Information Sciences 549 (2021) 328–346

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Random walk based distributed representation learning and prediction on Social Networking Services

Junwei Li [a], Le Wu [a,*], Richang Hong [a], Jinkui Hou [b,*]

[a] Hefei University of Technology, China
[b] Weifang University, China

ARTICLE INFO

Article history:
Received 17 June 2018
Received in revised form 14 October 2020
Accepted 16 October 2020
Available online 23 October 2020

Keywords:
Random walk
Distributed representation learning
Social behavior
Consumption behavior

ABSTRACT

Social Networking Services (SNSs) provide online platforms for users with two kinds of behavior: user-user social behavior (e.g., following a user, making friends with others) and user-item consumption behavior (e.g., rating, showing likeness, clicking, giving thumbs up to items). With the increasing popularity of SNSs and demand for SNS features, predicting potential social links and recommending preferable items to users have become two hot research lines. However, previous works either modeled just one of these two kinds of behaviors in isolation or only considered the observed user behavior data. In fact, social scientists have long recognized that the user-user and user-item behaviors have a mutual reinforcement effect. On the one hand, the two behaviors have correlations, and they can influence each other. On the other hand, due to the sparsity of the observed user behavior data, the user behavior prediction performance is far from satisfactory, although, using two types of behavioral data at the same time can mitigate the sparsity problem. These two problems remains open: how to better model the correlation of user-user social and user-item consumption activities and how to mitigate the data sparsity issue. In this paper, we propose a random walk based distributed representation learning model to jointly predict these behaviors on SNSs. Specifically, we first construct a joint behavior graph that combines the two behaviors, with the edges denoting the sparse observed user behavior data. Then, we adopt a random walk to capture higher-order relationships between users and items. After that, we utilize a distributed learning approach to embed both users and items into a latent space. In this way, the behavior prediction tasks are transformed into similarity calculations in the latent space. Finally, extensive experimental results using two real-world datasets demonstrate the effectiveness of our proposed approach on the two behavior prediction tasks.

1. Introduction

With the popularity of online social networks and online services, Social Networking Services (SNSs) have become increasingly popular in modern daily life. SNSs provide online platforms that focus on facilitating the building of social relationships among people who share similar item preferences or online social connections. Specifically, users perform two kinds of behavior on these platforms: user-item consumption behavior (reflected in user-item interaction behavior, e.g., rating, showing likeness, clicking, giving thumb up to items), and user-user social behavior (e.g., following users, making friends

* Corresponding authors.

https://doi.org/10.1016/j.ins.2020.10.045
0020-0255/© 2020 Published by Elsevier Inc.

with others). For example, on the popular online social product review platform *Epinions.com*, users have followers, and the users' reviews are displayed on the followers' front pages. One of the most popular social networking services in China, *Douban*, provides a platform for users to make online connections and share their preferences of movies, books, and other media. On these platforms, users prefer to connect with others that have similar tastes. Likewise, users tend to consume items that are shared by their social friends. With the mutual influence of users' two kinds of behaviors, SNSs evolve with prosperity.

In the real world, as there are hundreds of millions of users and items on SNSs, information overload has become a serious problem preventing users from identifying potential social relationships and possibly interesting items. Therefore, recommending items that users may be interested in and predicting potential social relationships have become two hot research topics. As one of the most successful approaches for item recommendation [1], collaborative filtering (CF) infers each user's future consumption preference by mining historical observed user-item interactions from like-minded users. Among all CF models, latent factor based models are widely used because of their relatively high performance [2]. These models project users and items embeddings into a lower latent space, and then the preference of a user for an item is measured by their inner product. For social behavior, the observed relationships among users are also widely used for social recommendation [3,4].

The above classical models considered the users' two kinds of behavior in isolation. However, sociologists have long converged on the opinion that the users' social behavior and consumption behavior are highly correlated with each other. Specifically, the social influence theory proposes that the users' consumption behavior is affected by their social relationships, leading to similar consumption interests among socially connected users [5,3,4]. Similarly, the homophily effect of consumption behavior shows that users like to connect with others who share common tastes [6,7]. Based on these theories, many researchers have proposed leveraging one type of behavior as side information to enhance the other. For example, Tang et al. proposed a method that exploits users' consumption behavior to assist social trust prediction [7]. Also, social recommendation is an active research topic in recommender systems, seeking to leverage social information to enhance consumption behavior prediction [8,9]. Furthermore, since users perform these two behaviors on SNSs and due to the mutual reinforcement between them, in recent years, researchers have proposed unified frameworks to jointly predict these kinds of user behavior [10,11].

Although these models showed more promising performance than those modeling users' behaviors in isolation, most of the proposed classical methods rely heavily on the observed user-item and user-user links. As each user connects with only a few social users and interacts with only several items among many, these methods still suffer from the data sparsity problem, which is a key challenge in the user behavior prediction task. In fact, beyond the observed links, there are many complex, higher-order, user-user and user-item links. Consider these examples: (a) Two unlinked users have bought the same item, so they may be highly correlated. The connection $user_1$ - $item_1$ - $user_2$ implies that $user_1$ and $user_2$ may become friends due to their similar preferences. However, this hidden relationship is not able to be observed in the original data. (b) People prefer to buy things based on word of mouth. The connection $user_1$ - $user_2$ - $item_1$ hints that $user_1$ may prefer $item_1$ to items not recommended by $user_2$. These complex relationships cannot be modeled only by the observed user behavior links. Although researchers try to acknowledge these situations by carefully designing models, ignoring transitive unobserved higher-order relationships sometimes results in suboptimal predictions of user behaviors.

To this end, in this paper, we propose a distributed learning approach based on random walks to better model user-user and user-item kinds of behaviors. The key idea of the proposed approach is to adopt short random walks, which can correlate users' two kinds of behavior and generate semantic node sequences. Furthermore, these walks help to alleviate the sparsity problem. Specifically, because the two behaviors have a mutual reinforcement relationship, we first construct a joint behavior graph that combines them, with edges denoting the sparse observed user behavior data. Then, by adopting random walks on the joint behavior graph, we extend relations for each node and generate higher-order links. After that, we employ the language model Skipgram [12] and embed both users and items into the same latent representation space for behavior prediction.

In summary, our contributions are as follows:

- We argue that as most user behavior prediction models only exploit the sparse and observed user-user and user-item relations, the data sparsity issue is still a challenge for enhancing prediction performance.
- We propose a random walk based distributed learning approach to model and predict user-user and user-item behaviors on SNSs. The proposed method mutually enhances the two prediction tasks with a joint graph. Besides, this method mitigates the data sparsity issue by extending the relations and capturing unobserved higher-order relationships with truncated random walks.
- We obtain extensive experimental results based on two real-world datasets. The experimental results clearly show the effectiveness of our proposed model, especially in cold-start scenarios.

2. Preliminaries

For ease of presentation and understanding, we use uppercase calligraphic symbols (e.g., $\mathcal{G}$) to denote graphs or networks, boldface lowercase symbols (e.g., $\mathbf{a}$) to denote nodes in a graph, italic boldface lowercase symbols (e.g., $\boldsymbol{a}$) to denote vectors, italic uppercase symbols (e.g., A) to denote sets, italic uppercase symbols T to denote a random walk sequence set (with the subscript representing the starting node and superscript i denoting the i-th step), and boldface uppercase symbols (e.g., $\mathbf{A}$) to denote matrices.

Deepwalk. DeepWalk is a graph embedding method for learning the representations of nodes in a graph $\mathcal{G} =< V, E >$, where V is the vertex set representing the nodes in the graph, and E is the observed edge set referring to the relationships among nodes. To simplify the representation, we utilize $f(\cdot)$ to denote the graph embedding operation. After that, each node can be represented as a low-dimensional vector, which preserves the topological information in the graph.

Specifically, the key idea of Deepwalk contains two parts. First, for each node in the graph, Deepwalk utilizes a random walk to generate many node sequences. For example, given the walk length k, a random walk sequence starting from node $\mathbf{a}$ is represented as $T^1_{\mathbf{a}}, T^2_{\mathbf{a}}, T^3_{\mathbf{a}}, \ldots, T^k_{\mathbf{a}}$.

Then, as words and sentences in texts are similar to nodes and generated sequences, Deepwalk adopts the language model Skipgram [12], which maximizes the co-occurrence probability among the appearing words within a flexible window length in a sentence, to model the relationships among nodes. For ease of understanding, Fig. 1 shows an example of SkipGram on a node sequence. In the sequence from node 0 to node 9, the window length of the blue box is seven, and the center red node has a background neighborhood with three length steps (considering both left and right). Thus the SkipGram model will maximize the co-occurrence probability from node 2 to node 8, and the center node with each neighbor node will be transformed into node pairs for training, e.g., (node 2, node 5), (node 3, node 5), etc. For more information, please refer to Skipgram.

3. Related work

Collaborative Filtering. As one of the most successful approaches of recommender systems, Collaborative Filtering (CF) models users' preferences by collecting the past historical record from the like-minded users with similar tastes. Typically, CF-based methods are divided into two categories: memory-based and model-based methods. Memory-based methods [20,27,21] utilize observed interactions to find neighborhood users and make the recommendation by gathering their ratings. Model-based methods [28,2,29,30,23] usually provide unified models with training samples to use to learn the model's parameters. Once the parameters are learned (i.e., once the model adapt correctly to the training samples), the model can directly output the results of the test data. Therefore, model-based methods do not store the training data, and so they are usually faster than Memory-based methods in the test phase. Nevertheless, nearly all CF methods rely on the observed user-item consumption records to make predictions. Due to the sparsity of the observed behavior data, the performance of these models is sometimes unsatisfactory.

