Understanding the Long-term Dynamics of Mobile App Usage Context via Graph Embedding

Yali Fan, Zhen Tu, Tong Li, Hancheng Cao, Tong Xia, Yong Li, *Senior Member, IEEE*, Xiang Chen, *Member, IEEE*, Lin Zhang, *Senior Member, IEEE*

Abstract—With the increasing diversity of mobile apps, users install many apps in their smartphones and often use several apps together to meet a specific requirement. Because of the evolution of user habits and app functions, the set of apps using at the same time, *i.e.*, app usage context, may change over time, which represents the dynamic correlation of different apps and even the evolution trend of the whole app ecosystem. Therefore, understanding how an app's usage context changes over time is very meaningful. In this paper, based on a seven-year app usage dataset, we explore the long-term app usage context dynamics and understand the underlying reasons and influence factors behind. Specifically, we build app co-occurrence graphs in different periods and learn app embeddings accordingly by leveraging graph embedding algorithm. We then measure the change of app usage context by the distance between neighboring app embeddings. As for the whole app ecosystem, we find that the change rate of app usage context undergoes up and down phrases, and varies in different app-categories. Furthermore, we explore three influence factors correlated with such dynamics. These results will be helpful for stakeholders to better understand the evolution of mobile users' app usage behavior.

Index Terms—Mobile apps, app usage context, long-term evolution, graph representation learning.

1 Introduction

With the rapid development of communication technology and smartphones, the number of mobile apps in the app markets has increased a lot in recent years. Taking App Store for example, there were around 900,000 mobile apps in 2013, but the number turned to be 2,000,000 in 2016 [1]. Due to the diversity and fine-grained function of mobile apps, nowadays mobile users install a lot of apps in their phones and often use several apps together to meet a specific requirement. For example, people often use Taobao (a popular e-commerce app) and Alipay (a large online payment app) together when shopping online [2], [3], and users who use Ele.me (an app to order food) usually use Keep (an app focusing on exercise) at a same period [4]. Recent estimates suggest that people use more than 30 apps per month and 10 apps per day on average [5], [6]. What's more, because of the evolution of app markets, app functions and user habits [7], [8], mobile users may change the used apps from time to time, and the set of apps using at the same time may also change over time. For example, as mobile phone photography becomes convenient and high-quality, people like to use photography apps to take photos and then share them through social apps, so these two types apps are often used together. Recently, since Instagram is getting more popular among young people than Facebook in fast photo sharing [9], Instagram has been replacing Facebook to be used with other photography apps to complete photo taking and sharing. Therefore, for a specific app, its **usage context** (i.e., other apps used together) is probably not stable in the long term, which indeed represents the dynamic correlation of different apps and even the evolution trend of the whole app ecosystem. Thus, understanding how app usage context changes over time is very meaningful.

- *Yali Fan and Xiang Chen are with the School of Electronics and Information Technology, Sun Yat-Sen University, Guangzhou, China. E-mail: fanyali@mail2.sysu.edu.cn, chenxiang@mail.sysu.edu.cn. Xiang Chen is the corresponding author.*
- *Lin Zhang is with Tibet University, Lhasa, China, and also with the School of Electronics and Information Technology, Sun Yat-Sen University, Guangzhou, China. E-mail: isszl@mail.sysu.edu.cn.*
- *Zhen Tu, Tong Xia and Yong Li are with the Department of Electrical Engineering, Tsinghua University, Beijing, China. E-mail: cherrytu@tencent.com, xia-t17@mails.tsinghua.edu.cn, liyong07@tsinghua,edu,cn.*
- *Tong Li is with Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China. E-mail: t.li@connect.ust.hk.*
- *Hancheng Cao is with Department of Computer Science, Stanford University, USA. E-mail: hanchcao@stanford.edu.*

However, limited studies pay attention to this important topic of long-term app usage dynamics [10], [11]. On the one hand, most of existing works only concern the static usage context and they are often conducted based on short-term datasets, whose periods are usually shorter than one year or even only a few days [2], [12], [13], [14], [15]. They aim to find out what apps are frequently used together and the mutual impact of these apps. For instance, with a week-long app usage dataset, Huang et al. discovered frequent app usage sets (*e.g.*, Taobao and Alipay) and reported the association within apps belonged to the same category (*e.g.*, News apps promote the use of each other), regardless of time difference [2]. On the other hand, some others conducted studies on understanding short-term dynamics, so they focus on observing the changes of app usage context in one day. For example, based on a four-day dataset, Liu et al. divided one day into seven phases (*e.g.*, morning, noon) and selected an app at a time to watch changes in the apps used together in different time phases [4]. However, such works study the approximately hourly changes of app usage context, so they cannot provide insights about the long-term evolution of app usage behavior and the app ecosystem. Therefore, how the engagement of mobile users and multiple apps changes over time and how an app's usage context change over time remain unexplored up till now.

In this paper, based on a long-term app usage dataset from 2012 to 2018, covering seven-year time spans and containing more than 14 million usage records, we investigate long-term dynamics of app usage context and explore general law behind, aiming to provide a better understanding of app usage long-term behavior and the evolution of the whole app ecosystem [10], [11]. Although this task is meaningful, it also faces many challenges. **First**, the app's usage context is quite complex and changeable. On the one hand, due to higher and higher requirement from mobile app users, apps with various functions continue to appear, so the app's usage context is changeable over a long period of time. On the other hand, even for a short period of time, there are also various other apps co-used with a certain app for a specific requirement, which causes a quite complex usage context. Thus, for an app, how to effectively and completely extract its usage context information from massive app usage behavior becomes a critical problem. **Second**, it is difficult to measure the dynamics of app usage context in different periods. To investigate long-term dynamics of app usage context, it is inevitable to compare the complex usage context information of different periods to obtain evolution results. Therefore, it requires a unified representation format for app usage context in different periods and a quantitative measurement of the usage context dynamics. However, since our research problem is quite different from that of existing works, the above challenges are never considered before, e.g., the integration of dynamic temporal information and the measurement of the app usage context dynamics. To the best of our knowledge, it is the first time to study the long-time dynamics of app usage context.

In order to overcome the above challenges, in this paper we design app embeddings to investigate the long-term dynamics of app usage context. For the first challenge, in order to effectively capture the usage context dynamics over time, we divide the whole long period into several short periods. For each period, by constructing an app graph with co-used relationship (we call it **app co-occurrence graph**), all the complex app usage context information can be represented effectively and completely, clearly showing the usage context for all apps. As for the second challenge, to measure the dynamics of app usage context across different periods, we design a graph representation learning based framework to transfer app usage contexts into quantifiable graph embeddings. Compared to traditional methods (e.g., co-occurrence matrix), graph representation learning is more efficient and effective. On the one hand, its embedding vectors are much low-dimensional and much denser, which is very helpful to enhance the calculation efficiency and reduce the computing resource cost. Moreover, during graph representation leaning, the necessary information about the nodes and relations will be distilled into node embeddings, so that these embeddings can capture more high-order information hidden in the local context (e.g., relationship with multi-hop neighbors). Thus, we first quantify an app's usage context by graph representation learning framework and further obtain the quantified dynamics.

However, it is not easy to design a suitable graph representation learning based framework for our dynamic graphs. Currently, dynamic graph representation learning has emerged as a leading way to distill both structural information and temporal information in dynamic graphs [16], [17]. According to the way of integrating time information into graph embeddings, the existing dynamic graph representation methods can be generally divided into three categories, i.e., Sequence-Model based methods [17], [18], Decomposition based methods [19], and Random Walk based methods [16], [20]. In terms of applications, Sequence-Model based methods and Decomposition based methods are often designed for dynamic graphs with fixed nodes, in other words, nodes are not added or removed, such as social networks. Additionally, they are usually applied to solve supervised or semi-supervised issues (e.g., link prediction and node classification) [16]. However, Random Walk based methods are not limited by these applications. They are more adaptable to handle various types of node observations, like node addition and node deletion, and they perform better on unsupervised tasks (e.g., clustering) due to the inherent advantage of random walks.

In our research, our app graphs are quite changeable due to the fierce competition in app markets. For example, the rate of overlapped nodes between two consecutive years are only about 60%. What's more, our task is unsupervised. Consequently, Random Walk based methods are more suitable for our problem. Following the framework of Random Walk based methods, we design a dynamic graph embedding algorithm for our task. To be specific, we first generate random walks on each snapshot and then input them into an encoder to learn app embeddings (we call it **app usage context embedding** to represent its usage context). Second, to incorporate the effect of temporal dynamics, we use a time regularizer as a smoothness constraint to adjust the app embeddings over consecutive snapshots. Through a series of evaluation in Section 4, we demonstrate that the app embeddings have strong ability to represent the app's usage context. Thus, the quantified measurement of usage context dynamics can be conducted between app embeddings in different time periods. Then we explore general laws of app usage context dynamics by investigating the underlying reasons and influence factors behind, and obtain a series of interesting findings.

To sum up, the main contributions of this work can be summarized as follows:

- We make the first effort to study the dynamics of app usage context with a long-term vision, to better understand how an app's usage context changes over time and even the evolution trend of the whole app ecosystem.
- We design a graph representation learning based framework to transfer app usage context into dynamic graph embeddings based on app co-occurrence graphs, which distills the relationships between apps and models the dynamics of app usage contexts.
- Based on a 7-year app usage dataset, we explore the

long-term app usage contexts dynamics over time and different app-categories, and then conduct an in-depth research on the underlying reasons and influence factors behind.

Among the many insightful results and observations, the following are the most prominent.

- The evolution of app's usage context from 2012 to 2018 undergoes two phases: app's usage context changes more and more drastically from 2012 to 2014, but changes more and more smoothly after 2015. In terms of app categories, Book and Productivity apps change the fastest, while Communication and Social apps have the smallest change rate.
- We find that three factors (i.e., app usage frequency, app contextual diversity, app usage popularity) are related to the usage context changes. More specific, if an app has a high usage frequency, its usage context will change slowly on average. When an app's usage context is quite diverse, its usage context will be robust and also change very slowly. What's more, apps with stable popularity will also have stable usage context.

The rest of the paper is organized as follows. First we introduce the related work in Section 2. Then we introduce our long-term app usage dataset and how we construct app graphs and learn embeddings in Section 3. We provide the visualization analysis of app usage context embeddings in Section 4. In Section 5 we demonstrate how the app usage context changes over time, and we further explore three potential intrinsic influence factors correlated with such dynamics in Section 6. Finally Section 7 concludes the paper.

