be impacted by noise from other considerations in the recommendation engine. Second, there is an artificial limit on the number of apps the iOS app store will recommend. Thus, some recommendation relationships may be missing. Lastly, due to the absence of time-series data, our analysis primarily focuses on the correlation of variables instead of causation. Nevertheless, our results provide insights into overall trends.

8 Conclusion

In this paper, we characterized the app recommendation relationships in the iOS app store from the complex network perspective. Our study covered 1.3 million mobile apps and 50 million app relations. Overall, our analysis suggests that the app recommendation network is a complex network, which shares key similarities and differences with other real-world complex networks. The properties of the network have interesting implications in the context of app recommendation. We further analyzed what kinds of apps gain more exposure and what kinds of apps are more likely to be recommended together. Our observations show that apps from specific categories, highly-rated apps, and better-maintained apps receive more exposure, while popular apps, highly-rated apps, apps with similar functionality, and apps that fulfill specific user demands are recommended together frequently. Finally, we revealed the usage scenarios of the network. The local structures of the network can be used to understand user behavior patterns, and detect potential policy-violating apps. We firmly believe our research efforts and insights can positively contribute to the mobile app ecosystem and shed light on app relationship analysis.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 62325201, 62102009), Beijing Outstanding Young Scientist Program (Grant No. BJJWZYJH01201910001004), and Center for Data Space Technology and System, Peking University. The work of Haoyu WANG was supported in part by National Natural Science Foundation of China (Grant No. 62072046).

References

1 Guo Q, Wang H Y, Zhang C W, et al. AppNet: understanding app recommendation in Google Play. In: Proceedings of the 3rd ACM SIGSOFT International Workshop on App Market Analytics, 2019. 19–25
2 Leskovec J, Adamic L A, Huberman B A. The dynamics of viral marketing. ACM Trans Web, 2007, 1: 5
3 Basuchowdhuri P, Shekhawat M K, Saha S K. Analysis of product purchase patterns in a co-purchase network. In: Proceedings of the 4th International Conference of Emerging Applications of Information Technology, 2014. 355–360
4 Ghosh S, Mitra A, Basuchowdhuri P, et al. Analysis of online product purchase and predicting items for co-purchase. In: Proceedings of the 3rd International Conference on Advanced Computing, Networking and Informatics, 2016. 581–591
5 Prasad U, Kumari N, Ganguly N, et al. Analysis of the co-purchase network of products to predict amazon sales-rank. In: Proceedings of the International Conference on Big Data Analytics, 2017. 197–214
6 Srivastava A. Motif analysis in the Amazon product co-purchasing network. 2010. ArXiv:1012.4050
7 Leskovec J, Lang K J, Dasgupta A, et al. Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. Internet Math, 2009, 6: 29–123

