SamWalker++: recommendation with informative sampling strategy

Can Wang, Jiawei Chen, Sheng Zhou, Qihao Shi, Yan Feng and Chun Chen

Abstract—Recommendation from *implicit feedback* is a highly challenging task due to the lack of reliable negative feedback data. Existing methods address this challenge by treating all the un-observed data as negative (dislike) but downweight the confidence of these data. However, this treatment causes two problems: (1) Confidence weights of the unobserved data are usually assigned manually, which lack flexibility and may create empirical bias on evaluating user's preference. (2) To handle massive volume of the unobserved feedback data, most of the existing methods rely on stochastic inference and data sampling strategies. However, since a user is only aware of a very small fraction of items in a large dataset, it is difficult for existing samplers to select *informative* training instances in which the user really dislikes the item rather than does not know it.

To address the above two problems, we propose two novel recommendation methods SamWalker and SamWalker++ that support both adaptive confidence assignment and efficient model learning. SamWalker models data confidence with a social network-aware function, which can adaptively specify different weights to different data according to users' *social contexts*. However, the social network information may not be available in many recommender systems, which hinders application of SamWalker. Thus, we further propose SamWalker++, which does not require any side information and models data confidence with a constructed pseudo-social network. In the pseudo-social network, similar users are connected with specific item nodes or community nodes. This way, the inference of one's data confidence can benefit from the knowledge from other similar users. We also develop fast random-walk-based sampling strategies for our SamWalker and SamWalker++ to adaptively draw informative training instances, which can speed up gradient estimation and reduce sampling variance. Extensive experiments on five real-world datasets demonstrate the superiority of the proposed SamWalker and SamWalker++.

Index Terms—Recommendation, Implicit feedback, Sampling, Exposure

1 Introduction

With the exponential growth of information on electronic commerce websites, Collaborative Filtering (CF) as a prevalent approach in recommender systems are drawing more and more attention from both academia and industry [1], [2], [3]. There are two types of feedback data in Collaborative Filtering systems. The first is called *explicit feedback*, where the numerical ratings directly reflecting users' preference are provided. The other is *implicit feedback*, which is a natural byproduct of users' behavior such as consumption, viewing or clicking. Since implicit feedback are more easily available, recent research attention is increasingly shifted from explicit feedback to implicit feedback. However, learning a recommender system from implicit feedback is more challenging due to the lack of reliable negative data. Only the positive feedback are observed, while the negative feedback are mixed with missing values in unobserved data. In other words, items interacted by the user reflect that the user favors the items, while non-interacted items does not necessarily mean the user dislikes the items. In most cases, users may just not know the items that they have not interacted.

A conventional strategy to address the problem is treating all the un-observed data as negative (dislike) but downweight the confidence of these data. However, this treatment poses two key research problems for implicit recommendation:

(P1) How to assign appropriate confidence weights for data? Data confidence weights, which controls the contribution of the data on learning a recommendation model, usually significantly affect the model's accuracy. However, assigning appropriate confidence weights is challenging, as the real data confidence may change for various user-item combinations. Some unobserved data can be attributed to user's preference while others are the results of users' limited scopes. Most of existing methods rely on manual assignment of confidence weights to the data. Choosing confidence weights usually require human rich experience or large computational resource for grid search. Furthermore, it is unrealistic for researchers to manually set flexible and diverse weights for millions of data. Coarse-grained manual confidence weights will create empirical bias on estimating user's preference.

(P2) How to efficiently learn a recommendation model from the large-scale implicit feedback data? The large-scale unobserved data incur inefficiency problem. It is computationally impractical to traverse over the whole data set to obtain the gradients. To address this problem, two types of strategies have been adopted in previous works. The first is batch-based gradient descent with memorization, such as ALS [4], eAlS [5], FAWMF [6]. However, this kind of methods are only suitable for the specific models with K-separable property and L2 loss function, which will sacrifice models capacity and lead to sub-optimal performance. In fact, the models with more flexible deep structure,

• *Can Wang, Jiawei Chen, Sheng Zhou, Qihao Shi, Yan Feng and Chun Chen are with the School of Computer Science and Technology, Zhejiang University, HangZhou, China*
Jiawei Chen is corresponding author
E-mail: {wcan,sleepyhunt,zhousheng_zju,shiqihao321,fengyan,chenc}@zju.edu.cn

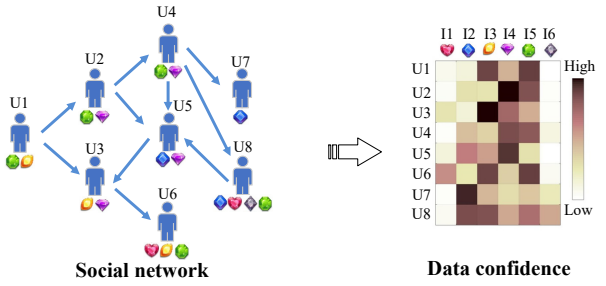


Fig. 1. SamWalker estimates data confidence based on user's social relations. The left part of the figure illustrates a social network including implicit feedback expressed by users. The consumed items (i.e. the items with positive feedback) are shown below the users. The right part shows our inferred data confidence.