2 Related Work

2.1 App Usage Analysis

Recently, with the increasing popularization of mobile devices and apps, enormous app usage records have been left on the Internet and such data contains rich information. This has attracted a variety of researchers to be devoted to from various directions, such as app prediction [21], [22], app recommendation [23], [24], app usage pattern discovery [25], [26], app privacy protection [27], [28], app ranking fraud [29], [30]. In fact, many app usage analyses have taken context information (*e.g.*, time, location) into consideration, which makes the analyses more approximate to practical scenarios. The contextual factors can be divided into three broad types, including sensor context, social context and **usage context** [12]. Sensor context is context information that is sensed through the device sensors, such as time [31], [32] and location [33], [34], while social context refers to the social relationship among users [35]. Within our scope, we mainly focus on app usage context, which often represents the apps used together at the same time period.

In order to know more about how users use apps on their mobile devices, lots of studies have payed attention to app usage context to find out what apps are frequently used together and the mutual impact of these apps [2], [4], [12], [13], [14]. Some researchers only concern the static app usage context, which means they consider the app usage dataset (often spanning a few days to one year) as a snapshot of users' app usage behavior and then detect app sets that are frequently used together, regardless of time difference. For example, with a week-long app usage dataset, Huang et al. discovered frequent app usage sets (*e.g.*, Taobao and Alipay) and reported the association within apps belonged to the same category (*e.g.*, News apps promote the use of each other) [2]. Based on half-year app usage records, Tseng et al. firstly found out apps used together and further demonstrated the sequential impact between these apps [13]. However, with the removal of time information, these works are limited to understanding users' time-varying app usage behavior [36].

Some others find that users tend to use different apps at different times of one day, so they focus on observing the changes of app usage context at different time periods. Based on a four-day dataset, Liu et al. divided one day into seven phases (*e.g.*, morning, noon) and selected an app at a time to watch changes in the apps used together in different time phases [4]. However, such works study the approximately hourly changes of app usage context, so they cannot provide insights about the long-term evolution of app usage behavior and the app ecosystem. Different from these existing works, we focus on exploring the dynamics of app usage context with a long-term vision, to better understand how an app's usage context changes over time and even the evolution trend of the whole app ecosystem.

2.2 Long-term Behavior Analysis

A number of studies have focused on long-term human behavior analysis [37], [38], [39], [40], [41], [42], [43]. The aims of most of these works are to observe the changes in people's behavior, and further summarize the latent laws [37], [38], [39] or analyze the reasons behind these changes [40], [41], [42]. The types of these human behavior can be divided into language usage behavior [44], [45], health activity behavior [37], [42], [46], social behavior [40], [43], [47], [48], mobile behavior [49], [50], online behavior [51], [52], [53], [54], [55], etc.

Related to our targeted problem, we mainly focus on previous works dedicated to long-term analysis on app usage behavior [11], [53], [54], [56], [57]. Some researchers just concentrate on the usage behavior of a single app and analyse users' daily habits and preferences [53], [54], [56]. For instance, Lin et al. conducted a 31-month observation of user re-engagement patterns within a health tracking app and demonsrated that the multiple lives paradigm is helpful to users' engagement [56]. Shameli et al. studied walking challenges in a mobile activity tracking app over a period of one year and observed that the competitions during walking can significantly increase the users' participation in physical activities [53]. While some others focus on user interaction with multi apps and provide overall view of the evolution of the app ecosystem [11], [57]. For example, Wang et al. conducted a comprehensive study on the evolution of app ecosystems from different aspects with a dataset scaling more than three years [57]. Li et al. analysed the evolution of mobile app usage from both macro-level (*i.e.* app category) and micro-level (*i.e.*, individual app) and detailed how users' usage changes over time [11]. However, these researches have not mined any correlation between apps which is important to understand the app usage behavior of mobile users. Therefore, different from existing works, we look at the co-occurrence relationship between apps and further explore how such relationship changes over time, which indeed represents the dynamic correlations of different apps.

2.3 Graph Representation Learning Application

Graph is often used to construct a network to represent intricate relationships and recently many studies have focused on graph representation learning [58], [59], [60], [61], [62]. As a result, a lot of researchers have applied it to many areas [63], [64], [65], [66], [67], [68]. For example, in the field of human mobility, to capture human mobility motivation, Shi et.al. jointly mapped the user, time, location and semantic information into the same vector space, based on a heterogeneous graph constructed by the co-occurrence relationship among these four types of entities [68]. In terms of social relationship, Tian et al. constructed a heterogeneous graph based on social relationships between Twitter users and location context information, and learned vector representations to improve the accuracy of location inference [67]. However, to the best of our knowledge, only a few works have applied graph representation learning to app usage behavior [65], [66]. For example, Chen et al. learned embeddings from three subgraphs containing the relationships between app-location, app-time and app-app respectively, realizing context-aware app usage prediction [65]. In this paper, we construct app co-occurrence graphs to more comprehensively preserve the correlations of different apps in usage context, and then we learn app embeddings to represent apps' usage context information. Our work is a new exploration of the application of graph representation learning in app usage behavior.

2.4 Dynamic Graph Embedding Methods

For dynamic graphs, there are also many researchers devoting to proposing novel approaches to learn their graph embeddings with temporal information [16], [17], [18], [19], [20], [69], [70], [71], [72]. According to the way of integrating time information into graph embeddings, these methods can be generally divided into three categories, i.e., Sequence-Model based methods, Decomposition based methods and Random Walk based methods. Specifically, the Sequence-Model based methods often utilize RNNs to capture the temporal dynamics and then combine these RNNs with embedding encoders to produce graph embeddings with time information. For instance, Manessi et al. jointly learn structural relationships among the nodes and temporal information between graphs based on Graph Convolutional layers connected with LSTMs [18]. For Decomposition based methods, by decomposing a tensor that aggregates all structural relationships and temporal observations, they are able to generate graph embeddings with temporal patterns. For example, Dunlavy et al. successfully predict the adjacency matrix of the next timestamp via node embeddings obtained by decomposing a 3-order tensor which is formed by stacking those historical consecutive matrices [19]. In Random Walk based methods, in order to leverage temporal aspect of dynamic graphs, time is utilized as a regularizer to constrain the node embeddings over consecutive snapshots. A striking example is that Bian et al. adopt metapath2vec to generate random walks on each static knowledge graph and then they use a time regularizer to adjust embeddings for those affected nodes [20].

Comparing these three types of methods, we find that a critical limitation of the Sequence-Model based methods and the Decomposition based methods is that their dynamic graphs are often assumed to have fixed nodes, in other words, nodes are not added or removed. This may be because in their models, the time information is considered as part of the input of encoders to produce dynamic graph embeddings. Thus, they require rich historical observations for each node. Additionally, they are often designed to solve supervised issues (e.g., link prediction and node classification) and do not have good performance in unsupervised tasks (e.g., clustering). However, Random Walk based methods do not have these limitations. On the one hand, in their models, time information is not directly involved in encoders, but just serves as a regularizer to impose a smoothness constraint, so these models do not desire for strong time dependency. Thus, they are more adaptable to handle various types of observations, such as node addition, node deletion, edge addition, edge deletion, etc. On the other hand, they are an ideal option for unsupervised tasks, since one of the major advantages of random walk is that they do not need to be combined with a decoder when learning embeddings [16]. Look at our research, our app graphs are quite changeable and our task is unsupervised. Therefore, we follow the idea of Random Walk based methods to solve our problem.

3 App Embedding Learning Framework

In this section, we introduce how we learn app usage context embeddings based on a long-term app usage dataset in detail. We first construct an app co-occurrence graph for each short period and then we learn app usage context embeddings across different periods by leveraging the graph representation learning algorithm.

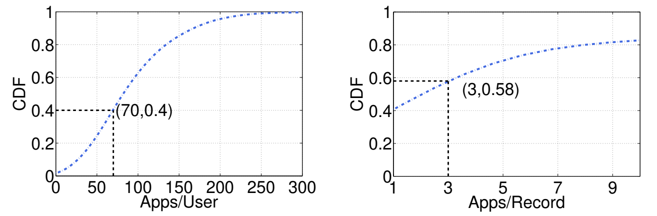
3.1 App Usage Dataset

To execute a longitudinal study of app usage, we have developed an Android-based mobile app to collect users' app usage behavior automatically. Considering users' privacy protection, all data collection items will be showed to users before the app is installed. The app will record users' app usage data every time 1% reduction in battery power. Each record contains an anonymous user ID, timestamp, battery level, and a complete list of apps currently interacting with the user. Up to now, the data collected by the app has covered over 100,000 apps and more than 30,000 mobile users from over 100 countries. Besides, the data has been partly released[1]. In the collected data, the recording durations of different users are quite different (ranging from several months to years) because users install and uninstall this app at different time. Since we aim to

1. https://www.cs.helsinki.fi/group/carat/data-sharing/

TABLE 1
Summary of our dataset.

#Records	14,800,960
#Apps	13038
#App Categories	49
#Time Span	7 years(2012-2018)
#System	Android



(a) Number of Apps Per User (b) Number of Apps Per Record

Fig. 1. Basic statistics of our dataset. (a): CDF of the number of apps used by per user. (b): CDF of the number of apps contained in per record.

explore the long-term dynamics of app usage, user records with more than three-year duration are selected. Finally, we obtain a long-term app usage dataset with over 14 million records from 1,608 Android users during 7 years (from 2012 to 2018), and it covers more than 13,000 apps. To better understand the characteristics of these apps, we also crawl the category information for them from Google Play. We find that these apps cover up to 49 categories, such as Productivity, Communication, Entertainment, Business and so on. The summary of our dataset is shown in Table 1.

Based on this dataset, we first conduct basic statistical analysis to show data quality for our research. First, we calculate the number of apps used by per user and show its Cumulative Distribution Function (CDF) in Fig. 1 (a). The result shows that 60% users have used more than 70 apps and top 40% have used more than 95 apps, illustrating the good coverage of apps that users interact with in their daily life. Second, to show the probability of app co-used, we calculate the number of apps contained in per record and plots its CDF in Fig. 1 (b). We can observe that 42% records contain more than 3 apps. This demonstrates that users often use a series of apps at the same time, so there exists a high probability of app co-used. Thus, it is necessary and valuable to pay attention to the app usage context, which may help us better understand the users' online habits. Therefore, although a new record will only be generated when the power drops by 1%, our app can still get a high-quality dataset which is large-scale, has a good app coverage and shows a high probability of app co-used.