Link Prediction. Given a snapshot of a social network, the task of link prediction refers to inferring potential new interactions among social members in the near future. Traditionally, unsupervised models calculate the predefined proximity between two nodes based on the topological information of a social network. For example, Node proximity based models utilize structural information to predict potential social links [13]. Path-based methods consider the length of the shortest path or the time cost to measure the proximity of indirectly connected users [13,14]. In contrast to the unsupervised methods, by treating observed links as positive samples while randomly selected unobserved links as candidate negative samples [31,32], many supervised methods achieve better performance [18,19].

Modeling User-user and User-item Behavior Simultaneously. Sociologists have long agreed that users' social behavior and consumption behavior are not independent and unrelated. On the one hand, the users' different behaviors can be utilized as side information to boost the prediction performance. For example, Tang et al. employed the homophily effect from consumption behavior for trust prediction [7]. Jamali et al. proposed SocialMF, which utilizes social neighborhood information to enhance collaborative filtering [5]. Zhao et al. leveraged social connections to improve personalized ranking [33], which extended Bayesian personalized Ranking [28] by considering the feedback from users' friends to model their preference. Moreover, with the popularity of deep learning, many works adopt deep neural networks and social information to enhance the consumption behavior prediction [34–37], which have shown the state-of-the-art performance. On the other hand, these two behaviors coexist and interact with each other on SNSs. Recently, many works have combined these behaviors to enhance performance. Yang et al. proposed a unified framework called FIP (Friendship and Interest Propagation) to model

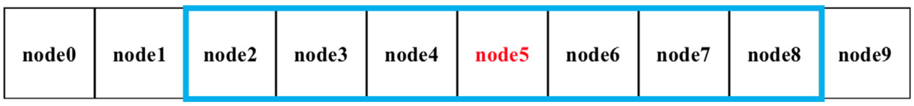


Fig. 1. A toy example of SkipGram. The center node 5 with its neighbor node in a sequence, SkipGram will maximize the co-occurrence probability from node 2 to node 8.

Table 1
Attributes of related works in specific application fields.

User Behavior Task	Model Attribute		Methods
Social behavior models	Pre-defined similarity on graph structure		[13,14]
	Network embedding		[15–17]
	Utilizing side information		[7]
	Matrix factorization		[18,19]
Consumption behavior models	Memory-based model		[20,21]
	Model-based model		[2,5,22,23]
Joint behavior models	Shallow matrix factorization model	Collaborative filtering + user profile + item attribute	[11]
		Collaborative filtering + Temporal information	[24,10].
	Deep neural model	Collaborative filtering	[25,26]

users' interests and friendships simultaneously [11]. It utilized the available profiles of users and items with the shared user latent representations. Besides, by considering temporal information, Wu et al. explored the evolution of users' preferences and social relationships on SNSs [24,10]. Moreover, there are methods jointly model two behavior recently [25,26]. For example, the recently proposed NJM (Neural Joint Modeling) jointly models dynamic user feedback and social links [25], which improves [24,10] by adopting neural networks, powerfully contributing to fitting more complex user-item interactions.

Graph Embedding. In recent years, many graph embedding methods have emerged. These methods have two clear goals. The first one is to reconstruct the relations among nodes, and the other is to preserve the topological information. Therefore, most of them not only model the first-order observed links but also consider the higher-order topological information. Typically, classic graph embedding methods aim at learning low-latent representations of nodes from the network structure, and the learned low-latent representations are convenient for subsequent applications. Besides, these embedding methods have been proved to be very effective on the link prediction task [15–17]. Specifically, since nodes in graphs are very similar to words in texts, Deepwalk [15] mapped the problem of word embedding into the social network. Grover et al. proposed an improved algorithm based on Deepwalk by modifying the random walk strategy [16]. Moreover, Tang et al. presented a large-scale node embedding algorithm that is suitable for arbitrary types of information networks [17].

We summarize the related works in Table 1.

4. Problem definition

On SNSs, given a set of N users $U = \{u_1, u_2, \ldots, u_N\}$, we define the social trust matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ to record the social relationships among users. We have s_{ab} = 1 if user a trusts or follows user b, otherwise it equals 0. Symbol "?" represents unknown social links. Similarly, given a set of M items $V = \{v_1, v_2, \ldots, v_M\}$, we define $\mathbf{R} \in \mathbb{R}^{N \times M}$ to represent the rating matrix, and the $(a,j)^{th}$ entry r_{aj} of $\mathbf{R}$ represents user u_a's rating of item v_j. For better understanding, we give a toy example of two behavioral graphs in Fig. 2.

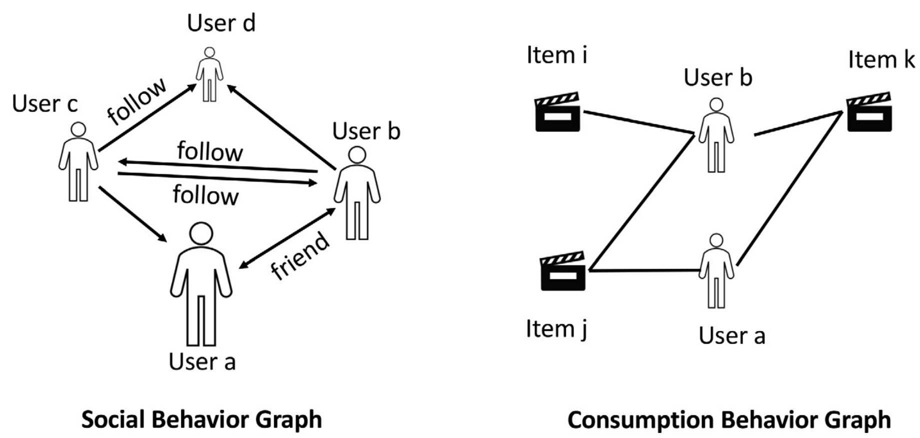


Fig. 2. A toy example of the Social Behavior Graph and the Consumption Behavior Graph. In the Social Behavior Graph, users follow others or make friends with other users. In the Consumption Behavior Graph, users have interactions with items.

Social behavior forms the connections between users. For example, in the social e-commerce platform Epinions, users follow other users, which constitutes a social behavior that is represented by a connection or link between the two. In the indirect social platform Facebook, two users have a social behavior if both of them agreed to begin the friendship with each other. Since these social behaviors link nodes, they naturally form a graph, leading to the following definition:

Definition 1 (*Social Behavior Graph.*). A *Social Behavior Graph (SBG)* is defined as a directed graph $\mathcal{G}_s = < U, \mathbf{S} >$, where U is the node set representing all users and $\mathbf{S}$ is the social trust matrix representing the social network; if user b is connected to user a because of a social behavior, then s_{ba} = 1, otherwise it equals zero.

Similarly, a consumption behavior refers to a connection between a user and an item. There are many types of consumption behavior. For example, on the community website Douban, users can rate movies with scores, or they can click and browse the movie.

Definition 2 (*Consumption Behavior Graph.*). A *Consumption Behavior Graph (CBG)* is defined as $\mathcal{G}_c = < U \cup V, \mathbf{R} >$, where U and V are the sets of users and items, respectively. $\mathbf{R}$ is the rating matrix containing the interactions between users and items. In the matrix, r_{aj}=1 denotes that user a is connected with item j because of a consumption behavior. Note that CBG is a bipartite graph; there are no direct links between two users or two items.

Based on the abovementioned definitions, our target goal is as follows:

Problem Definition. Given the Social Behavior Graph $\mathcal{G}_s$ and Consumption Behavior Graph $\mathcal{G}_c$, learn the distributed representations of users and items to predict the users' future user-user social behaviors and user-item consumption behaviors.

5. Random walk based distributed learning method

5.1. Joint behavior graph construction

According to social theories, the two kinds of user behavior we consider have a mutual reinforcement relationship. Users prefer to connect with others who have similar tastes. Likewise, users tend to consume items that are used or recommended by their social friends. Therefore, building a graph that contains both social behavior and consumption behavior should be beneficial for extracting correlations between them.

For this purpose, we follow the idea of a Social-Attribute Network (SAN) [38], which was proposed for link and attribute prediction [39,40]. In SAN, a social user not only has social links with other users but also is correlated with attribute nodes by attribute links (e.g., a user with various employers like Google, Intel, and Yahoo and various schools like Berkeley, Stanford, and Yale).

Therefore, we build a new augmented graph $\mathcal{G}_J$ called the *Joint Behavior Graph* (JBG). Specifically, we merge the Social Behavior Graph (SBG) and Consumption Behavior Graph (CBG) by sharing the same users. Then, for each user in the JBG, there are many social links with other social users, as well as connections to items, which are called attribute links.

Definition 3 (*Joint Behavior Graph.*). A *Joint Behavior Graph (JBG) is defined as* $\mathcal{G}_J = < U \cup V, \mathbf{S}, \mathbf{R} >$, where U and V are the sets of users and items, respectively.