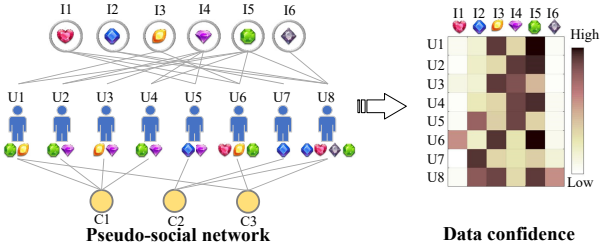


Fig. 2. SamWalker++ estimates data confidence based on the constructed pseudo-social network. The left part of the figure illustrates a pseudo-social network, where users are connected with additional item nodes ($I1$-$I6$) or community nodes($C1$-$C3$). We give links for the user-item pairs with positive feedback, and the user-community pairs if the user belongs to the community.

confidence weights and loss function have been validated achieving better performance [7], [8]. The other type is employing stochastic gradient descent solvers with data sampling strategies. However, in real-world applications, users typically are only aware of a relatively small fraction of the potential items [9]. In such cases, existing samplers usually select uninformative data with low confidence weights, in which the user just does not know the item rather than dislikes it. This will affect convergence and recommendation performance of the model.

To deal with these problems, we propose two novel recommendation methods SamWalker and SamWalker++ to simultaneously learn the personalized data confidence and draw informative training instances. We first present SamWalker which leverages social network information to address the problems. With the development of online social websites, social relations have become a major information resource when users select items to consume [10]. Users usually get item information from social friends [11] and their *exposure* to items (i.e. whether a user knows the items) will inevitable be dominated by their *social contexts* (i.e. whether their direct or indirect social neighbors have consumed the items). Thus, users' social relations and social contexts reflect how users are exposed to the items and suggest the confidence of the data. It is consistent with our intuitions. Note that there exists two reasons for negative feedback: unknown or dislike. The more popular an item is among the user's social neighbors (e.g. the purple gem $I4$ comparing with the green gem $I5$ for user $U1$ in Figure 1), the more likely it will be that the user knows the item and his feedback is attributed to his preference. Correspondingly, the data will be more reliable in deriving user's preference. To capture this insight, as illustrated in Figure 1, SamWalker simulates item information propagation along the social network and models individual confidence weights as a social context-aware function. By iteratively learning transformation function and user's preference based on EXMF framework (exposure-based matrix factorization [12]), SamWalker can adaptively specify different weights to different data based on user's social contexts.

A key limitation of SamWalker is that it requires the presence of social network information, which may not be available in many applications. To deal with this problem, we further propose SamWalker++, which only uses implicit feedback data and does not require any side information. The rationale of SamWalker++ is the "wisdom of the crowds", i.e. a user's behavior reflects not only his exposure but also the knowledge of other similar users. SamWalker++ constructs a pseudo-social network to replace social network, where users are connected with specific additional nodes, so that the knowledge of one's exposure can be transferred to other similar users. As shown in Figure 1, here we explicitly introduce two kinds of nodes to capture two kinds of similarities. On the one hand, note that a positive feedback signifies that the user knows the item. The users who have interacted with common items may have similar exposure. Thus, SamWalker++ leverages items as bridges so that the inference of the user's exposure can benefit from the rich information from the similar users with co-purchased items. On the other hand, recent social literatures [13], [14] suggest that users are clustered into some content-sharing communities. Item information will be spread in the community and the community members tend to share similar exposure. Motivated by this point, SamWalker++ deduces the latent communities for users and leverages communities as medium to exploit the knowledge of other community members. With the pseudo-social network, SamWalker++ devises a novel exposure model on the network and specifies data confidence weights with a network-aware function, which naturally encodes rich correlation information between users into the data confidence and potentially boosts recommendation performance.