Ethics. We want to point out that the privacy of users has been well protected during data collection and processing. First, before installing this app, all the data collection items are displayed to the user, and only after the user agrees to these items, our app can be installed on the user's device. Therefore, the involved users completely know and assent to our data collection. Second, at the time of installation, the user will be assigned a randomly generated identifier, and any sensitive personal information will not be involved during collection. Hence, the user's true identity and privacy have been well protected. Third, all of the researchers have signed a strict non-disclosure agreement before processing this dataset. Besides, throughout our processing procedure, this dataset is located on our private server protected by authentication mechanisms and firewells. Thus, whether in data collection or data processing, we are very careful to and have taken a series of measures to protect the privacy of users.

3.2 Constructing App Co-occurrence Graphs

Our long-term large-scale dataset contains abundant usage context information. However, such usage context information is quite complex and changeable over time, due to the complex and changeable app usage behavior of mobile users. Thus, how to effectively and completely extract the usage context information is very important. In this paper, we propose to utilize graph, a flexible and extensible structure, to help store such information. Specifically, we firstly divide the whole long period of the dataset into several short periods. Then, for each short period, an app graph with co-used relationship will be constructed, *i.e.*, app co-occurrence graph, containing all of the usage context information in the period. In detail, the graph takes apps as its nodes, and there will be an edge between two apps if they have ever co-used, in other words, if they have ever appeared in the same record in our dataset. Besides, in our study, the frequency of co-used between two apps plays a critical role, since it reflects the co-used probability and further affects the closeness of apps. Thus, we use it as the edge weight in the app graph. So for any app in a graph, its usage context can be completely stored and clearly displayed. Therefore, the abundant but complex usage context information can be extracted from the massive app usage behaviors effectively.

After the above processing, we have obtained a series of app graphs of different periods. However, how to measure the dynamics of app usage contexts across different graphs is still a thorny problem. Next we will introduce our solution to this problem.

3.3 Learning App Usage Context Embeddings

To measure the dynamics of app usage context across different periods, we propose to first quantify an app's usage context by graph representation learning, and then obtain the quantified dynamic distance. In other words, through graph representation learning, any app's usage context information is expected to be represented by a vector, *i.e.*, **app usage context embedding**, which can participate in mathematical calculations. In addition, if two apps are often co-used, and their respective co-used apps are similar, their usage context information will be more similar. Then they will be closer to each other in the app graph. Thus, accordingly, in the embedding space, we need to assure that the embeddings of these two apps will also closer to each other. In other words, their embeddings will be more similar, or the distance of their embeddings will be smaller.

Among the leading methods of dynamic graph representation learning, we follow the framework of Random Walk based methods [16], due to its better performance on unsupervised tasks and its higher adaptability on node dynamics. Based on this framework, we design a dynamic graph embedding algorithm for our task. To be specific, we first generate random walks on each snapshot and then input them into an encoder to learn app embeddings (we call it **app usage context embedding** to represent its usage context). Second, to incorporate the effect of temporal dynamics, we use a time regularizer as a smoothness constraint to adjust the app embeddings over consecutive snapshots. To this end, we learn app embeddings through node2vec algorithm [58], which first samples random-walks from the graph as sentences and then learns node embeddings by skip-gram model. The reasons of employing node2vec instead of other random walk based algorithms, like walk2vec [73], are mainly as follows. First, node2vec is good at dealing with weighted graphs. In our study, the app graphs are all weighted and their edge weights play a critical role in our study as they reflect the closeness of co-used apps, so node2vec is a good model to integrate such information. Second, it performs better on capturing the homophily similarity of a graph than other algorithms. That is, if two nodes are highly interconnected and belong to similar graph clusters or communities, they will be embedded closely together [58], so it is helpful to capture topological information hidden in the local contexts in our app co-occurrence graphs. In fact, in a graph, there are two kinds of node similarity, i.e., homophily and structural equivalence. And in this algorithm, return parameter p and in-out parameter q jointly control the similarity choice. In our study, by setting the appropriate values of p and q, we capture the homophily similarity of the app graph and obtain embeddings for all apps. So for any app graph, if two apps are closer in the app graph, their embeddings will be also closer in the corresponding embedding space.

However, app embeddings of different periods cannot be compared, because they distribute in different embedding space. To solve this problem, we need to align the different embedding spaces of different periods to the same coordinate axes. In our study, we use orthogonal Procrustes [74] to achieve this goal. The specific procetures are as follows.

We denote d as the dimension of embedding vectors, S as the intersection apps of period (t) and period $(t+1)$, $n^{(t)}$ as the number of apps of period (t) and $n^{(t+1)}$ as the number of apps of period $(t+1)$. $\mathbf{A}^{(t)}$ is the embedding matrix of period (t), and $\mathbf{A}^{(t)} \in \mathbb{R}^{d \times n^{(t)}}$. Similarly, $\mathbf{A}^{(t+1)}$ is the embedding matrix of period $(t+1)$, and $\mathbf{A}^{(t+1)} \in \mathbb{R}^{d \times n^{(t+1)}}$. $\mathbf{A}_S^{(t)}$ is the submatrix of $\mathbf{A}^{(t)}$ and it only contains the embeddings of S, and $\mathbf{A}_s^{(t)} \in \mathbb{R}^{d \times |S|}$. So as $\mathbf{A}_S^{(t+1)}$, the submatrix of the embedding matrix $\mathbf{A}^{(t+1)}$. Then the optimizing function $\mathbf{R}^{(t)}$ is as follows:

$$\mathbf{R}^{(t)} = \arg \min_{\mathbf{Q}^{\mathrm{T}}\mathbf{Q}=\mathbf{I}} \left\| \mathbf{Q}\mathbf{A}_s^{(t)} - \mathbf{A}_s^{(t+1)} \right\|_F, \tag{1}$$

where $\mathbf{A}_s^{(t)} \in \mathbb{R}^{d \times |S|}$, $\mathbf{A}_s^{(t+1)} \in \mathbb{R}^{d \times |S|}$. Besides, $\mathbf{Q} \in \mathbb{R}^{d \times d}$, which is the rotational matrix to adjust the coordinate axes between two consecutive periods. For pair-wise continuous periods, we adjust their coordinate axes successively.

Finally, for all periods, their aligned embedding matrices are all obtained, with which we are able to compare embeddings from different periods and quantify their distance. Therefore, it becomes possible to investigate how the app usage context changes over time and further explore general laws.

In a short summary, we effectively extract apps' co-used relationship and their usage context information from the original app usage behavior data, by constructing the app co-occurrence graph for each period. Then node2vec is performed on each app graph to transform each app node into an embedding vector, which can represent its usage context information. After aligning these embeddings into the same coordinate axes, we can measure the app usage context dynamics across different periods. Thus, we are able to study the long-term dynamics of app usage context and further explore general laws, via app usage context embeddings.

4 Visualization Analysis of App Usage Context Embeddings

According to the procetures in Section 3, an app's usage context will be represented by an embedding. Based on these embeddings, we can investigate the long-term dynamics of app usage context. However, before this investigation, it is necessary to evaluate the representation ability of these embeddings. In addition, according to the algorithm in Section 3, the closer embeddings will have more similar usage context, in other words, will have larger probability to co-occur. So our evaluation task is to verify whether closer embeddings will be more likely to co-occur.

To this end, we first look at the app graph, which is constructed on the basis of real-world app usage data and has extracted the co-occurrence relationship from the data. In the app graph, if two apps co-occur more often, they will be more closer. Naturally, we consider to verify whether closer embeddings will be also closer in the app graph. In addition, users often use a series of apps to satisfy their habits in the daily life. For example, in the leisure time at home, they often use Entertainment apps, Video apps and Music apps. When they are travelling, they often use Travel apps and Finance apps. So we can also verify whether closer embeddings will co-occur to satisfy users' habits. Besides, users may use several apps together to complete a complex task. For instance, to post a wonderful news on Instagram (a popular Social app), users may use Instagram, Photography apps and Video apps at the same time. So these function-relevant apps are likely to be used together. Thus, we can also verify whether closer embeddings will co-occur to realize an app's peculiar function.

In a short summary, to evaluate the representation ability of the app embeddings, we cosider to conduct this verification from the following three aspects: 1) app usage context embeddings restore graph structure information: to verify whether closer embeddings will be also closer in the app graph; 2) app usage context embeddings reflect user's habits: to verify whether closer embeddings will co-occur to satisfy users' habits; 3) app usage context embeddings show app's peculiar functions: to verify whether closer embeddings will co-occur to realize an app's peculiar function.

Before showing the evaluation experiments, we will introduce the fundamental experiment settings in our study. First, we divide the 7-year dataset into 7 yearly groups. And

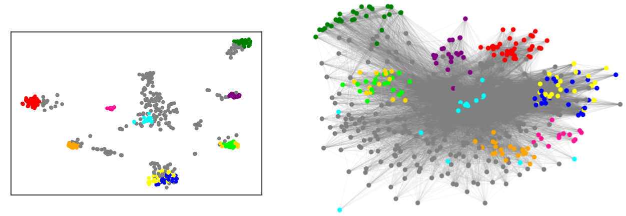
TABLE 2
Statistics of the graph in every year.

Year	2012	2013	2014	2015	2016	2017	2018
#Nodes	655	1834	4152	5302	4015	3017	1892
#Edges	1718	27765	25343	38551	28666	61527	40056
#Average Degree	5.24	30.27	12.20	14.54	14.28	40.78	42.34
#Overlapped Nodes	—	18.9%	27.4%	43.4%	64.4%	61.2%	65.1%

we mainly have two reasons to choose the time period as one year. On the one hand, our work aims to macroscopically investigate the dynamics of app usage context and explore general laws, so it seems not necessary to choose a very short period. On the other hand, a shorter period may cause sparser graphs, so it will be difficult to learn good embedding representations. Thus, we use the year as the time period. The detailed statistics of the graph in every year is shown in Table 2. Note that "#Overlapped Nodes" means the proportion of the overlapped nodes between every two consecutive years in all nodes in the latter year of these two years. For example, "18.9%" means that the number of overlapped nodes between 2012 and 2013 accounted for 18.9% of the total number of nodes in 2013. We can see that the number of nodes in a graph gradually increased before 2015, while decreasing after 2015. This may be because after 2015, users are inclined to frequently use a set of apps according to their personal habits, making the apps with poor user experience eliminated in the fierce market competition, while making the apps with good user service survived. What's more, for the proportion of overlapped nodes between two consecutive years, we can observe that the value of this metric is always below 50% before 2015, and its maximum is only 65.1%, implying that these app graphs are quite changeable, with active node deletion and addition. For other parameters, the return parameter p and the in-out parameter q are experimentally set to be 1 and 0.25 respectively when learning embeddings by node2vec, to capture the homophily similarity best. Besides, the size of the embedding vector is set to 128. Based on these experimental settings, we conduct the following evaluation experiments.