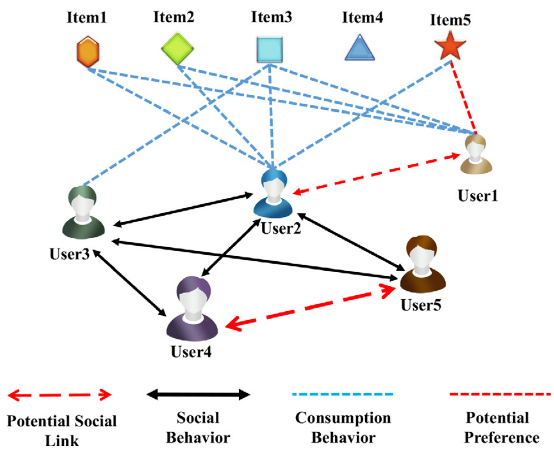


Fig. 3. A toy example of a Joint Behavior Graph including both social behavior and consumption behavior. $user_4$ and $user_5$ may become friends, as they share similar neighborhood friendships ($user_2$ and $user_3$). $user_1$ has a high probability of consuming $item_5$, as $user_1$ and $user_2$ have the same taste ($item_1, item_2, item_3$). $user_1$ is a cold-start user in the social network; cold-start users do not have any social relations, so it is difficult to predict their social links. However, the side information from consumption behavior indicates that $user_1$ and $user_2$ are similar.

For clarity, we show a simple toy example of Joint Behavior Graph with undirected friendships in Fig. 3.

Obviously, the JNB combines the information of a social trust matrix **S** and rating matrix **R** into a unified network structure. We argue that the advantages of JBG are manifested in two ways:

(1) Social and consumption behaviors have a mutual reinforcement relationship and can be deemed as side information for each other, so the fusion of the two behavior graphs is a way for them to complement each other and mitigate data sparsity and cold-start problem. The term "cold-start" refers to the phenomenon that when many users have limited behavior data, it is hard to recommend users and items for them. For example, in Fig. 3, as a cold-start user in the social network, $user_1$ does not have any social relationships. However, a relationship between $user_1$ and $user_2$ is likely as they have the same taste ($item_1, item_2, item_3$). In addition, cold-start $user_5$ does not have any consumption records, but they are likely to prefer to $item_3$ because all of $user_5$'s social neighbors ($user_2$ and $user_3$) have interactions with $item_3$.

(2) Graph structure provides connectivity information and serendipitous unobserved relationships among nodes. For instance, $user_1$ and $user_3$ have a high approximate similarity, because both have bought $item_3$. Also, $user_4$ and $user_5$ share the same neighbors ($user_2$ and $user_3$). Thus, they are likely to become friends in the future. These complex relationships are not observable in the original SBG and CBG.

However, as we only have observed links, capturing these complex user-user and user-item higher-order relationships is still a problem.

5.2. Generating semantic random walk sequences

In order to extract more complex user-user and user-item interactions from the original data, we resort to the random walk strategy, which describes a path that consists of a succession of random steps. For example, as shown in Fig. 3, (1) although $user_5$ does not have any consumption records, with the random walk sequence $user_5 \leftrightarrow user_2 \leftrightarrow item_3$, the hidden link between $user_5$ and $item_3$ is captured. This hidden link cannot be observed in the original CBG. Similarly, $user_1$ is a cold-start user in the social network. By adopting random walks, we can recommend $user_2$ ($user_1 \leftrightarrow item_3 \leftrightarrow user_2$), $user_3$ ($user_1 \leftrightarrow item_3 \leftrightarrow user_3$), and even $user_4$ ($user_1 \leftrightarrow item_3 \leftrightarrow user_3 \leftrightarrow user_4$) to $user_1$. Obviously, we extend the relations and capture many unobserved hidden relationships for $user_5$ and $user_1$ from random walks.

However, in some cases, several paths exist which are completely different from each other. For instance, for $user_5$, there is another path $user_5 \leftrightarrow user_3 \leftrightarrow item_3$, which means the unobserved hidden relationships among nodes cannot be captured by only one path. We assume that different paths may lead to different results, because these random walk paths/sequences have rich semantics. Specifically, (1) $user_1 \leftrightarrow item_3 \leftrightarrow user_2$ means $user_1$ and $user_2$ have the same taste for $item_3$. (2) If $user_5$ eventually bought $item_3$, they may be influenced by $user_2$ ($user_5 \leftrightarrow user_2 \leftrightarrow item_3$) or by $user_3$ ($user_5 \leftrightarrow user_3 \leftrightarrow item_3$). Based on this, we need to collect different paths to sufficiently explore the relationships for each node.

The above strategy is similar to a breadth-first traversal. Meanwhile, for each node, we adopt truncated random walks with a fixed path length, which could be deemed as a depth-first traversal. Through such a random walk mechanism, we combine the two behavioral graphs and generate many meaningful sequences.

5.3. Distributed representation learning

As nodes in a sequence resemble words in a sentence, we adopt SkipGram [12] on the generated sequences from random walks to project each node $\mathbf{n}$ into a low-dimensional space $f(\mathbf{n}) \in \mathbb{R}^d$. Specifically, for each length k sequence $T = \{T^1_{\mathbf{n}_b}, T^2_{\mathbf{n}_b}, \ldots, T^k_{\mathbf{n}_b}\}$ that starts from $\mathbf{n}_b$, the approximate conditional probability based on assuming independence is:

$$\prod_{-w \leqslant j \leqslant w, j \neq 0} \Pr(T^{i+j}_{\mathbf{n}_b} | T^i_{\mathbf{n}_b}), \tag{1}$$

where w is the window size, in which the center node with each neighbor node will be transformed into node pairs for training:

$$\Pr(T^{i+j}_{\mathbf{n}_b} | T^i_{\mathbf{n}_b}) = \frac{sim(f(T^{i+j}_{\mathbf{n}_b}), f(T^i_{\mathbf{n}_b}))}{\sum_{j' \in V} sim(f(T^{j'}_{\mathbf{n}_b}), f(T^i_{\mathbf{n}_b}))}, \tag{2}$$

Here, sim is a method to calculate the similarity between two vectors.

Since there are N users and M items, the number of total nodes are $|\mathrm{M}+\mathrm{N}|$. For each node, we utilize random walks to generate $|\mathrm{L}|$ sequences. Then, our goal is to optimize all the sequences as:

$$\max \prod_{T=1}^{|\mathrm{H}|} \prod_{-w \leqslant j \leqslant w, j \neq 0} \Pr(T^{i+j}_{\mathbf{n_b}} | T^i_{\mathbf{n_b}}), \tag{3}$$

where $|\mathrm{H}| = |\mathrm{M}+\mathrm{N}| \ * \ |\mathrm{L}|$ denotes the total number of generated sequences.

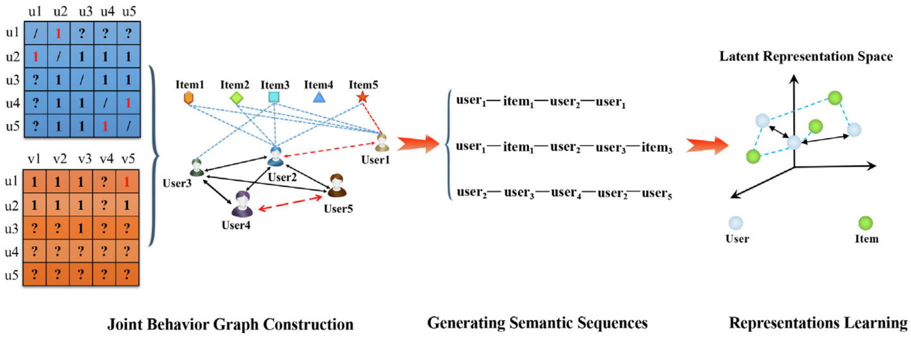


Fig. 4. Flow chart of the process. We first combine the two kinds of behavior and obtain the Joint Behavior Graph. Then, we utilize the random walk mechanism to collect meaningful and semantic sequences. Finally, we learn the representations for each node and measure the similarity among them.

For optimization, we utilize Stochastic Gradient Descent (SGD) [41] with mini batch strategy to minimize the loss function. Then, the learned representations contain both of users and items. To determine similarity between two nodes, we utilize the Euclidean distance:

$$sim(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}. \tag{4}$$

Finally, we summarize the flow chart of the proposed method in Fig. 4. In this way, we transform the behavior prediction tasks into similarity calculations and ranking tasks.

5.4. Complexity analysis

In this section, we analyze the computational complexity and space complexity of the proposed method. Our method generates many sequences (a total of $|\mathrm{H}|$ sequences) from each node since each node with all its neighbors in the window size w will be transformed into node pairs for training ($O(|\mathrm{H}| \cdot 2w)$). Besides this, we employ Eq. 2 to model the probability, so we need to consider all the nodes in the Joint Behavior Graph. Summing these, the computational complexity is $O(|\mathrm{H}| \cdot 2w \cdot |\mathrm{M}+\mathrm{N}|)$. As for the space complexity, we only utilize one vector to represent each node. Thus, the space complexity is $O(|\mathrm{M}+\mathrm{N}|)$.