8 Jebabli M, Cherifi H, Cherifi C, et al. Overlapping community detection versus ground-truth in amazon co-purchasing network. In: Proceedings of the 11th International Conference on Signal-Image Technology and Internet-Based Systems, 2015. 328–336
9 Yamazaki T, Shimizu N, Kobayashi H, et al. Weighted micro-clustering: application to community detection in large-scale co-purchasing networks with user attributes. In: Proceedings of the 25th International Conference Companion on World Wide Web, 2016. 131–132
10 Albert R, Jeong H, Barabási A L. Diameter of the World-Wide Web. Nature, 1999, 401: 130–131
11 Adamic L A, Huberman B A. Power-law distribution of the World Wide Web. Science, 2000, 287: 2115
12 Barabási A L, Albert R. Emergence of scaling in random networks. Science, 1999, 286: 509–512
13 You J, Leskovec J, He K, et al. Graph structure of neural networks. In: Proceedings of the International Conference on Machine Learning, 2020. 10881–10891
14 Xu R F, Du J C, Zhao Z S, et al. Inferring user profiles in social media by joint modeling of text and networks. Sci China Inf Sci, 2019, 62: 219104
15 Song G J, Li Y H, Wang J S, et al. Inferring explicit and implicit social ties simultaneously in mobile social networks. Sci China Inf Sci, 2020, 63: 149101
16 Liu H J, Cao J D, Huang W, et al. Complex network approach for the evaluation of asphalt pavement design and construction: a longitudinal study. Sci China Inf Sci, 2022, 65: 172204
17 Lu X, Chen Z, Liu X, et al. PRADO: predicting app adoption by learning the correlation between developer-controllable properties and user behaviors. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2017. 1–30
18 Baeza-Yates R, Jiang D, Silvestri F, et al. Predicting the next app that you are going to use. In: Proceedings of the 8th ACM International Conference on Web Search and Data Mining, 2015. 285–294
19 Liu B, Kong D, Cen L, et al. Personalized mobile app recommendation: reconciling app functionality and user privacy preference. In: Proceedings of the 8th ACM International Conference on Web Search and Data Mining, 2015. 315–324
20 Chen J, Cao B, Liu J, et al. MR-UI: a mobile application recommendation based on user interaction. In: Proceedings of the IEEE International Conference on Web Services, 2020. 134–141
21 Pu C, Wu Z, Chen H, et al. A sequential recommendation for mobile apps: what will user click next app? In: Proceedings of the 2018 IEEE International Conference on Web Services, 2018. 243–248
22 Liu X Z, Huang G, Zhao Q, et al. iMashup: a mashup-based framework for service composition. Sci China Inf Sci, 2014, 57: 012101
23 Peng M, Zeng G, Sun Z, et al. Personalized app recommendation based on app permissions. World Wide Web, 2018, 21: 89–104
24 Liu B, Wu Y, Gong N Z, et al. Structural analysis of user choices for mobile app recommendation. ACM Trans Knowl Discov Data, 2017, 11: 1–23
25 Hughes J. IPhone and IPad Apps Marketing: Secrets to Selling Your IPhone and IPad Apps. New York: Que Publishing, 2010
26 Ebel H, Mielsch L I, Bornholdt S. Scale-free topology of e-mail networks. Phys Rev E, 2002, 66: 035103
27 Newman M E J, Forrest S, Balthrop J. Email networks and the spread of computer viruses. Phys Rev E, 2002, 66: 035101
28 Barabási A L, Albert R, Jeong H. Scale-free characteristics of random networks: the topology of the world-wide web. Phys A-Stat Mech Its Appl, 2000, 281: 69–77
29 Broder A, Kumar R, Maghoul F, et al. Graph structure in the Web. Comput Networks, 2000, 33: 309–320
30 Backstrom L, Huttenlocher D, Kleinberg J, et al. Group formation in large social networks: membership, growth, and evolution. In: Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining, 2006. 44–54
31 Clark J, Holton D A. A First Look at Graph Theory. Singapore: World Scientific, 1991
32 Tutte W T. Graph Theory. Cambridge: Cambridge University Press, 2001
33 Watts D J, Strogatz S H. Collective dynamics of 'small-world' networks. Nature, 1998, 393: 440–442
34 Wang H, Liu Z, Guo Y, et al. An explorative study of the mobile app ecosystem from app developers' perspective. In: Proceedings of the 26th International Conference on World Wide Web. 2017. 163–172
35 Girvan M, Newman M E J. Community structure in social and biological networks. Proc Natl Acad Sci USA, 2002, 99: 7821–7826
36 Rosvall M, Bergstrom C T. Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci USA, 2008, 105: 1118–1123
37 Wang H, Liu Z, Liang J, et al. Beyond Google Play: a large-scale comparative study of Chinese android app markets. In: Proceedings of the Internet Measurement Conference 2018. 2018. 293–307
38 Wang H, Li H, Guo Y. Understanding the evolution of mobile app ecosystems: a longitudinal measurement study of Google Play. In: Proceedings of the World Wide Web Conference, 2019. 1988–1999
39 Reimers N, Gurevych I, Reimers N, et al. Sentence-BERT: sentence embeddings using siamese BERT-networks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2019. 671–688
40 Reimers N, Gurevych I. Making monolingual sentence embeddings multilingual using knowledge distillation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2020. 4512–4525
41 Milo R, Shen-Orr S, Itzkovitz S, et al. Network motifs: simple building blocks of complex networks. Science, 2002, 298: 824–827
42 Wang H, Si J, Li H, et al. RmvDroid: towards a reliable Android malware dataset with app metadata. In: Proceedings of the 16th International Conference on Mining Software Repositories, 2019. 404–408
43 Lin F, Wang H, Wang L, et al. A longitudinal study of removed apps in ios app store. In: Proceedings of the Web Conference 2021, 2021. 1435–1446
44 Li Z T, Li W L, Lin F Y, et al. Hybrid malware detection approach with feedback-directed machine learning. Sci China Inf Sci, 2020, 63: 139103