4.1 App Usage Context Embeddings Restore Graph Structure Information

Firstly, we want to verify whether closer embeddings will be also closer in the app graph. To this end, we need to extract apps whose embeddings are close in the embedding space, and then observe their positions in the app graph. Thus, on one hand, to obtain closer app embeddings, we cluster them by K-Means. On the other hand, in the app graph, we mark the apps in the same cluster with the same color. Taking the 2017's app embeddings for example, they are clustered into around 60 clusters, where 60 is the best number of clusters according to Calinski-Harabasz Index [75]. Then we mark the apps in the different clusters with different colors. Note that in order to show more clearly, we just color the top 10 largest clusters, and the results are shown in Fig. 2. Fig. 2 (a) is the clustering result of its embeddings which are mapped into 2-dimensional space by t-SNE [76], with the closeness in high-dimensional space preserved. Fig. 2 (b) reflects the positions of the apps in the same cluster in the app graph. Note that these positions are generated by Fruchterman-Reingold Algorithm [77], one of the well-known algorithms for the graph layout [78]. We can observe that the apps in the same cluster are also neighbours in the corresponding app graph, in other words, the closer embeddings will be also closer in the app graph. This demonstrates that the app usage context embeddings restore the structure information in the graph, so the embeddings learn the usage context information well.



(a) Clustering Result of App Embeddings (b) App Positions in the Initial Graph

Fig. 2. Comparison of the node closeness in the embedding space and the graph. (a) represents the clustering result of app embeddings of year 2017 and the top 10 largest clusters are colored. (b) reflects the corresponding positions of the apps in (a) in the initial app graph.

4.2 App Usage Context Embeddings Reflect User's Habits

Secondly, we further verify whether closer embeddings will co-occur to satisfy users' habits. To achieve this goal, first of all, we also need to cluster all the app embeddings to obtain those closer embeddings. Then for each cluster, we observe if the apps in it can satisfy users' typical habit, such as entertainment and travelling, with the help of the apps' category information. Without loss of generality, we take app embeddings of another year 2018 for example. We cluster them into 60 clusters by K-Means according to Calinski-Harabasz Index [75]. The result is shown in Fig. 3 (a). We can observe that there are two kinds of clusters: the central largest cluster and the peripheral small clusters. First, for the central largest cluster, we find that many of its apps are very popular, and even top 10 popular apps in 2018 are all in it, such as Facebook, Twitter, Chrome, Gmail and so on. This indicates that popular apps often be co-used by users. This is reasonable for their high applicability anytime and anywhere. Second, for the peripheral small clusters, we find that the apps are more likely to co-occur to satisfy users' typical habits. For example, in one of these clusters, we label the apps in it with app-category information, and then we observe that the most app-categories (also called main app-categories) are Travel, Transportation and Photography, which are very common to co-occur when users are travelling. We label this cluster as "Travel" and regard the label as its habit label. The sketch map of it is shown in Fig. 3 (b), where the blue solid dots represent apps and they are labeled with their app-categories, *e.g.*, Travel, Transportation and Photography. Finally, according to the main categories in each small cluster, we summary five kinds of typical habits, and they are entertainment, travel, sports, game and

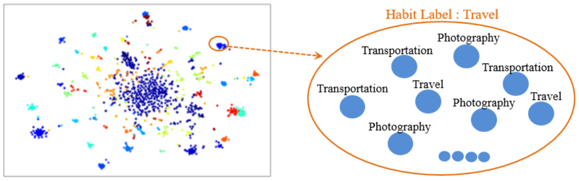


Fig. 3. App usage context embeddings reflect user's habits. The left picture shows the clustering result of app embeddings of year 2018. The right picture is the sketch map of a cluster whose habit label is "Travel", where the blue solid dots represent apps and they are labeled with their app-categories, *e.g.*, Travel, Transportation and Photography.

TABLE 3
Five kinds of user habits and their main representative app-categories.

Habit Label	Main App-Categories	Number of Clusters
Entertainment	entertainment, music, media	13
Travel	travel, transportation, photography	16
Sports	sports	4
Game	game	3
Business	business, news, education, books	4

business. The detailed information is shown in Table 3. Note that some clusters are ignored because they contain very few entities (*e.g.*, less than 8). We can observe that every cluster can be labeled with a typical habit. In other words, for every cluster, we can successfully infer the habit it may satisfy, according to the main app-categories it contains. Thus, it is obvious that closer embeddings will co-occur to satisfy users' habits.

4.3 App Usage Context Embeddings Show App's Peculiar Functions

Thirdly, we plan to verify whether closer embeddings will co-occur to realize an app's peculiar function. To this end, we need to focus on some specific apps and their peculiar functions, instead of only considering their app-categories. Then we seek for their neighbours in the embedding space and further observe the neighbours functions, to verify if they can cooperate to complete a task. Specifically, in the experiment, first, we choose two groups of apps: Social apps and Travel apps The apps in each group have their own peculiar functions. The detailed information is shown in Table 4. From the table, we can observe that although Instagram and LinkedIn are both Social apps, their peculiar functions are quite different. Instagram mainly provides picture sharing services for young people, while LinkedIn mainly provides social services for professionals. So as Airbnb and Uber. Then, to obtain their neighbors in embedding space, we quantify the distance between them as follows:

$$s^{(t)}\left(a_i, a_j\right) = \operatorname{cosDist}\left(\boldsymbol{a}_i^{(t)}, \boldsymbol{a}_j^{(t)}\right), \tag{2}$$

where $\boldsymbol{a}_i^{(t)}$ and $\boldsymbol{a}_j^{(t)}$ are embeddings of apps a_i and a_j in period (t), respectively. cosDist means Cosine Distance, and $s^{(t)}\left(a_i, a_j\right)$ denotes the cosine distance of apps a_i and a_j in the embedding space in period (t).

For each app, we obtain its top 20 closest neighbors in each period, and the distribution of these neighbors' app-categories is shown in Fig. 4. In each sub-figure, the sum

TABLE 4
Peculiar functions of different apps.

App	Peculiar Function
Group1: Social apps	
App	**Peculiar Function**
Instagram	photos, young people
LinkedIn	career, proffessionals
Group2: Travel apps	
App	**Peculiar Function**
Airbnb	house review
Uber	transportation

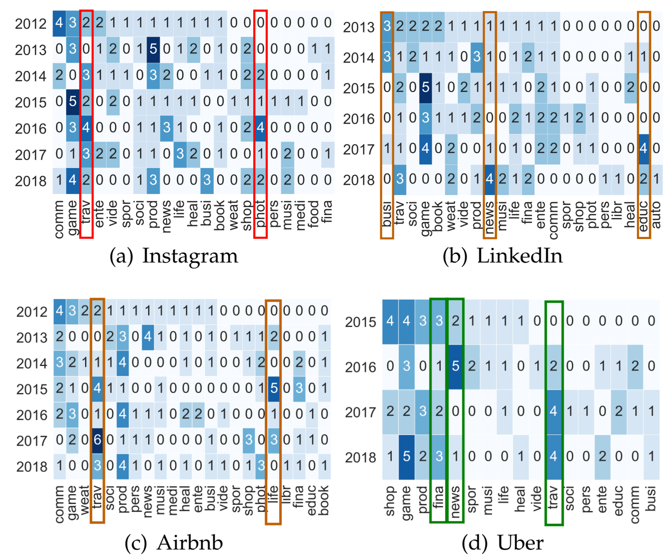


Fig. 4. Comparison of the neighbours of four apps with different peculiar functions. Co-occurrence for app's peculiar functions. Each sub-figure shows the distribution of app-categories of the target app's top 20 closet neighbors in each period. Specifically, in each sub-figure, the sum of the values in each row is 20, and each value represents the number of apps belonging to a certain app-category in a certain period.

of the values in each row is 20, and each value represents the number of apps belonging to a certain category. For example, in Fig. 4 (a), the number "4" in the first row and the first column represents that there are 4 Communication apps in the top 20 closest neighbours of Instagram in 2012. In these figures, we have marked the representative categories of each app which are the categories that best distinguish this app from other apps. For example, the representative categories of Instagram are Travel and Photography, and they appear almost twice as likely in Instagram neighbors as LinedIn's. The probability that Travel and Lifestyle will appear in the neighbors of Airbnb is 23% higher than that of Uber. We can observe that for each app, its representative categories are consistent with its peculiar functions. For example, for Uber, its representative categories (*i.e.*, Finance, News and Travel) are highly related to short-distance payment task. For LinkedIn, the categories Business, News and Education can reflect the highly educated characteristics of its users. Therefore, closer embeddings will co-occur to realize an app's peculiar function, in other words, the app's neighbors may co-occur with this app to complete a specific task together, and we can infer an app's function from its closest embedding neighbours.

To sum up, the above three experiments verify that apps that are closer in the embedding space will be also closer in the app graph. Besides, they will also co-occur to satisfy users' habits or to realize an app's peculiar function. This indicates that the apps' usage contexts are well learned by their embeddings. Based on these embeddings with strong representation ability, next we will investigate how the app usage context changes over time and further try to explore general laws.

5 Long-term Dynamics of App Usage Context

In this work, we aim to macroscopically investigate long-term dynamics of app usage context. With the rapid development of communication technology and the high popularity of mobile phones, app markets have boomed and changed a lot during recent years [1]. This inspires us to ask how the app's usage context evolves over time. In addition, with the category information of all apps, we can further explore the usage context dynamics of different app categories. However, before these investigations, we need to quantify the change of usage context of the same app across different periods. Since the embeddings of all periods have been aligned into the same coordinate axes, we use the following method to measure the change of usage context of the same app across different periods.

$$\Delta^{(t)}\left(a_i\right) = \operatorname{cosDist}\left(\boldsymbol{a}_i^{(t)}, \boldsymbol{a}_i^{(t+1)}\right), \tag{3}$$

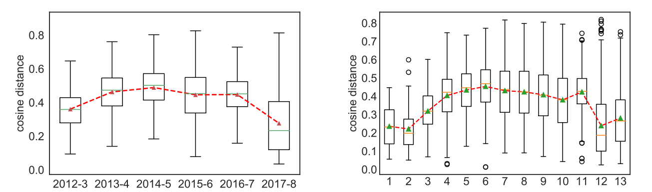
where $\boldsymbol{a}_i^{(t)}$ and $\boldsymbol{a}_i^{(t+1)}$ are the embeddings of the app a_i in period (t) and period $(t+1)$, respectively. $\Delta^{(t)}\left(a_i\right)$ is the change of usage context of the app a_i from period (t) to period $(t+1)$. Next we will investigate how the apps' usage contexts change across different periods, and further explore the usage context dynamics of different app categories.