6. Experiments

In this section, we conduct experiments designed to answer the following three key questions, which aim at certifying the effectiveness of our proposed methods:

RQ1 How does our proposed method perform compared to other methods that focus on a single task?
RQ2 Is the proposed method useful to alleviate the cold start problem?
RQ3 How do the key parameters impact the performance of the method?

6.1. Experimental Settings

Dataset Description. We conduct experiments on two real-world datasets: Epinions and Douban. Epinions is a who-trusts-whom online product sharing platform, and Douban is a community website. Both of them include trust data (social behavior) and rating data (consumption behavior). Trust data utilizes a triple to describe each social trust relationship. For example, if user a trusts user b, then the record triple is defined as $\{a\ b\ 1\}$. Rating data records the users' consumption behavior with a score (1–5). We consider explicit feedback or implicit feedback according to different baselines on consumption behavior prediction. For explicit feedback, if a rating score is greater than or equal to 3, it remains unchanged; otherwise, the score is 0. For implicit feedback, the records with rating scores greater than or equal to 3 are treated as positive samples (positive sample with label 1), and the others are 0.

For trust and rating data on the two datasets, we randomly select some records as the training set and validation set, and the remaining data is the test set. Table 2 shows the characteristics of the two datasets after pruning. Because many methods utilize negative samples to form counterparts in the training process, each observed interaction is paired with some negative instances.

Evaluation Metrics. When we evaluate the performance on the test set, we randomly select some negative samples in the item set for each user. Then, we mix together negative samples and positive samples (in the test set) to select the top K

Table 2
The dataset after pruning.

Dataset	Epinions	Douban
Max User ID	9,713	6,395
Max Item ID	139,738	59,701
Number of Users	9,713	6,395
Number of Items	91,110	20,308
Total Consumptions	347,064	214,160
Total Links	328,935	142,941
Training Consumptions	277,505	147,678
Test Consumptions	62,603	59,833
Validation Consumptings	6,956	6,649
Training Links	266,671	121,823
Test Links	55,415	18,796
Validation Links	6,849	2,322
Consumption Sparsity	99.98%	99.89%
Link Sparsity	99.72%	99.34%

potential candidates. In this way, we alleviate the time-consuming problem of ranking all users or items for each user during evaluation. We repeated the experiment five times and report the average results.

For the evaluation metrics, we adopt four widely used metrics in link prediction and recommendation, *precision@k*, *recall@k*, *F1-score@k* [42], and *Normalized Discounted Cumulative Gain (NDCG@k)* [43]. The *precision@k* represents the percentage of objects which are successfully predicted to be in the top-k list, and *recall@k* is the percentage of objects that are successfully predicted relative to the total number of positive samples in the top-k list. *F1-score@k* is a trade-off result based on *precision@k* and *recall@k*. *NDCG@k* is a precision-based measure that accounts for the relative accuracy of the predicted positions of the positive instances. Larger values indicate better performance for all of these metrics.

Baselines. To evaluate the effectiveness of our proposed method, we introduce baselines and compare the performance on two prediction tasks. For social behavior prediction, we evaluate the performance of our method with the following methods:

Adamic/Adar(AA) [13]. AA is an unsupervised measure for link prediction. For two users, we define the similarity by the number of users that both of them follow.

FIP [11]. This method presents a framework considering both friendship prediction and interest targeting simultaneously.

NMF [44]. Nonnegative Matrix Factorization decomposes the social trust matrix **S** into two low-rank matrices. Then, it predicts the possible trust relationships by using matrix multiplication.

LINE [17]. LINE is a graph embedding method. It preserves the first-order neighborhood information and second-order topology information by an elaborately designed objective function. The learned node representations have a form that is convenient for subsequent applications.

hTrust [7]. hTrust utilizes the homophily effect to improve the performance of trust prediction. The homogeneity information comes from the rating data; thus, hTrust utilizes trust data and rating data.

As for consumption behavior prediction, we compare our method with the following methods:

PMF [2]. PMF is a method utilizing matrix factorization to project users and items into lower latent spaces. PMF utilizes explicit feedback to predict user preferences.

BPR [28]. BPR optimizes matrix factorization (MF) with a pairwise ranking loss. Also, BPR is a highly competitive baseline for item recommendation and is more precise for the ranking task.

SocialMF [5]. This method incorporates social information when modeling users' preferences. This method is used only for social recommendation. SocialMF utilizes explicit feedback to predict users' preferences.

FIP [11]. This method presents a joint model to performed friendship prediction and interest targeting simultaneously. FIP utilizes explicit feedback to predict users' preferences.

LINE [17]. LINE preserves the first-order and second-order proximity by an elaborately designed objective function. The learned node representations are in a convenient form for subsequent applications.

We have thoughtfully chosen the above baselines to cover a diverse range of social behavior prediction or consumption behavior prediction methods. PMF is a traditional model-based method utilizing point-wise loss. BPR is a competitive user-based collaborative filtering approach to evidence the state-of-the-art performance of recommendation from implicit feedback on the ranking task. Thus, we compare with BPR to prove the effectiveness. Then, as our method contains two prediction tasks, we introduce FIP for comparison (FIP can also predict the two behaviors simultaneously). Also, since two behavioral data complement each other, we adopt hTrust and SocialMF, which utilize side information to enhance their performance. Specifically, hTrust utilizes the homophily effect for trust prediction, while SocialMF focuses on social recommendation. To compare with another embedding method, we adopt LINE, which is a state-of-the-art graph embedding method for preserving the structure information. We employ NMF to compare with the traditional matrix factorization model.

Table 3
Social behavior prediction results for Epinions.

Metrics	Precision		Recall		F1-score		NDCG	
	Top-k							
Methods	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10
AA	0.286	0.228	0.267	0.371	0.276	0.282	0.362	0.375
NMF	0.365	0.294	0.400	0.572	0.381	0.388	0.498	0.531
FIP	0.332	0.270	0.368	0.535	0.349	0.358	0.453	0.488
hTrust	0.383	0.305	0.425	0.597	0.402	0.404	0.525	0.558
LINE	0.331	0.257	0.316	0.449	0.323	0.327	0.446	0.458
RWSBG	0.361	0.279	0.431	0.582	0.393	0.377	0.532	0.557
RWJBG	**0.408**	**0.306**	**0.498**	**0.644**	**0.449**	**0.415**	**0.604**	**0.624**

Table 4
Social behavior prediction results for Douban.

Metrics	Precision		Recall		F1-score		NDCG	
	Top-k							
Methods	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10
AA	0.138	0.100	0.194	0.263	0.161	0.145	0.208	0.225
NMF	0.204	0.149	0.300	0.416	0.243	0.219	0.317	0.348
FIP	0.225	0.162	0.355	0.476	0.275	0.242	0.367	0.400
hTrust	0.259	0.181	0.404	0.522	0.316	0.269	0.430	0.460
LINE	0.239	0.160	0.360	0.447	0.287	0.236	0.397	0.412
RWSBG	0.246	0.175	0.420	0.543	0.310	0.265	0.430	0.466
RWJBG	**0.274**	**0.187**	**0.450**	**0.565**	**0.341**	**0.281**	**0.467**	**0.495**

Table 5
Consumption behavior prediction results for Epinions.

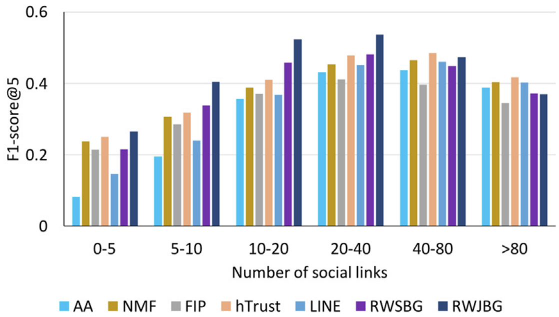
Metrics	Precision		Recall		F1-score		NDCG	
	Top-k							
Methods	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10
PMF	0.503	0.344	0.554	0.637	0.527	0.447	0.712	0.701
BPR	0.513	0.355	0.584	0.681	0.546	0.467	0.729	0.725
FIP	0.496	0.352	0.608	0.748	0.545	0.479	0.720	0.738
SocialMF	0.527	0.364	0.599	0.695	0.561	0.478	0.748	0.744
LINE	0.495	0.356	0.617	0.760	0.549	0.485	0.716	0.739
RWCBG	0.524	0.369	0.619	0.758	0.568	0.496	0.731	0.744
RWJBG	**0.552**	**0.381**	**0.654**	**0.797**	**0.599**	**0.516**	**0.753**	**0.764**

Table 6
Consumption behavior prediction results for Douban.

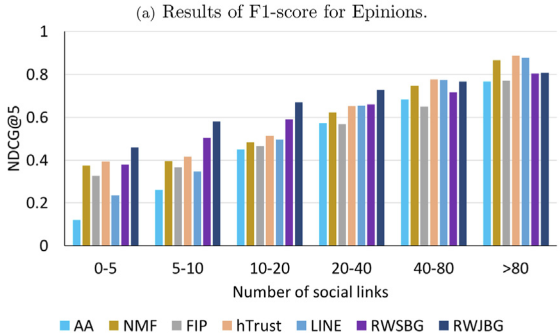
Metrics	Precision		Recall		F1-score		NDCG	
	Top-k							
Methods	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10	Top-5	Top-10
PMF	0.655	0.497	0.562	0.708	0.605	0.584	0.808	0.802
BPR	0.669	0.517	0.587	0.751	0.625	0.612	0.828	0.833
FIP	0.686	0.528	0.608	0.783	0.645	0.631	0.852	0.860
SocialMF	0.691	0.533	0.606	0.777	0.646	0.632	0.856	0.862
LINE	0.640	0.514	0.588	0.779	0.613	0.619	0.792	0.821
RWCBG	0.696	0.530	0.614	0.783	0.652	0.632	0.860	0.870
RWJBG	**0.705**	**0.541**	**0.624**	**0.795**	**0.662**	**0.644**	**0.872**	**0.877**

Table 7
Information about six groups according to trust data.