Let us use an example to illustrate why constructing pseudo-social network can help us infer the data confidence. Figure 2 illustrates an example of pseudo-social network. The inference of the target user's exposure can refer to the behavior of graph neighbors on the network. For example, we can deduce that user $U3$ may know the item $I5$ (green gem), because his similar users $U1$, $U3$, $U4$ who also belong to community $C1$ have interacted with $I5$. Correspondingly, his feedback data ($U3$-$I5$) can be attributed more to the preference and therefore has relatively large confidence weight. Another example can be seen from the higher confidence weights of the user-item pair ($U1$-$I4$) over the pair ($U1$-$I2$), as the similar users $U2$, $U4$ who share co-purchased items with $U1$ have consumed the item $I4$. Thus, the pseudo-social network is an effective and efficient tool to transfer the knowledge from the connected or even higher-order connected similar users to the target user, which improves the accuracy of the learned data confidence.

Due to the large number of unobserved data, develop-

ing an efficient informative sampling strategy is crucial. It is computationally infeasible to estimate and rank the current learned confidence weights for every data to select informative data. Instead, we propose efficient sampling strategies based on the random walk along the network for our SamWalker and SamWalker++. Intuitively, the more and closer neighbors have consumed the item, the more likely the user will know the item, in which case the feedback can be more confidently attributed to the user's preference. Consequently, we conduct personalized random walk for each user to explore his local (pseudo-) social network contexts and pick out items consumed by those similar users. Theoretical analysis proves that the distribution of the proposed sampling strategy is proportional to the data confidence while the sampling complexity is linear to the number of sampled instances, which heavily reduces the sampling variance and speed up gradient estimation.

Although this paper is extension of our previous work [15], in which we present a social recommendation model SamWalker that adaptively learns the data confidence based on users' social context. In this article, we further deliver the following contributions:

- As the social network information may not be available in many recommender systems, we propose to construct pseudo-social network to replace social network, where similar users are connected with specific item nodes and community nodes.
- We propose a novel recommendation model SamWalker++ on the pseudo-social network, which does not require any side information and can adaptively deduce the data confidence with the knowledge from other similar data.
- We develop an efficient random walk-based sampling strategy along the pseudo-social network to draw informative training instances for SamWalker++, which can both reduce sampling variance and speed up gradient estimation.
- Extensive experiments on five well-known benchmark datasets demonstrate that SamWalker++ outperforms a range of state-of-the-art methods and show the superiority of the proposed sampling strategy.

The rest of this paper is organized as follows. We briefly review related works in section 2. We give the problem definition and background in section 3. The SamWalker model is introduced in section 4. We further present our novel non-social method SamWalker++ in section 5. The informative sampling strategy and learning algorithm are presented in section 6. The experimental results are presented in section 7. Finally, we conclude the paper and present some directions for future work in section 8.

2 Related work

In this section, we review the most related works from the following four perspectives.

Data confidence in implicit recommendation. As the unobserved data are unreliable, learning a recommendation model from implicit feedback requires assigning confidence weights for the data. Most of the existing methods assign confidence weights manually. For example, the classic weighted matrix factorization (WMF) [4] and many neural-based collaborative methods (e.g. CDAE [16], NCF [7], LightGCN [17]) used a simple heuristic where all negative feedback data are equally downweighted vis-a-vis the positive feedback data; [5] and [18] assign the confidence weights based on item popularity.

More recently, a new probabilistic model EXMF [12] was proposed to incorporate user's exposure to items into the CF methods. When inferring user's preference, EXMF can translate user's exposure as data confidence. However, as analysed in section 3, this method suffers from overfitting and low efficiency. [6] further proposes to model data confidence with a community-based neural network in their FAWMF. However, FAWMF is designed for fast memorization-based learning so that the capacity of the network is constrained. It can be seen from the low rank of the confidence matrix. The confidence weights are modeled from a global perspective and usually lack personality. Our experiments show that FAWMF performs poorly on the sparse dataset.

Efficient Recommendation. For efficient recommendation, two strategies have been proposed in previous works. The first is employing stochastic gradient descent and sampling strategy to accelerate learning. The most popular sampling strategy is to draw un-observed feedback data uniformly, which is applied in many recommendation models, including classic matrix factorization [4], [19], pair-wised models (e.g. Bpr [20], NCR [21]) and sophisticated neural network-based methods (e.g. CDL [22], NCF [7], lightGCN [17]). However, uniform sampler will cause high variance and poor convergency.