5.1 Overall App Usage Context Dynamics

For every two consecutive periods, we compute the changes of usage context of the apps existing in both periods, and then display the results by Boxplot as shown in Fig. 5. In the figure, green horizontal line represents the median of the usage context changes of all apps, and red upper triangle represents the average.

Generally speaking, the apps' usage context macroscopically undergoes up and down phrases. For the first phase (Phase1), from 2012 to 2014, there is a trend that the dynamic of app usage context is on the rise, which indicates that in these years, apps are used in drastically unstable and changeable contexts. While for the second phase (Phase2), from 2014 to 2018, just the opposite of the first phase, such distance has a downward trend, which suggests that the apps' usage context is getting stable during these years. Around 2014, the dynamic of app usage context reaches its maximum, and it is about 1.4 times that of 2012.

In order to better observe the evolution of app usage context over time, we shorten the time span from one year to half a year. Based on these 14 half-year app co-occurrence graphs, we obtain the fine dynamics of app usage context, as illustrated in Fig. 5 (b). In general, the evolution trend



(a) Time Span = 1 Year (b) Time Span = 1 Half of a Year

Fig. 5. Boxplots of the evolution of app usage context when time span is set to be 1 year and half year, respectively. "2012-3" on the Horizontal axis in (a) means from year 2012 to year 2013, and so on. "1" on the Horizontal axis in (b) means from the 1st half of 2012 to the 2nd half of 2012, and so on. In the figure, green horizontal line represents the median of the usage context changes of all apps, and red upper triangle represents the average.

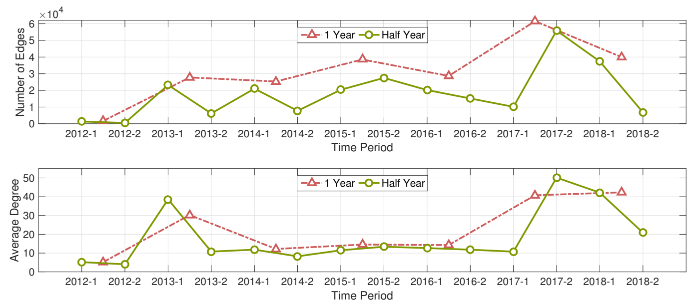


Fig. 6. Comparison of one-year app co-occurrence graphs and half-year app co-occurrence graphs on the number of edges and the average degree. "2012-1" on the Horizontal axis means the 1st half of 2012, while "2012-2" means the 2nd half of 2012.

revealed in Fig. 5 (b) is consistent with that in Fig. 5 (a), indicating that the time span has little effect on the macroscopic evolution of app usage context. However, a shorter time span will cause sparser graphs, e.g., the edge size and the average degree of a graph will decrease significantly, as shown in Fig. 6. Consequently, compared to the one-year app co-occurrence graph, the half-year app co-occurrence graph will involve less app co-occurrence relationships and incomplete app usage context. Therefore, we still choose one year as our time span to develop other experiments.

As for the reasons behind the phenomenon in Fig. 5, one of the possible reasons to cause the instability in the first phase is the development of mobile networks. During 2012 to 2014, resulting from the breakthrough in communication technology, the fourth-generation (4G) mobile networks gradually replaced the traditional 3G networks [79]. Thus, the app market ushered in a new round of prosperity. Out of curiosity and freshness, mobile users faced more various app choices, and they might uninstall an app which had not been downloaded for a long time. Therefore, generally, the context of an app would change fast, which led to the large usage contextual change distance during these years. After 2015, with the fierce competition in the app market and more stable usage habits of users, the usage context of a specific app gradually became stable. Therefore, for these years, their usage contextual change distance started to get smaller.

During periods like Phase1, to avoid being replaced

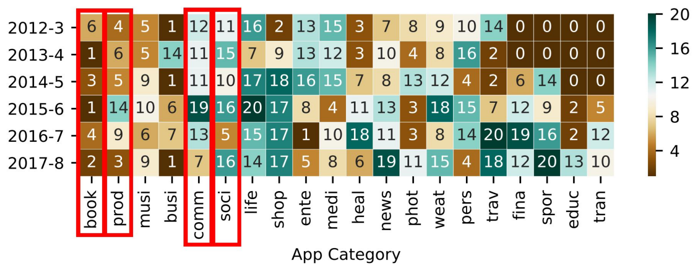


Fig. 7. The app usage context change of app-categories between two consecutive years. Each value represents the rank of app usage context change of an app-category in all app-categories in a certain period. Rank "1" means the largest change, while rank "20" means the smallest change. The value 0 means its corresponding app-category has less than 10 apps, and will not participate in ranking for large contingency.

quickly in the fierce competition of the app market, app developers should keep sensitive to new technologies and develop apps with different functions. While during periods like Phase2, apps' usage contexts are getting stable, which means the mobile users usually have stable co-used apps [80], so the app developers can consider to design one-stop apps to make mobile users enjoy more functions without switching between apps.

5.2 App Usage Context Dynamics of Different App Categories

Apps of different categories have different functions and their usage context is quite different, such as the apps in Table 3. Next, we will explore the usage context dynamics of different app categories.

Firstly, we quantify the usage context change of a specific app category between two consecutive periods by the median of all its apps' usage context changes. For example, if there are totally 100 Social apps existing in both 2012 and 2013, the usage context change of the category Social from 2012 to 2013 is the median of changes of the 100 Social apps. To track the long-term (7-year) dynamics, we filter out those categories that don't appear in all periods. Then we obtain the ranking results for the app categories as shown in Fig. 7. Each value represents the rank of app usage context change of an app-category in all app-categories in a certain period. Rank "1" means the largest distance while rank "20" means the smallest distance. The value 0 in this table means its corresponding category has less than 10 apps and will not participate in ranking for large contingency.

From Fig. 7, we find that the change of usage context of most app categories does not have a stable trend. However, there are still some apps that almost always ranking in top 10, like Books and Productivity, while some other apps are almost always in the last 10, such as Communication and Social. This means that the contexts of Books apps and Productivity apps change more faster than other apps, while Communication apps and Social apps have the slower change rate of usage contexts.

For Productivity and Books apps, they usually need to cost users longer time, so they are more easily interrupted by other apps, which increases the randomness of their neighbors. Therefore, their positions in the embedding space are more unstable. While for Communication apps and Social apps, it doesn't need to take a long time to use them, so the probability of being interrupted tends to be smaller. In addition, they are usually initiatively used by people, so their neighbours often have related functions with them. Therefore, the characteristics of their neighbours are more stable than Productivity apps and Books apps. Thus, their usage contextual change distance is smaller. Apps that have shorter usage time tend to have stable usage context, so app developers can focus on fragmented reading function to shorten their apps' usage time to obtain stable usage context.

In a short summary, to study the dynamics of usage context via app embeddings, we first quantify the change of usage context between two consecutive periods. Then we observe its evolution over time and find that this evolution undergoes two phases: app's usage context changes more and more drastically from 2012 to 2014, while changing more and more smoothly after 2015. In addition, in terms of app categories, we find that the usage contexts of Books apps and Productivity apps change more faster than other apps, while Communication apps and Social apps have the slower change rate of usage context. For these two findings, we also provide possible reasons and implications.

6 Influence Factors of App Usage Context Changes

In this section, to further reveal the underlying substantial factors behind the general laws, we pay attention to three possible factors: app usgage frequency, app contextual diversity and app usage popularity. Our goal is to investigate if they have an impact on the change of an app's usage context and if so, how do they affect it.

6.1 Influence of App Usage Frequency

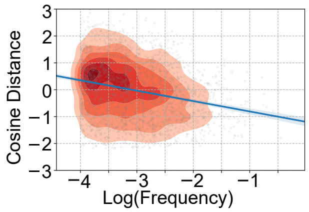
An app's usage frequency usually represents its usage necessity among users. How does the usage context change for apps with high/low frequency? Motivated by this question, we obtain both apps' usage context change distances (according to Equation 3) and their frequencies, which are defined as the following Equation 4.

$$f^{(t)}\left(a_i\right) = n^{(t)}\left(a_i\right) / \sum_{k=1}^{\left|S_{(t)}\right|} n^{(t)}\left(a_k\right), \tag{4}$$

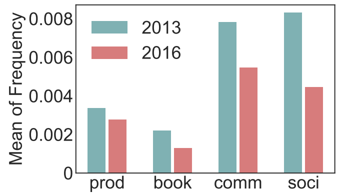
where $n^{(t)}\left(a_i\right)$ represents the number of times that the app a_i appeared in the records of the period (t), and $t \in \{2012, 2013, 2014, 2015, 2016, 2017\}$. $S_{(t)}$ denotes the intersection apps of period (t) and period $(t+1)$, and $i \in \{1, \ldots, \left|S_{(t)}\right|\}$. To ensure the robustness of our result, we ignore those apps whose frequency is less than 10^{-4}.

To make our results more obvious, we log-transform the frequencies and normalize the distance values to have zero mean and unit variance. We denote the normalized values with $\tilde{\Delta}^{(t)}\left(a_i\right)$. We draw the points determined by the distance score and frequency score on the map, and further obtain their scatter density map and 95% Confidence Interval (CI), as shown in Fig. 8 (a). Thus, the relationship can be represented by the following formula:

$$\tilde{\Delta}^{(t)}\left(a_i\right) \propto \beta_f \log\left(f^{(t)}\left(a_i\right)\right), \tag{5}$$



(a) Power Law between App's Usage Context Change and Its Usage Frequency



(b) App Usage Frequency of Different App-categories

Fig. 8. The statistical law between app's usage context change and usage frequency (a), and its corresponding verification (b).

where $\beta_f < 0$.

This result shows that the relationship of apps' usage context change distance and their frequency scale as a negative power law. Thus if an app's frequency is higher, its usage context change distance will be smaller or its context will change more slowly.

The negtive trend in Fig. 8 (a) may be because if an app's frequency is higher, the more initiatively users will use it, so their neighbour apps are often functionally dependent on them. Therefore, the characteristics of their neighbours will be more stable. Consequently, their context change distance will be smaller.