Number of occurrences	Epinions	Douban
0–5	3,261(34.2%)	1,790(28.5%)
5–10	1,756(18.4%)	1,479(23.6%)
10–20	1,614(16.9%)	1,447(23.1%)
20–40	1,286(13.5%)	927(14.8%)
40–80	850(9.0%)	430(7.0%)
>80	762(8.0%)	206(3.0%)



(a) Results of F1-score for Epinions.



(b) Results of NDCG on Epinions.

Fig. 5. Sparsity Analysis for Social Behavior Prediction for Epinions.

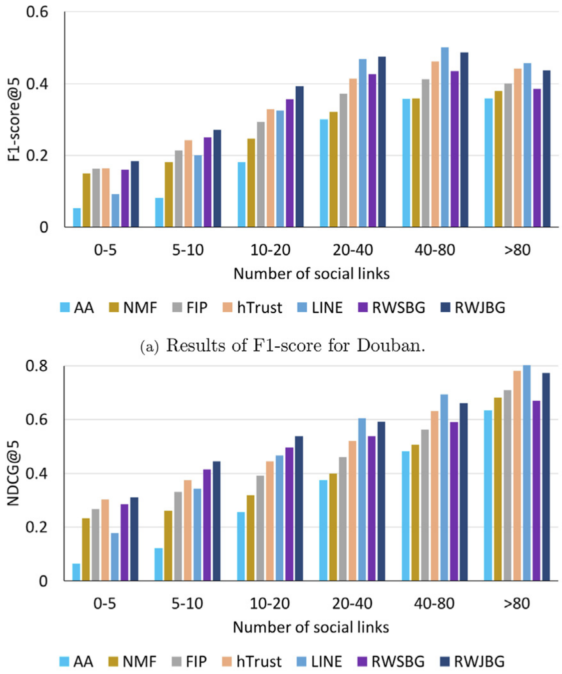
Finally, we chose AA because it is an unsupervised method, which computes the similarity between two nodes by topological properties.

6.2. Performance Evaluation and Comparison (RQ1)

6.2.1. Social behavior prediction

The average performance of the aforementioned social behavior prediction methods for Epinions is reported in Table 3, while the Douban result is reported in Table 4. RWSBG and RWJBG are the abbreviations of *Random Walk on Social Behavior Graph* and *Random Walk on Joint Behavior Graph*, respectively. We show the top-5 and top-10 results.

On both datasets, AA underperforms the other methods, as AA is only based on network structure with some predefined logical assumptions. FIP is a joint model, which shares the same user latent representation for social and consumption behavior. As we can see, FIP sometimes slightly underperforms other methods on these two datasets. A possible reason is that our datasets lack the profile information of users and items, which is different from the original settings in FIP. hTrust performs better than AA, FIP, NMF, and LINE on social behavior prediction, which shows the effectiveness of leveraging the homophily effect. Besides this, hTrust is better than RWSBG on the two datasets in some cases, because hTrust utilizes users' preference information while RWSBG only considers social behavior. LINE works well for link prediction; nevertheless, when the network is sparse and the average number of neighbors of a node is insufficient, the second-order proximity cannot add meaningful neighbors, making the results inaccurate. RWJBG considers both social behavior and consumption behavior information to learn the representations. As expected, RWJBG performs well on both datasets.



(a) Results of F1-score for Douban.

(b) Results of NDCG for Douban.

Fig. 6. Sparsity Analysis for Social Behavior Prediction for Douban.

Table 8
The information of 6 groups.

Consumption number	Epinions	Douban
0–5	918(9.5%)	784(12.3%)
5–10	2,106(21.7%)	1,437(22.5%)
10–20	2,708(27.9%)	1,701(26.5%)
20–40	2,265(23.3%)	1,533(24.0%)
40–80	1,105(11.4%)	754(11.7%)
>80	610(6.2%)	186(3.0%)

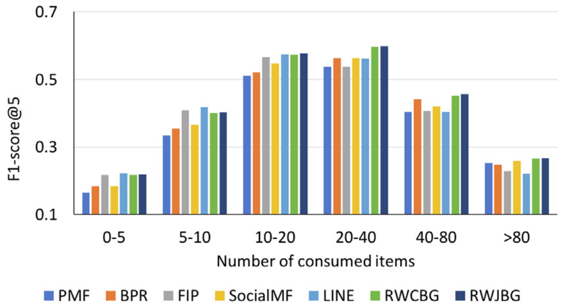
6.2.2. Consumption behavior prediction

In this part, we show the performance of consumption behavior prediction. The results for Epinions are shown in Table 5, and results for Douban are shown in Table 6. RWCBG is the abbreviation of *Random Walk on Consumption Behavior Graph*.

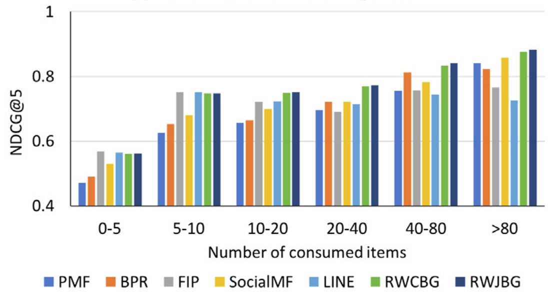
For consumption behavior prediction, RWCBG outperforms other methods in almost all of our tests, proving the effectiveness of utilizing both two kinds of behavior information and capturing higher-order relationships to alleviate the sparsity problem. For CF methods, SocialMF performs better than others, no doubt because SocialMF considers the social trust propagation from neighbors. This result indicates that social information has a positive effect on users' preference prediction. BPR is a competitive algorithm considering the pairwise ranking loss, which is optimized by increasing the gap between positive and negative samples. FIP models these two kinds of user behaviors simultaneously. However, due to our datasets' lack of the user profile information (e.g., users' self-crafted registration files) and item attributes (e.g., a textual description of a service item), FIP does not perform very well in our tests. Moreover, we argue that another shortcoming of FIP comes from the shared user latent representations, which need to fit two kinds of behaviors at the same time, making the performance far from satisfactory.

6.3. Sparsity analysis (RQ2)

Because each user only connects to limited social users and interacts with several items, the problem of sparsity is unavoidable in behavior prediction tasks. To verify the ability to mitigate sparsity, a common experiment is to divide users into different groups according to the sparsity and analyze the performance of the different groups [45,46].



(a) Results of F1-score for Epinions.

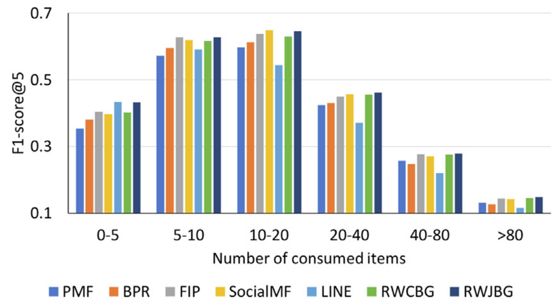


(b) Results of NDCG for Epinions.

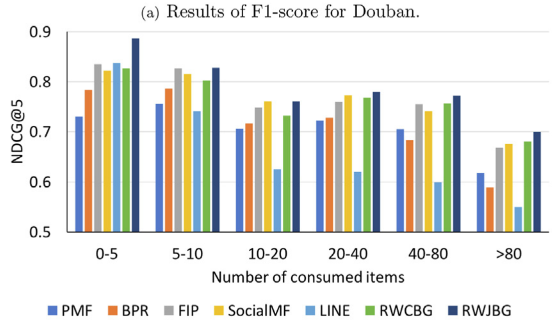
Fig. 7. Sparsity Analysis for Consumption Behavior Prediction for Epinions.

6.3.1. *Sparsity analysis for social behavior prediction*

Generally speaking, the sparser the data, the more difficult it is to fit the data and the worse the overall results. Since the number of times each user appears is different, we need to know how our method works for them. The fewer times users appear, the fewer links they have and the more sparse they are.



(a) Results of F1-score for Douban.