Some other sampling strategies are proposed to improve convergence from different perspectives: [18] and [23] propose item popularity-based and item-user co-bias sampling strategy to reduce sampling variance; [24] presents several sampling strategies to balance backward computation of the item-dependent neural network and the user-item interaction function; [25] attempts draw positive data based on random walk along user-item bipartite graph. Our sampling strategies differs from [25] in that we pay more attention to sample informative negative data; [25], [26], [27], [28] present subtle dynamic sampling strategies to over-sample the "difficult" negative examples in which the prediction is much different from the ground-truth. Although effective, sampling "difficult" data for advanced preference model still suffers from low efficiency. Also, the "difficult" instances do not suggest that the user really dislike the item. The stochastic gradient estimator is biased and the natural noise in user-item feedback data may be amplified [29]; To capture real negative data, [30], [31] propose to leverage exposure data, which however is not available in many situations.

Another strategy for efficient recommendation is memorization strategy. When learning a recommendation model from implicit feedback, [4], [5], [6], [32], [33], [34] propose to memorize some important intermediate variables so that the massive repeated computation can be avoided. However, these strategies are just suitable for the model with K-separable property and L2 loss function. In fact, more

flexible loss functions(e.g. cross-entropy loss), confidence weights and neural network structure (e.g. NCF, LightGCN) have been validated achieving better performance.

Social recommendation. Social information has been utilized to improve recommendation performance in recent works. These methods mainly assume that connected users will share similar preference [35], [36], [37], [38], [39]: Sorec [40], TrustMF [41], PSLF [42], jointly factorize rating matrix and trust (social) matrix by sharing a common latent user space; In [43], [44], [45], [46], [47], [48], users' feedback is considered as synthetic results of their preference and social influence; [47], [49] utilize a social regularization term to constrain user's latent preference close to his trusted friends; [50], [51] extend pair-wise BPR framework by further assuming that for all items with negative feedback, a user would prefer the items consumed by their friends over the rest.

Also, there are two recent works claim that comparing with users' preference, users' exposure is more influenced by their social friends (neighbors). Thus, [52] and [8] integrate social influence on user's exposure into the generative process of EXMF model. However, these two methods need to infer large number of parameters of user's exposure, which will suffer from overfitting and inefficiency problems.

Random walk in recommendation. Random walk strategy has been widely applied in recommendation. [53] performs random walk along the social network to search relevant users who have similar preference with the target user for better rating prediction; [54] exploits random walk to obtain diverse recommendation; [55] further extends [54] in heterogenous information network to generate valuable meta-paths; [54] performs random walk to complete implicit feedback matrix for mitigating data sparsity problem; Similarly, [25] employs random walk to find more positive instances. We remark that these works adopt static (uniform or pre-defined) transfer probability in their random walk strategy. Besides, these random walks are not designed for sampling informative training instances.

3 PRELIMINARIES

In this section, we first give the problem definition of recommendation with implicit feedback. Then, we introduce exposure-based matrix factorization (EXMF) [12] framework from a variational perspective to provide usual insight into the relation between user's exposure and data confidence.

3.1 Problem definition

Suppose we have a recommender system with user set U (including n users) and item set I (including m items). The implicit feedback data is represented as $n \times m$ matrix X with entries x_{ui} denoting whether or not the user u has interacted with (e.g. consume[1]) the item i. Social information represented as $n \times n$ matrix T, with T_{ij} indicating connection between user i and j. Also, $\mathcal{T}_u$ denotes the set of the connected social friends (direct neighbors) of the user u. The task of a recommender system can be stated as follow: recommending items for each user that are most likely to be consumed by him.

1. Throughout the paper, we will use the term consume to denote any kind of implicit interaction, unless otherwise stated.

3.2 Exposure-based matrix factorization (EXMF)

EXMF [12] directly incorporates user's exposure into collaborative filtering. This is achieved by first generating the latent variable a_{ui}, which indicates whether user u has been exposed to item i. Then, EXMF models user's consumption x_{ui} based on a_{ui} as follow:

$$a_{ui} \sim Bernoulli(\eta_{ui}) \tag{1}$$

$$(x_{ui}|a_{ui}=1) \sim Bernoulli(\sigma(p_u^\top q_i)) \tag{2}$$

$$(x_{ui}|a_{ui}=0) \sim \delta_0 \approx Bernoulli(\varepsilon) \tag{3}$$

where δ_0 denotes $p(x_{ui}=0|a_{ui}=0)=1$; η_{ui} is the prior probability of exposure. Here we relax function δ_0 as $Bernoulli(\varepsilon)$ to make model more robust, where ε is a small constant (e.g. ε=0.001). When $a_{ui}=0$, we have $x_{ui} \approx 0$, since when the user does not know the item he can not consume it. When $a_{ui}=1$, when the user has learned the item, he will decide whether or not to consume the item based on his preference. x_{ui} can be generated with the classic preference model (e.g. matrix factorization)[2] and factorized by the latent vectors p_u and q_i, which respectively characterize latent preferences of the user u and latent attributes of the item i. To facilitate the description, here we collect the parameters of the preference model as $\theta = \{p_u, q_i\}_{u\in U, i\in I}$. Also, we remark that it would be straightforward to replace the matrix factorization with more sophisticated models such as factorisation machines [56] or neural networks [7], whenever needed.