To further verify the validity of this finding, we take the previous results in Section 5 for verification. Reflect on the results we observed earlier in Section 5.2: the contexts of Books and Productivity apps change faster than Communication and Social apps. According to the negative power law obtained above, the frequencies of Books and Productivity apps will be lower than those of Communication and Social apps. Thus we further observe the frequencies of these categories. In our experiment, for each of these categories, we regard the mean of its frequency values in a year as its frequency in this year. When selecting the period for observation, according to Fig. 5, we have avoided the typical years, such as 2012, 2014 and 2018, while choosing 2013 and 2016. The result is shown in Fig. 8 (b). Note that the frequency values of the y-axis in this figure is the initial frequency values $f^{(t)}(a_i)$. Obviously, the frequencies of Books and Productivity in both 2013 and 2016 are lower than those of Communication's and Social's in 2013 and 2016. This result further supports that an app's frequency does have an impact on the change rate of its usage context and their relationship scales as a negative law.

6.2 Influence of App Contextual Diversity

With the inspiration of the concept of "polysemy" in linguistics, we consider the contextual diversity of an app's neighbours as a possible factor to have an impact on the dynamics of app usage context. "Polysemy" means the number of senses of a word, which is often used in word research [81], [82]. Afterwards, Hamilton et al. [83] measured a word's contextual diversity as its polysemy in their task. In their study, words that occur in many distinct, unrelated contexts will tend to be highly polysemous. In our research, each app has its usage context, so it also has contextual diversity. Following the meaning of polysemy in their research, in our study we use "polysemy" as a proxy for an app's contextual diversity, in other words, function diversity of an app's neighbours. We measure an app's polysemy by its local clustering coefficient within the app graph [84]. The local clustering coefficient d of app a_i is calculated through the following formulas:

$$\begin{aligned} d(a_i) &= \frac{\sum_{c_k c_l \in N_{\text{PPMI}}(a_i)} \mathbb{I}\{\text{PPMI}(c_k, c_l) > 0\}}{|N_{\text{PPMI}}(a_i)|\,(|N_{\text{PPMI}}(a_i)| - 1)}, \\ N_{\text{PPMI}}(a_i) &= \{a_j : \text{PPMI}(a_i, a_j) > 0\}, \\ \text{PPMI}(c_k, c_l) &= \max\left\{\log\left(\frac{\hat{p}(c_k, c_l)}{\hat{p}(c_k)\,\hat{p}(c_l)}\right) - \alpha, 0\right\}. \end{aligned} \tag{6}$$

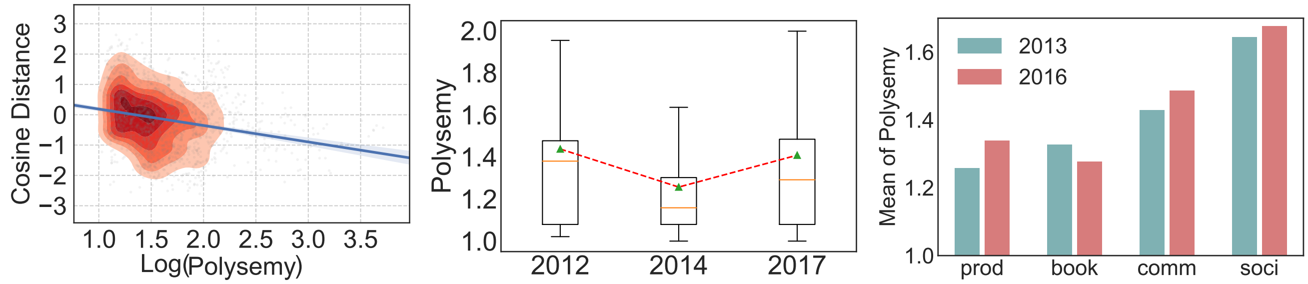
where $\alpha > 0$, and it is a prior value, which provides a smoothing bias and is set to be 0.75 in previous work [85]. PPMI means Positive Point-wise Mutual Information, and the $\hat{p}$ corresponds to the smoothed empirical probabilities of app occurrences. This measure counts the proportion of a_i's neighbors that are also neighbors of each other. According to this measure, an app will have a high cluster coefficient (and thus a low polysemy score) if the apps that it co-occurs with also tend to co-occur with each other. Polysemous apps that are contextually diverse will have low clustering coefficients, since they appear in disjointed or unrelated contexts.

To observe the relationship of an app's polysemy and its usage context change, we also obtain the quantified distance scores according to Equation 3, and obtain the polysemy scores by Equation 6. Like frequency, we also need to ignore those apps whose local clustering coefficients are more than 0.1. Besides, we also normalize the distance values to have zero mean and unit variance (*i.e.*, $\tilde{\Delta}^{(t)}(a_i)$). As for the local clustering coefficients, we first log-transform them, and then make them negative, in order to assure the larger scores correspond to the higher polysemy (more polysemous or more contextual diverse). The result is shown in Fig. 9 (a). In this figure, the greater the horizontal coordinate, more polysemous it will be. and such relationship can be represented by the following formula:

$$\tilde{\Delta}^{(t)}(a_i) \propto \beta_p\left[-\log\left(d^{(t)}(a_i)\right)\right], \tag{7}$$

where $\beta_p < 0$. Thus, the relationship of apps' change of usage context and their polysemy scores also scales as a negative power law. Thus if the functions of the neighbours of an app are more diverse, the change of the app's usage context will be smaller, which means that its usage context will be more stable. This may be because if an app's neighbors are more diverse, it will be difficult for such apps to be affected by sudden popularity or unpopularity from its neighbors [8], which indicates that its usage context will be more stable under the changeable app market. App developers should improve app's contextual diversity to enjoy a stable usage context.

To verify the reliability of this finding, we look back at Fig. 5, and count the polysemy scores of 2012, 2014 and 2017. The results are shown in Fig. 9 (b). We can see that there is a downward trend of polysemy scores from 2012 to 2014, but an upward trend from 2014 to 2017. Additionally, so as Fig. 8 (b), we also observe the polysemy scores of these app-categories. The results are shown in Fig. 9 (c). We can see



(a) Power Law between App's Usage Context Change and Its Polysemy (b) App Polysemy of Different Years (c) App Polysemy of Different App-categories

Fig. 9. The statistical law between app's usage context change and polysemy (a), and its corresponding verification (b-c).

that the polysemy scores of Books and Productivity in both 2013 and 2016 are lower than those of Communication's and Social's in 2013 and 2016. These results are also consistent with the law in Fig. 9 (a): if an app is more polysemous (in other words, the functions of the neighbours of the app are more diverse), its usage context will be more stable.

6.3 Influence of App Usage Popularity

In app markets, most apps have a very short life cycle. There is a common phenomenon that some new apps grab people's eyes quickly and become popular, but they may also quickly disappear. However, some other apps have always been popular. When an app's popularity changes, will its usage context also change? And what is the relationship of them? To explore their answers, we need to first obtain the popularity change score for each app. In our study, for year (t), the popularity of app a_i is regarded as its popularity rank $r^{(t)}(a_i)$ which is decided by its number of being used, where rank 1st corresponds the largest number of occurrence. Thus the popularity change of app a_i between period (t) and period $(t+1)$ is:

$$c^{(t)}(a_i) = r^{(t+1)}(a_i) - r^{(t)}(a_i). \tag{8}$$

Thus if an app's rank rises/falls, its popularity change score will be positive/negative.

To make the results more robust, we also normalize the distance values to have zero mean and unit variance. As for popularity change scores, they are transformed as follows:

$$\tilde{c}^{(t)}(a_i) = \begin{cases} \log\left(c^{(t)}(a_i)\right), & c^{(t)}(a_i) \geq 0 \\ -\log\left(-c^{(t)}(a_i)\right), & c^{(t)}(a_i) < 0 \end{cases} \tag{9}$$

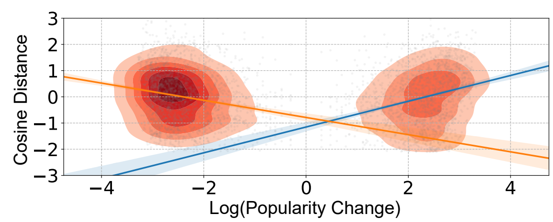
Thus whether this value is positive or negative, its app has a ranking change if it is close to 0.

The result is shown in Fig. 10 (a), and their relationship can be represented as follows:

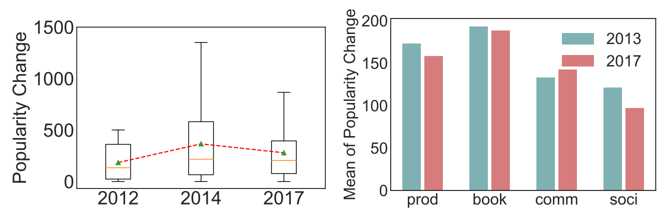
$$\tilde{\Delta}^{(t)}(a_i) \propto \beta_c \left|\tilde{c}^{(t)}(a_i)\right|, \tag{10}$$

where $\beta_c > 0$.

The relationship of apps' usage context change scores and their popularity change scores scales as a positive power law. The result shows that, generally, if an app's popularity change more, its usage context change will be larger. This indicates that the usage context of those short-lived apps tend to change a lot during long period. Besides, if an app is always popular, its usage context will be more stable.



(a) Power Law between App's Usage Context Change and Its Popularity Change



(b) App Popularity Change of Different Years (c) App Popularity Change of Different App-categories

Fig. 10. The statistical law between app's usage context change and popularity change (a), and its corresponding verification (b-c).

This may be because for those short-lived apps, there is not a stable structural relationship between itself and other apps, leading to their unstable usage context. As for those popular apps, after fierce competition, they stand out from the other apps with the same category, and form a stable ecosystem with other apps of different categories. For app developers, if they want to make their app's life cycle long, it is better to let it have a stable relationship with other apps.

We also further count the popularity scores for different years and different app categories, and the results are shown in Fig. 10 (b) and (c). Note that the popularity change values of the y-axis in this figure is the initial absolute values of ranking change $\left|\tilde{c}^{(t)}(a_i)\right|$. From Fig. 10 (b), we can find that generally the popularity change in 2014 is larger than that of 2012 and 2017. This is corresponding to the result in Fig. 5: the app usage context in 2014 changes more drastically than that of 2012 and 2017. At the same time, in Fig. 10 (c), the popularity change of Books and Productivity is also larger than that of Social and Communication, which is corresponding to the result in Fig. 7. These results support the statistic law in Fig. 10 (a) from two different aspects.