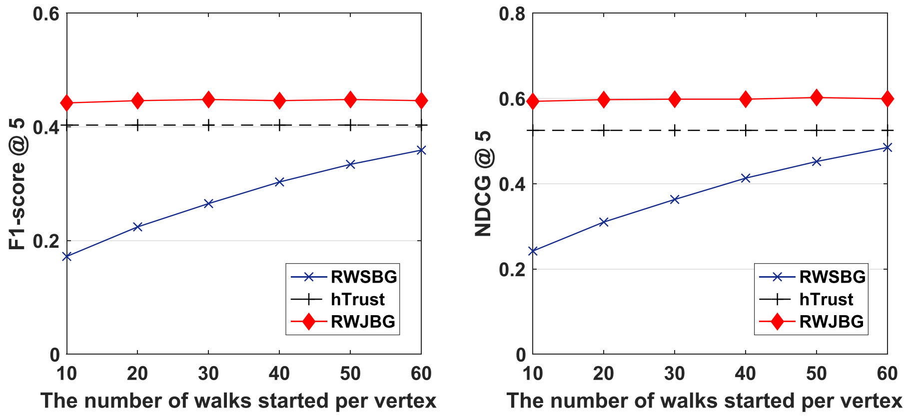


(b) Results of NDCG for Douban.

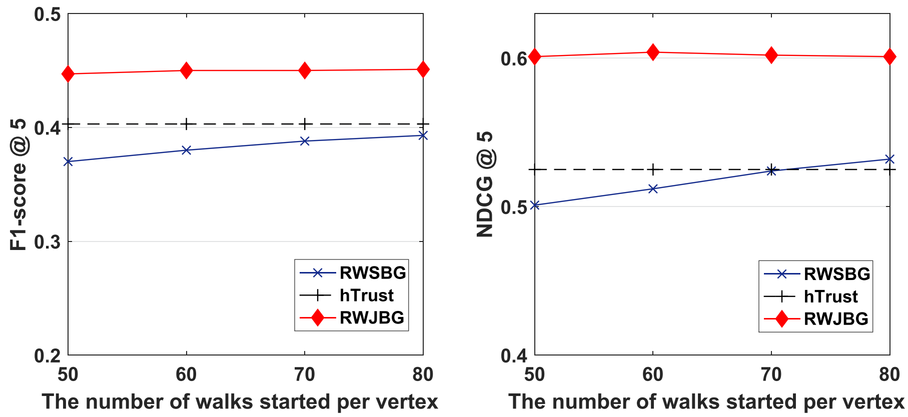
Fig. 8. Sparsity Analysis for Consumption Behavior Prediction for Douban.

Table 9
Three examples of recommendation results of our method.

Typical Examples for Epinions Dataset
Our method recommends $user_{26}$ and $item_{125}$ for $user_{76}$ while other methods ignore them. $user_{26}$ appeared 12 times, and only 14 users bought $item_{125}$.
Explanation 1: Trust relations can be transmitted. (a) There are 12 users who trust $user_{26}$ in Epinions, so we define the set as Γ_{26} and $\|\Gamma_{26}\| = 12$. (b) The set of users that $user_{76}$ has followed is γ_{76}. (c) We find that $\gamma_{76} \cap \Gamma_{26} = 7$.
Conclusion 1: We believe $user_{76}$ will trust $user_{26}$ because of connectivity hidden links. Then, as a follower, $user_{76}$ has a high probability to trust someone their followee has trusted.
Explanation 2: Consumption behavior enhances the social behavior prediction. $user_{76}$ and followees (who also trust $user_{26}$) have bought some of the same items. $user_{76}$: $item_{47}, item_{61}, item_{122}, item_{165}, item_{103}, item_{200}, item_{220}$. $user_{62}$: $item_{47}, item_{61}$. $user_{234}$: $item_{103}, item_{122}$. $user_{282}$: $item_{122}, item_{165}, item_{200}, item_{220}$.
Conclusion 2: $user_{76}$ will trust $user_{26}$ also because $user_{76}$ and their followee have bought some of the same items. Starting from these items, we can also capture the relations between $user_{76}$ and $user_{26}$.
Explanation 3: Social relationships can affect consumption behavior. (a) There are five users who have bought $item_{125}$. We define the set of these users as Ψ_{125}, and $\|\Psi_{125}\| = 5$. (b) The set of users that $user_{76}$ has followed is γ_{76}. (c) We find that $\|\gamma_{76} \cap \Psi_{125}\| = 4$.
Conclusion 3: We believe $user_{76}$ will buy $item_{125}$ because of their social friends.



(a) Result when λ_{γ} changes from 10 to 60.



(b) Result when λ_t changes from 50 to 80.

Fig. 9. Impact of different values of λ_{γ} and λ_t on the performance of social behavior prediction for Epinions.

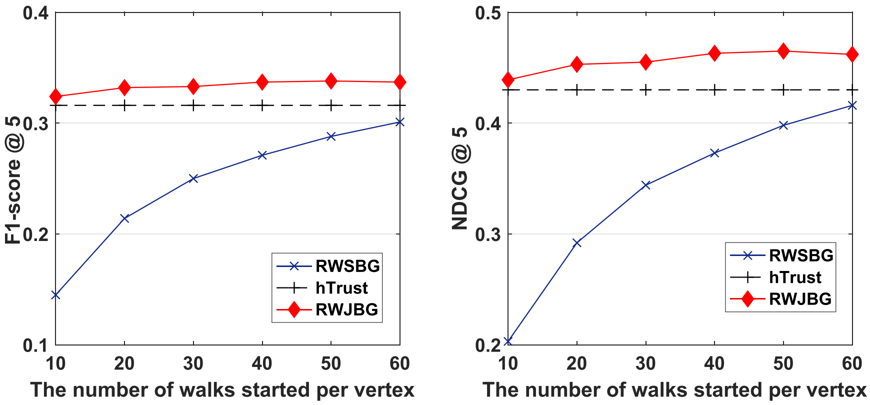
We divide the users into six groups on each dataset according to their social links; the details of each group are shown in Table 7. We can see that the majority of users appear less than ten times (with 52.6% for Epinions, 52.1% for Douban), which means these users have less than ten social links with other users. Then, for each group of users, we observe the performance on the test set; the results are shown in Figs. 5 and 6. Here, we choose the trade-off metric F1-score ($k = 5$) and NDCG ($k = 5$).

We can see that if a user appears less than five times, AA is almost nonfunctional. This happens because when the social trust matrix **S** is sparse, nearly all the adjacent vectors of each user in the **S** are zero. The same problem exists for all methods. However, a random walk can extend the relations for each user and item, which gives our proposed method better performance than the other methods. Our test results show the effectiveness of our method in mitigating the sparsity problem. If, however, a user appears many times, then many users trust or are trusted by that user. Cumulatively, these users have enough social links and are not limited by sparsity anymore. For these active users, accurate predictions can be captured by many methods. Under such circumstances, our approach may underperform others.

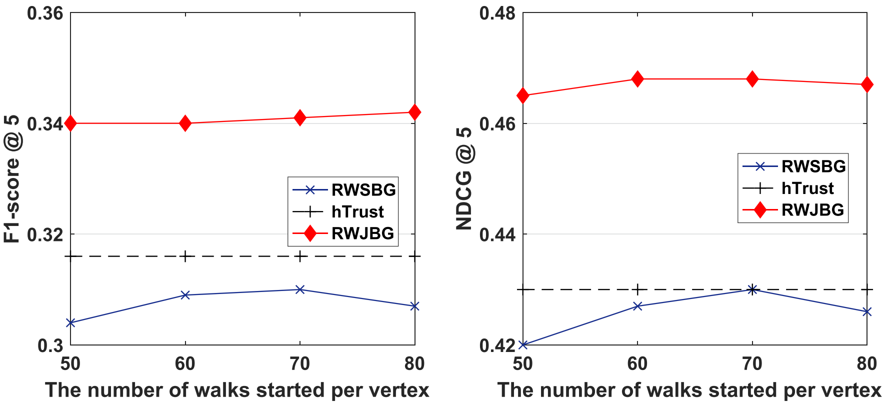
6.3.2. Sparsity analysis for consumption behavior prediction

One aspect affecting consumption behavior prediction is that users consume different numbers of items. Naturally, users who consume more items are called active users. Thus, we analyze the differences between users who buy different numbers of items. As before, we divided the users into six groups according to the number of items they have purchased in the two datasets. The information about all the groups is presented in Table 8.

Fig. 7 and Fig. 8 show the results for users with varying consumption records in the two datasets. We can see that in these tests, our method outperforms PMF and BPR, which only utilize user-item observed links, and even SocialMF. These results show the effectiveness of utilizing random walks on the Joint Behavior Graph to extract higher-order links. However, com-



(a) Result when λ_γ changes from 10 to 60.



(b) Result when λ_t changes from 50 to 80.

Fig. 10. Impact of different values of λ_γ and λ_t on the performance of social behavior prediction for Douban.

pared to other methods that also utilize the social network for recommendation (i.e., FIP, LINE), it seems that our method sometimes does not perform very well when users have bought only a few items. A possible reason is that random walks rely on the observed user-item relationships for generating random walk sequences. As shown in Table 2, the consumption behavior is much sparser than social behavior. Given such data characteristics, each user is more likely to be linked to a social link than a consumption link. Therefore, there are less consumption-based link paths than social-based link paths, leading to the proposed model focusing more on the social link performance. Therefore, the performance improvement in consumption behavior prediction is not as significant as in social behavior prediction.