To sum up, in this section, we focus on three possible factors related to an app's usage context change, *i.e.*, app usage frequency, app contextual diversity and app usage popularity. We find that if an app has a higher usage frequency, its usage context will change slower and such change scales as a power-law. The same law is for an app's contextual diversity. In addition, we also find that apps with stable usage context will also have stable popularity. So if an app is always popular, its usage context will be more stable. In addition, to further verify the validity of these findings, we take the previous observation in Section 5 for verification and the results are generally consistent with our findings. Therefore, these three factors deserve to be considered to study the usage context change of apps. Additionally, these findings can also bring valuable implication for the stakeholders. For example, for the app developers, if they want their apps to enjoy a longer life cycle, it is better to provide them with stable co-occurrence relationships with other apps. To provide such relationships, a possible way is to improve their apps' contextual diversities, in other words, improve the functional diversities of their apps, e.g., one-stop apps. Therefore, we believe that our findings concerning on these three factors will inspire the related researchers and stakeholders.

7 Conclusion and Future Work

In this paper, we utilize a long-term app usage data with seven years to understand the longitudinal app usage context dynamics via graph embedding. By building app co-occurrence graphs during different periods, we learn app embeddings accordingly and measure how an app's usage context changes by using the distance of neighboring app embeddings. Overall, our findings suggest that app usage context changed more and more drastically from 2012 to 2014, and then changed more and more smoothly after 2015. In terms of app categories, book and productivity apps change the fastest, while communication and social apps have the slowest change rate. Further, we find that three factors (i.e., app usage frequency, app contextual diversity, app usage popularity) are related to the usage context changes, and their relationships follow a power law. Our study opens up a new perspective for long-term app usage context analysis and provides meaningful implications for app developers.

Acknowledgments

The work is supported in part by the National Key Research and Development Program of China (No. 2019YFE0114000), Guangdong R&D Project in Key Areas under Grant (2019B010158001,2019B010156004), Industry-University-Research Cooperation Project in Zhuhai (No. ZH22017001200072PWC) and Guangdong Provincial Special Fund For Modern Agriculture Industry Technology Innovation Teams (No. 2021KJ122).

References

[1] M. E. Joorabchi, A. Mesbah, and P. Kruchten, "Real challenges in mobile app development," in *Acm*, 2013.

[2] J. Huang, F. Xu, Y. Lin, and Y. Li, "On the understanding of interdependency of mobile app usage," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2017, pp. 471–475.

[3] H. Cao, Z. Chen, M. Cheng, S. Zhao, T. Wang, and Y. Li, "You recommend, i buy: How and why people engage in instant messaging based social commerce," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW1, pp. 1–25, 2021.

[4] L. Yang, Y. Ke, X. Wu, Y. Shi, and Y. Tan, "Association rules mining analysis of app usage based on mobile traffic flow data," in *IEEE International Conference on Big Data Analysis*, 2018.

[5] A. Annie, *App Annie*, May 2017. [Online]. Available: https://www.appannie.com/cn/

[6] ——, *App Annie Report*, May 2017. [Online]. Available: https://tech.sina.com.cn/i/2017-05-05/doc-ifyeycfp9252047.shtml

[7] T. Li, Y. Fan, Y. Li, S. Tarkoma, and P. Hui, "Understanding the long-term evolution of mobile app usage," *IEEE Transactions on Mobile Computing*, 2021.

[8] V. F. Taylor and I. Martinovic, "To update or not to update: Insights from a two-year study of android app evolution," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS '17. New York, NY, USA: ACM, 2017, pp. 45–57. [Online]. Available: http://doi.acm.org/10.1145/3052973.3052990

[9] S. Alhabash and M. Ma, "A tale of four platforms: Motivations and uses of facebook, twitter, instagram, and snapchat among college students?" *Social Media + Society*, vol. 3, no. 1, p. 205630511769154, 2017.

[10] Z. Tu, H. Cao, E. Lagerspetz, Y. Fan, H. Flores, S. Tarkoma, P. Nurmi, and Y. Li, "Demographics of mobile app usage: Long-term analysis of mobile app usage," *CCF Transactions on Pervasive Computing and Interaction*, pp. 1–18, 2021.

[11] T. Li, M. Zhang, H. Cao, Y. Li, S. Tarkoma, and P. Hui, ""what apps did you use?": Understanding the long-term evolution of mobile app usage," in *Proceedings of The Web Conference 2020*, ser. WWW '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 66–76. [Online]. Available: https://doi.org/10.1145/3366423.3380095

[12] A. Rahmati, C. Shepard, C. Tossell, L. Zhong, and P. Kortum, "Practical context awareness: Measuring and utilizing the context dependency of mobile usage," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1932–1946, 2015.

[13] W. R. Tseng and K. W. Hsu, "Smartphone app usage log mining," *International Journal of Computer & Electrical Engineering*, vol. 6, no. 2, pp. 151–156, 2014.

[14] H. Han and R. Li, "A practical compartmentation approach for the android app coexistence," in *Proceedings of the 2017 International Conference on Cryptography, Security and Privacy*, ser. ICCSP '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 49–55. [Online]. Available: https://doi.org/10.1145/3058060.3058073

[15] T. Li, Y. Li, M. A. Hoque, T. Xia, S. Tarkoma, and P. Hui, "To what extent we repeat ourselves? discovering daily activity patterns across mobile app usage," *IEEE Transactions on Mobile Computing*, 2020.

[16] S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart, "Representation learning for dynamic graphs: A survey," 2019.

[17] J. Skarding, B. Gabrys, and K. Musial, "Foundations and modelling of dynamic networks using dynamic graph neural networks: A survey," 2020.

[18] F. Manessi, A. Rozza, and M. Manzo, "Dynamic graph convolutional networks," *Pattern Recognition*, 2019.

[19] D. M. Dunlavy, T. G. Kolda, and E. Acar, "Temporal link prediction using matrix and tensor factorizations," *ACM Trans. Knowl. Discov. Data*, vol. 5, no. 2, Feb. 2011.

[20] R. Bian, Y. Koh, G. Dobbie, and A. Divoli, "Network embedding and change modeling in dynamic heterogeneous networks," in *the 42nd International ACM SIGIR Conference*, 2019.

[21] C. Hong and L. Miao, "Mining smartphone data for app usage prediction and recommendations: A survey," *Pervasive & Mobile Computing*, vol. 37, pp. 1–22, 2017.

[22] Z. Tu, Y. Li, P. Hui, L. Su, and D. Jin, "Personalized mobile app prediction by learning user's interest from social media," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.

[23] Y. Xu, Y. Zhu, Y. Shen, and J. Yu, "Leveraging app usage contexts for app recommendation: a neural approach," *World Wide Web*, no. 8, pp. 1–25, 2018.

[24] H. Yin, C. Liang, W. Wang, X. Du, Q. V. H. Nguyen, and X. Zhou, "Mobi-sage: A sparse additive generative model for mobile app recommendation," in *IEEE International Conference on Data Engineering*, 2017.

[25] S. Zhao, J. Ramos, J. Tao, Z. Jiang, S. Li, Z. Wu, G. Pan, and A. K. Dey, "Discovering different kinds of smartphone users through their application usage behaviors," in *Proc. ACM UbiComp*, 2016, pp. 498–509.

[26] Z. Yan, L. Wei, Y. Lu, Z. Wu, and B. Tao, "You are what apps you use: Transfer learning for personalized content and ad recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, ser. RecSys '17, 2017.

[27] B. Liu, D. Kong, C. Lei, N. Z. Gong, and X. Hui, "Personalized mobile app recommendation: Reconciling app functionality and user privacy preference," in *Eighth Acm International Conference on Web Search & Data Mining*, 2015.

[28] H. Zhu, X. Hui, G. Yong, and E. Chen, "Mobile app recommendations with security and privacy awareness," 2014.

[29] H. Zhu, H. Xiong, Y. Ge, and E. Chen, "Discovery of ranking fraud for mobile apps," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 1, pp. 74–87, 2015.

[30] M. Rahman, M. Rahman, B. Carbunar, and D. H. Chau, "Search rank fraud and malware detection in google play," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1329–1342, 2017.

[31] 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.

[32] S. Zhao, Z. Luo, Z. Jiang, H. Wang, F. Xu, S. Li, J. Yin, and G. Pan, "Appusage2vec: Modeling smartphone app usage for prediction," 2012.

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

[34] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver, "Tfmap: optimizing map for top-n context-aware recommendation," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 155–164.

[35] M. Böhmer, L. Ganev, and A. Krüger, "Appfunnel: A framework for usage-centric evaluation of recommender systems that suggest mobile applications," in *Proceedings of the 2013 international conference on Intelligent user interfaces*. ACM, 2013, pp. 267–276.

[36] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer, "Falling asleep with angry birds, facebook and kindle: A large scale study on mobile application usage," in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 47–56. [Online]. Available: https://doi.org/10.1145/2037373.2037383

[37] R. Telama, X. Yang, J. Viikari, I. Välimäki, O. Wanne, and O. Raitakari, "Physical activity from childhood to adulthood: a 21-year tracking study," *American journal of preventive medicine*, vol. 28, no. 3, pp. 267–273, 2005.

[38] J. D. Coie and K. A. Dodge, "Continuities and changes in children's social status: A five-year longitudinal study," *Merrill-Palmer Quarterly (1982-)*, pp. 261–282, 1983.

[39] T. H. Ollendick, M. D. Weist, M. C. Borden, and R. W. Greene, "Sociometric status and academic, behavioral, and psychological adjustment: a five-year longitudinal study." *Journal of consulting and clinical psychology*, vol. 60, no. 1, p. 80, 1992.

[40] F. Kooti, H. Yang, M. Cha, K. P. Gummadi, and W. A. Mason, "The emergence of conventions in online social networks," in *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[41] G. Dror, D. Pelleg, O. Rokhlenko, and I. Szpektor, "Churn prediction in new users of yahoo! answers," in *Proceedings of the 21st International Conference on World Wide Web*. ACM, 2012, pp. 829–834.

[42] N. Ducheneaut, N. Yee, E. Nickell, and R. J. Moore, "The life and death of online gaming communities: a look at guilds in world of warcraft," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2007, pp. 839–848.

[43] H. Cao, C.-J. Lee, S. Iqbal, M. Czerwinski, P. N. Wong, S. Rintel, B. Hecht, J. Teevan, and L. Yang, "Large scale analysis of multitasking behavior during remote meetings," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–13.

[44] C. Danescu-Niculescu-Mizil, R. West, D. Jurafsky, J. Leskovec, and C. Potts, "No country for old members: User lifecycle and linguistic change in online communities," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 307–318.

[45] M. Ros, M. Pegalajar, M. Delgado, A. Vila, D. T. Anderson, J. M. Keller, and M. Popescu, "Linguistic summarization of long-term trends for understanding change in human behavior," in *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*. IEEE, 2011, pp. 2080–2087.

[46] P. Gordon-Larsen, M. C. Nelson, and B. M. Popkin, "Longitudinal physical activity and sedentary behavior trends: adolescence to adulthood," *American journal of preventive medicine*, vol. 27, no. 4, pp. 277–283, 2004.

[47] J. Preece and B. Shneiderman, "The reader-to-leader framework: Motivating technology-mediated social participation," *AIS transactions on human-computer interaction*, vol. 1, no. 1, pp. 13–32, 2009.

[48] L. Wu, Y. Ge, Q. Liu, E. Chen, R. Hong, J. Du, and M. Wang, "Modeling the evolution of users' preferences and social links in social networking services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1240–1253, 2017.

[49] B. NIU, Y. Chen, Z. Wang, f. li, B. Wang, and H. Li, "Eclipse: Preserving differential location privacy against long-term observation attacks," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[50] T. Anwar, C. Liu, H. L. Vu, M. S. Islam, and T. Sellis, "Capturing the spatiotemporal evolution in road traffic networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1426–1439, 2018.

[51] D. Yang, R. Kraut, T. Smith, E. Mayfield, and D. Jurafsky, "Seekers, providers, welcomers, and storytellers: Modeling social roles in online health communities," in *ACM CHI Conference on Human Factors in Computing Systems*. ACM, 2019.

[52] K. Hata, R. Krishna, L. Fei-Fei, and M. Bernstein, "A glimpse far into the future: Understanding long-term crowd worker quality," in *CSCW: Computer-Supported Cooperative Work and Social Computing*, 2017.

[53] A. Shameli, T. Althoff, A. Saberi, and J. Leskovec, "How gamification affects physical activity: Large-scale analysis of walking challenges in a mobile application," in *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 455–463.

[54] T. Althoff, P. Jindal, and J. Leskovec, "Online actions with offline impact: How online social networks influence online and offline user behavior," in *Proceedings of the tenth ACM international conference on web search and data mining*. ACM, 2017, pp. 537–546.

[55] J. Tang, H. Gao, A. D. Sarma, Y. Bi, and H. Liu, "Trust evolution: Modeling and its applications," *IEEE Transactions on Knowledge & Data Engineering*, vol. 27, no. 6, pp. 1–1, 2015.

[56] Z. Lin, T. Althoff, and J. Leskovec, "I'll be back: On the multiple lives of users of a mobile activity tracking application," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, 2018, pp. 1501–1511.

[57] H. Wang, H. Li, and Y. Guo, "Understanding the evolution of mobile app ecosystems: A longitudinal measurement study of google play," in *The World Wide Web Conference*, ser. WWW '19.

[58] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," 2016.

[59] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2015, p. 1067–1077. [Online]. Available: https://doi.org/10.1145/2736277.2741093

[60] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[61] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017.

[62] R. A. Rossi, R. Zhou, and N. K. Ahmed, "Deep inductive graph representation learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 438–452, 2020.

[63] H. Cao, F. Xu, J. Sankaranarayanan, Y. Li, and H. Samet, "Habit2vec: Trajectory semantic embedding for living pattern recognition in population," *IEEE Transactions on Mobile Computing*, 2019.

[64] C. Zhang, K. Zhang, Q. Yuan, H. Peng, Y. Zheng, T. Hanratty, S. Wang, and J. Han, "Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17, 2017, pp. 361–370.

[65] X. Chen, Y. Wang, J. He, S. Pan, Y. Li, and P. Zhang, "Cap: Context-aware app usage prediction with heterogeneous graph embedding," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 1, Mar. 2019. [Online]. Available: https://doi.org/10.1145/3314391

[66] Y. Zhou, S. Li, and Y. Liu, "Graph-based method for app usage prediction with attributed heterogeneous network embedding," *Future Internet*, vol. 12, no. 3, p. 58, 2020.

[67] H. Tian, M. Zhang, X. Luo, F. Liu, and Y. Qiao, "Twitter user location inference based on representation learning and label propagation," in *Proceedings of The Web Conference 2020*, ser. WWW '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 2648–2654. [Online]. Available: https://doi.org/10.1145/3366423.3380019

[68] H. Shi, Y. Li, H. Cao, X. Zhou, C. Zhang, and V. Kostakos, "Semantics-aware hidden markov model for human mobility," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2019.

[69] Y. Lu, X. Wang, C. Shi, P. S. Yu, and Y. Ye, "Temporal network embedding with micro- and macro-dynamics," 2019.

[70] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. B. Schardl, and C. E. Leiserson, "Evolvegcn: Evolving graph convolutional networks for dynamic graphs," 2019.

[71] J. Liu, C. Xu, C. Yin, W. Wu, and Y. Song, "K-core based temporal graph convolutional network for dynamic graphs," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.

[72] Z. Zhang, Z. Li, H. Liu, and N. N. Xiong, "Multi-scale dynamic convolutional network for knowledge graph embedding," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.

[73] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk:online learning of social representations," in *Acm Sigkdd International Conference on Knowledge Discovery & Data Mining*, 2014.

[74] J. M. F. T. Berge, "Orthogonal procrustes rotation of two or more matrices," *Psychometrika*, vol. 42, no. 2, pp. 267–276, 1977.

[75] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-Simulation and Computation*, vol. 3, no. 1, pp. 1–27, 1974.

[76] V. D. M. Laurens and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2605, pp. 2579–2605, 2008.

[77] V. Uher, P. Gajdoš, and T. Ježowicz, "Solving nearest neighbors problem on gpu to speed up the fruchterman-reingold graph layout algorithm," in *IEEE International Conference on Cybernetics*, 2015.

[78] T. Fruchterman and M. Reingold, "Graph drawing by force-directed placement," 1997.

[79] N. Linge and A. Sutton, "The road to 4g," *Journal*, vol. 8, pp. 9–16, 2014.

[80] C. Cho, "Mobile app creates one-stop shop for social networks," *Prweek*, 2013.

[81] J. S. Adelman, G. D. Brown, and J. F. Quesada, "Contextual diversity, not word frequency, determines word-naming and lexical decision times," *Psychol Sci*, vol. 17, no. 9, pp. 814–823, 2010.

[82] S. Crossley, T. Salsbury, and D. Mcnamara, "The development of polysemy and frequency use in english second language speakers," *Language Learning*, vol. 60, no. 3, pp. 573–605, 2010.

[83] W. L. Hamilton, J. Leskovec, and J. Dan, "Diachronic word embeddings reveal statistical laws of semantic change," 2016.

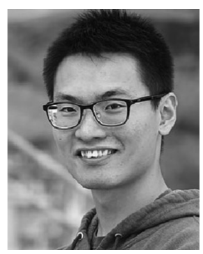
[84] D. J. Watts and S. H. Strogatz, "Collective dynamics of small world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[85] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *TACL*, vol. 3, pp. 211–225, 2015.

Yali Fan received B.E. and M.E. degrees from School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China, in 2018 and 2021. Her research interests include user behavior modeling and big data mining.

Zhen Tu received her B.S. degree in Electronics and Information Engineering and a second B.S. degree in Economics both from Wuhan University, Wuhan, China, in 2016, and currently she is working towards the master degree in electronic engineering department of Tsinghua University, Beijing, China. Her research interests include mobile big data mining, user behavior modeling, data privacy and security.

Tong Li received the B.S. degree and M.S. degree in communication engineering from Hunan University, China, in 2014 and 2017. At present, he is a dual Ph.D. student at the Hong Kong University of Science and Technology and the University of Helsinki. His research interests include distributed systems, edge network, and data-driven network. He is an IEEE student member.



Hancheng Cao received the BS degree in electronic information science and technology from Tsinghua University, Beijing, China, in 2018, and he is currently working toward the PhD degree in Computer Science Department, Stanford University, Stanford, California. His research interests include data mining, ubiquitous computing, and computational social science.

Tong Xia received the B.S. degree in electrical engineering from School of Electrical Information, Wuhan University, Wuhan, China, in 2017 and the M.S. degree in big data from Department of Electronic Engineering, Beijing, China, in 2020. Her research interests include human mobility, mobile big data mining, user behavior modelling and urban computing.

Yong Li (M'09-SM'16) received the B.S. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2007 and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012. He is currently a Faculty Member of the Department of Electronic Engineering, Tsinghua University. His papers have total citations more than 6900.

Dr. Li has served as General Chair, TPC Chair, SPC/TPC Member for several international workshops and conferences, and he is on the editorial board of two IEEE journals. His papers have total citations more than 6900. Among them, ten are ESI Highly Cited Papers in Computer Science, and four receive conference Best Paper (run-up) Awards. He received IEEE 2016 ComSoc Asia-Pacific Outstanding Young Researchers, Young Talent Program of China Association for Science and Technology, and the National Youth Talent Support Program.

Xiang Chen (M'08) received the B.E. and Ph.D. degrees both from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2002 and 2008, respectively. From July 2008 to July 2012, he was with the Wireless and Mobile Communication Technology R&D Center, Research Institute of Information Technology in Tsinghua University. From August 2012 to December 2014, was with Tsinghua Space Center, School of Aerospace, Tsinghua University. Since January 2015, he serves as an associate professor at School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China. He is also with Research Institute of Tsinghua University in Shenzhen as a chief researcher (part-time). His research interests mainly focus on 5G wireless communications, wireless big data, Internet of Things, and software radio.

Lin Zhang (M'16, SM'20) received her Ph.D. degree from Sun Yat-sen University in 2003 in information and communication engineering. She joined the School of Information Science and Technology of Sun Yat-sen University in 2003 and has served as an associate professor from 2007. From 2008 to 2009, she was a visiting researcher with the Electrical and Computer Engineering Department, University of Maryland, College Park, United States. From 2016, she joined the School of Electronics and Information Technology of Sun Yat-sen University, and also cooperated with Shandong Provincial Key Lab. of Wireless Communication Technologies. Her research has been supported by National Natural Science Foundation of China and the Science and Technology Program Project of Guangdong Province. Her current research interests are in the area of signal processing and their applications to wireless communication systems.