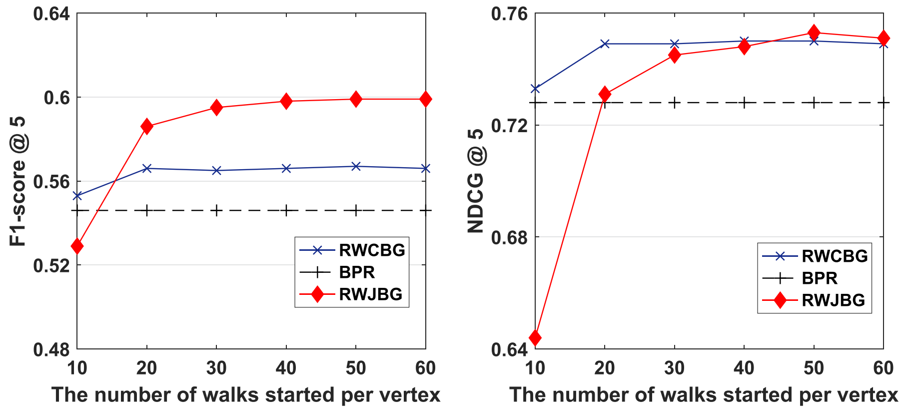
Moreover, two facts should be noted. (1) The number of items in the datasets is far more than the number of users, including users who bought more than 40 items, so many models still suffer from the cold-start problem when determining preferences. In terms of the overall effect, our method becomes better when users have bought more than 20 items. (2) Moreover, we use two metrics (F1-score and NDCG) to evaluate the performance. Compared to the comprehensive F1-score, our proposed method has better performance on the NDCG (the quality of candidates' ranking), which shows that meaningful high-order neighbors are uncovered by random walks.

The experimental results show that our proposed model can help to mitigate the data sparsity issue to some extent. Due to the unsupervised nature of a random walk, it can not best serve consumption behavior prediction when users have very few records. We would like to explore how to solve this phenomenon, and we propose it as future work.

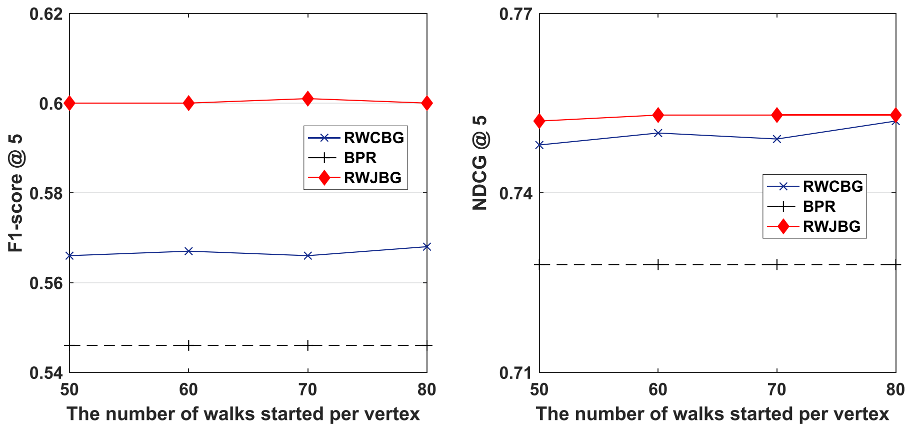
6.3.3. Case study

In order to further verify the effectiveness of our method and its ability to mitigate the sparsity problem, we show a case study result for Epinions with three discoveries. For these results, we randomly selected $user_{76}$ to observe the prediction list.

In the list, we found two relatively sparse nodes $user_{26}$ (14 records) and $item_{125}$ (12 records). Other methods ignored these two nodes, but by searching for the relations, we get the interesting results shown in Table 9.



(a) Result when λ_γ changes from 10 to 60.



(b) Result when λ_t changes from 50 to 80.

Fig. 11. Impact of different values of λ_γ and λ_t on the performance of consumption behavior prediction for Epinions.

6.4. Parameter analysis (RQ3)

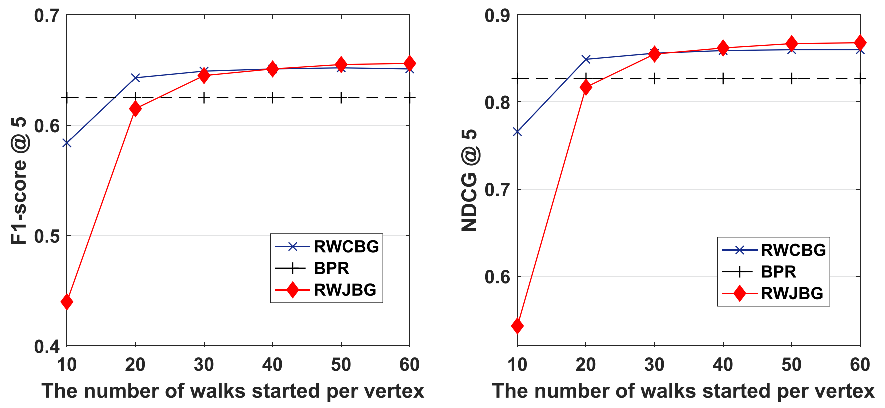
Many parameters affect the prediction performance: (a) The number of latent dimensions λ_d. (b) The number of paths λ_γ for each node, which controls the breadth-first traversal. (c) The walk length λ_t, which controls the depth-first traversal. (d) The window size λ_w, which defines the scope of the association with the current node.

To optimize our parameters, we started by setting the window size and the number of dimensions to $\lambda_w = 5$ and $\lambda_d = 64$. We focus on adjusting λ_γ and λ_t because they determine the characteristics of the random walks. We vary them in the order of λ_γ followed by λ_t, which means we will adjust λ_t assuming the best setting for λ_γ.

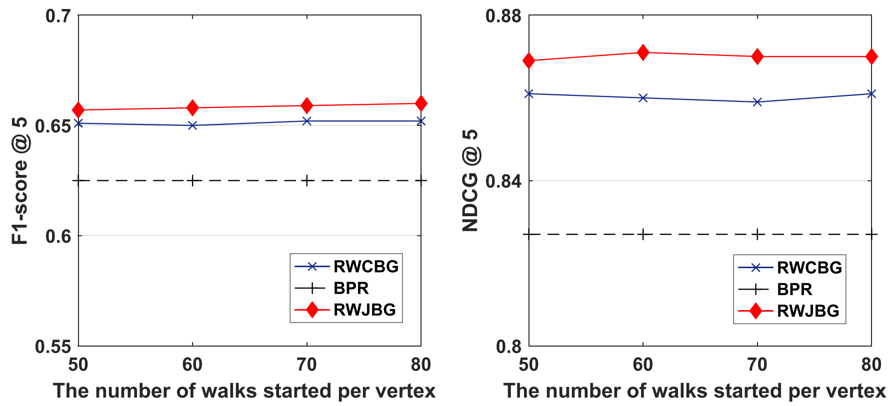
The default values are $\lambda_\gamma = 10$ and $\lambda_t = 40$. If we assign a small value to them, then the random walks connot identify the higher-order relationships (unobserved links) well, but overly large values will not increase performance (perhaps even leading to overfitting). Considering these factors, we varied λ_γ in {10,20,30,40,50,60} with λ_t chosen from {50,60,70,80}.

Figs. 9 and 10 show the results of RWJBG and RWSBG with the change of parameters for the two datasets. We start to increase λ_γ from 10. When RWSBG and RWJBG achieve the best performance, we fix λ_γ and begin to increase λ_t. Since the scale of rating data is more than that of trust data, the size of the user-item matrix is larger than the user-user matrix. We can see that RWJBG outperforms RWSBG, which indicates that a large network (CBG) is able to provide more information than a small one (SBG).

For a social behavior network, if the values of λ_γ and λ_t are small, RWSBG is not able to adequately capture the information of a network. With increases of λ_γ, the performance of RWSBG increases significantly while the RWJBG method fluctuates or even remains unchanged, which reveals that when adjusting parameters to get more information, the smaller graph allows



(a) Result when λ_γ changes from 10 to 60.



(b) Result when λ_t changes from 50 to 80.

Fig. 12. Impact of different values of λ_γ and λ_t on the performance of consumption behavior prediction for Douban.

Table 10
Performance with large parameters.

(a) Performance of social behavior prediction with large parameters.

	Epinions ($\lambda_\gamma = 90, \lambda_t = 100$)		Douban ($\lambda_\gamma = 110, \lambda_t = 100$)	
	Metrics			
Methods	F1-score	NDCG	F1-score	NDCG
RWSBG	0.428	0.581	0.330	0.458
RWJBG	0.451	0.605	0.342	0.471

(b) Performance of consumption behavior prediction with large parameters.

	Epinions ($\lambda_\gamma = 90, \lambda_t = 100$)		Douban ($\lambda_\gamma = 110, \lambda_t = 100$)	
	Metrics			
Methods	F1-score	NDCG	F1-score	NDCG
RWCBG	0.564	0.748	0.653	0.862
RWJBG	0.600	0.756	0.664	0.875

RWSBG to gain more useful information to enhance the performance. However, RWJBG always outperforms RWSBG, which indicates that the CBG enhances the performance of social behavior prediction on JBG.

Figs. 11 and 12 show the performance of consumption behavior prediction with the change of parameters. At first, RWCBG is not able to capture enough information from the CBG when $\lambda_\gamma = 10$ and $\lambda_t = 40$. Then, it quickly achieves the best performance and starts to fluctuate. RWJBG underperforms the RWCBG in the beginning stage, as we set a small value and try to capture the relationships in a larger graph containing the SBG and CBG. When we increase λ_t and λ_γ, the performance of RWJBG slightly outperforms RWCBG at the end because of the contained social information.

To fully verify the impact of these two parameters, we also set up larger parameters on two datasets to observe the results. The results summarized in Table 10(a) and (b) give us two conclusions: (1) Although RWSBG performs a little bit better than before, it still underperforms RWJBG. (2) Due to the acquisition of information being close to saturation, RWJBG and RWCBG have no significant performance improvements.

7. Conclusion

In this work, we first combine graphs of two kinds of user behavior to get the Joint Behavior Graph. The combination of different behavior graphs achieves the first step of information integration, allowing both users and items to be deemed nodes. Then, we utilize the random walk mechanism to extend the relations for each node by generating many meaningful and semantic sequences. Next, we follow the idea of distributed representation learning technology to obtain continuous latent representations of all nodes. Specifically, we adopt the SkipGram to learn the semantic sequences and capture the first-order and higher-order connectivity information among nodes. Finally, the learned representations can be utilized to predict the users' future behaviors.

Several problems remain and suggest future research. First, we roughly simplified the rating information; in other words, our experimental setup does not take the weight of links among nodes into consideration. In addition, the generated random walk sequences are very important, and the method of generating them will affect the final results. However, our proposed method is completely unsupervised. In future research, we will study how to generate balanced random walks to further improve the method. In the future, we hope to improve the prediction performance of SNSs by solving these problems.

CRediT authorship contribution statement

Junwei Li: Investigation, Software, Validation, Formal analysis, Writing - original draft, Writing - review & editing. **Le Wu:** Conceptualization, Investigation, Validation, Formal analysis, Writing - review & editing. **Richang Hong:** Funding acquisition, Validation, Writing - review & editing. **Jinkui Hou:** Validation, Writing - review & editing, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by grants from the National Natural Science Foundation of China(Grant No. 61972125, U1936219, 61722204, 61932009, 91846201), and the Fundamental Research Funds for the Central Universities (Grant No. JZ2020HGPA0114).

References

[1] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in Artificial Intelligence 2009 (2009) 4.
[2] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, Advances in Neural Information Processing Systems (2008) 1257–1264.
[3] M. Jiang, P. Cui, F. Wang, W. Zhu, S. Yang, Scalable recommendation with social contextual information, IEEE Transactions on Knowledge and Data Engineering 26 (11) (2014) 2789–2802.
[4] H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, in: Proceedings of the 4th ACM International Conference on Web Search and Data Mining, ACM, 2011, pp. 287–296.
[5] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in: Proceedings of the 4th ACM Conference on Recommender Systems, ACM, 2010, pp. 135–142.
[6] A. Anagnostopoulos, R. Kumar, M. Mahdian, Influence and correlation in social networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 7–15.
[7] J. Tang, H. Gao, X. Hu, H. Liu, Exploiting homophily effect for trust prediction, in: Proceedings of the 6th ACM International Conference on Web Search and Data Mining, ACM, 2013, pp. 53–62.
[8] X. Qian, H. Feng, G. Zhao, T. Mei, Personalized recommendation combining user interest and social circle, IEEE Transactions on Knowledge and Data Engineering 26 (7) (2014) 1763–1777.
[9] Z. Zhao, H. Lu, D. Cai, X. He, Y. Zhuang, User preference learning for online social recommendation, IEEE Transactions on Knowledge and Data Engineering 28 (9) (2016) 2522–2534.
[10] L. Wu, Y. Ge, Q. Liu, E. Chen, R. Hong, J. Du, M. Wang, Modeling the evolution of users' preferences and social links in social networking services, IEEE Transactions on Knowledge and Data Engineering 29 (6) (2017) 1240–1253.
[11] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, H. Zha, Like like alike: joint friendship and interest propagation in social networks, in: Proceedings of the 20th International Conference on World Wide Web, ACM, 2011, pp. 537–546.
[12] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781..

[13] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, Journal of the Association for Information Science and Technology 58 (7) (2007) 1019–1031.
[14] G. Jeh, J. Widom, Simrank: a measure of structural-context similarity, in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2002, pp. 538–543.
[15] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.
[16] A. Grover, J. Leskovec, node2vec Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 855–864.
[17] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: Large-scale information network embedding, Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.
[18] A.K. Menon, C. Elkan, Link prediction via matrix factorization, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 437–452.
[19] K. Miller, M.I. Jordan, T.L. Griffiths, Nonparametric latent feature models for link prediction, in: Advances in Neural Information Processing Systems, 2009, pp. 1276–1284.
[20] P. Massa, P. Avesani, Trust-aware recommender systems, in: Proceedings of the 2007 ACM Conference on Recommender Systems, ACM, 2007, pp. 17–24.
[21] Z.-D. Zhao, M.-S. Shang, User-based collaborative-filtering recommendation algorithms on hadoop, IEEE, 2010, pp. 478–481.
[22] H. Ma, I. King, M.R. Lyu, Learning to recommend with social trust ensemble, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2009, pp. 203–210.
[23] L. Wu, Y. Yang, K. Zhang, R. Hong, Y. Fu, M. Wang, Joint item recommendation and attribute inference: An adaptive graph convolutional network approach, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 679–688.
[24] L. Wu, Y. Ge, Q. Liu, E. Chen, B. Long, Z. Huang, Modeling users' preferences and social links in social networking services: a joint-evolving perspective, in: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016, pp. 279–286.
[25] P. Wu, Y. Tu, X. Yuan, A. Jatowt, Z. Yang, Neural framework for joint evolution modeling of user feedback and social links in dynamic social networks, in: Proceedings of the 27th International Joint Conference of Artificial Intelligence, 2018, pp. 1632–1638.
[26] J. Li, L. Wu, R. Hong, K. Zhang, Y. Ge, Y. Li, A joint neural model for user behavior prediction on social networking platforms, ACM Transactions on Intelligent Systems and Technology (TIST) 11 (6) (2020) 1–25.
[27] B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, et al, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, ACM, 2001, pp. 285–295.
[28] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, in: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2009, pp. 452–461.
[29] Q. Liu, E. Chen, H. Xiong, C.H. Ding, J. Chen, Enhancing collaborative filtering by user interest expansion via personalized ranking, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (1) (2011) 218–233.
[30] L. Wu, Q. Liu, E. Chen, N.J. Yuan, G. Guo, X. Xie, Relevance meets coverage: A unified framework to generate diversified recommendations, ACM Transactions on Intelligent Systems and Technology (TIST) 7 (3) (2016) 1–30.
[31] J. Leskovec, D. Huttenlocher, J. Kleinberg, Predicting positive and negative links in online social networks, in: Proceedings of the 19th International Conference on World Wide Web, ACM, 2010, pp. 641–650.
[32] M. Al Hasan, M.J. Zaki, A survey of link prediction in social networks, in: Social Network Data Analytics, Springer, 2011, pp. 243–275.
[33] T. Zhao, J. McAuley, I. King, Leveraging social connections to improve personalized ranking for collaborative filtering, in: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, ACM, 2014, pp. 261–270.
[34] P. Sun, L. Wu, M. Wang, Attentive recurrent social recommendation, in: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, 2018, pp. 185–194.
[35] L. Wu, P. Sun, R. Hong, Y. Ge, M. Wang, Collaborative neural social recommendation, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2018) 1–13.
[36] L. Wu, L. Chen, R. Hong, Y. Fu, X. Xie, M. Wang, A hierarchical attention model for social contextual image recommendation, IEEE Transactions on Knowledge and Data Engineering (2019) 1854–1867.
[37] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, M. Wang, A neural influence diffusion model for social recommendation, in: 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 235–244.
[38] N.Z. Gong, A. Talwalkar, L. Mackey, L. Huang, E.C.R. Shin, E. Stefanov, E. Shi, D. Song, Joint link prediction and attribute inference using a social-attribute network, ACM Transactions on Intelligent Systems and Technology (TIST) 5 (2) (2014) 1–20.
[39] Z. Yin, M. Gupta, T. Weninger, J. Han, Linkrec: a unified framework for link recommendation with user attributes and graph structure, in: Proceedings of the 19th International Conference on World Wide Web, ACM, 2010, pp. 1211–1212.
[40] Z. Yin, M. Gupta, T. Weninger, J. Han, A unified framework for link recommendation using random walks, in: Advances in Social Networks Analysis and Mining, IEEE, 2010, pp. 152–159.
[41] L. Bottou, Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010), Large-scale machine learning with stochastic gradient descent, Springer, 2010, pp. 177–186.
[42] C. Goutte, E. Gaussier, A probabilistic interpretation of precision, recall and f-score, with implication for evaluation, in: European Conference on Information Retrieval, Springer, 2005, pp. 345–359.
[43] X. He, T. Chen, M.-Y. Kan, X. Chen, Trirank Review-aware explainable recommendation by modeling aspects, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 1661–1670.
[44] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 2001, pp. 556–562.
[45] S. Sedhain, S. Sanner, D. Braziunas, L. Xie, J. Christensen, Social collaborative filtering for cold-start recommendations, in: Proceedings of the 8th ACM Conference on Recommender Systems, ACM, 2014, pp. 345–348.
[46] G. Guo, J. Zhang, N. Yorke-Smith, Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, 2015, pp. 123–129.