3.3 Analyses of EXMF from variational perspective

The marginal likelihood of EXMF is composed of a sum over the marginal likelihood of individual datapoint $\log p(X) = \sum_{ui} \log p(x_{ui})$, which can be rewritten as:

$$\begin{aligned}\log p(x_{ui}) &= E_q[\log p(x_{ui}, a_{ui}) - \log q(a_{ui}|x_{ui})] \\ &\quad + E_q[\log p(a_{ui}|x_{ui}) - \log q(a_{ui}|x_{ui})] \\ &= L(\theta, q; x_{ui}) + KL(q(a_{ui}|x_{ui})||p(a_{ui}|x_{ui}))\end{aligned} \tag{4}$$

where $q(a_{ui}|x_{ui})$ is defined as an approximated variational posterior of a_{ui}.Since the second KL-divergence term is non-negative, optimizing marginal likelihood can be translated to optimize the evidence lower bound (ELBO) $L(\theta, q; x_{ui})$ w.r.t. both the variational posterior and the preference parameters θ. Classic variational methods [57] usually employ conjugate variational distribution and individual variational parameters[3], i.e. $q(a_{ui}|x_{ui}) = Bernoulli(\gamma_{ui})$. For convenience we collect variational parameters γ_{ui} as matrix Y. Then, the ELBO can be transformed into:

$$\begin{aligned}L(\theta, Y; X) &= \sum_{ui} E_q[\log p(x_{ui}, a_{ui}) - \log q(a_{ui}|x_{ui})] \\ &= \sum_{ui} \gamma_{ui} \ell(x_{ui}, \sigma(p_u^\top q_i)) + \sum_{ui} g(\gamma_{ui})\end{aligned} \tag{5}$$

The EBLO is composed of the two terms. The first term is a weighted Cross-Entropy loss for the predicted preference,

2. Here the preference model is slightly different from the original model presented in work [12] in that we employ Bernoulli likelihood instead of Gaussian likelihood on x_{ui}. In fact, Bernoulli likelihood is more natural for the binary variable [8].

3. Note that the EM algorithm presented in [12] is a special case of the classic variational inference.

where $\ell(a,b) = alog(b) + (1-a)log(1-b)$. The second term is a loss function w.r.t γ_{ui}:

$$g(\gamma_{ui}) = (1-\gamma_{ui})\ell(x_{ui},\varepsilon) + \ell(\gamma_{ui},\eta_{ui}) - \ell(\gamma_{ui},\gamma_{ui}) \quad (6)$$

Exposure probability as the data confidence. One observation from equation (5) is observed that the variational parameters γ_{ui}, which characterize the probability of the event that user i is exposed to item j, act as the confidence weights of the corresponding data to infer the preference parameters θ ($\theta = \{p_u, q_i\}_{u\in U, i\in I}$). This is clear by considering the following fact: when γ_{ui} becomes larger (or smaller), the inferred user and item factors p_u, q_i make more (or less) contributions on the objective function. This finding is consistent with our intuitions. Only if the user has been exposed to the item, can he decide whether or not to consume the items based on his preference. Thus, the data with larger exposure are more reliable in deriving user's preference.

Weaknesses. Although EXMF is capable of adaptively deriving the confidence of the data, it has two critical weaknesses: (1) Calculating gradients over all the unobserved data EXMF ($O(n \times m)$) is computational expensive and thus it practical use is limited. Although some sampling strategies can be used to speed up the algorithm, the gradient estimator exhibits high variance. Typically, in real world large datasets, each user will only be exposed to a relatively small fraction of the potential items that they could interact with. That is, the γ_{ui} of most data are small and they make limited contribution on updating parameters θ. Existing sampling strategy will usually draw uninformative data with small γ_{ui}, which deteriorates the convergence and even accuracy of the model. (2) EXMF assumes user-independent posteriors of user's exposure. On the one hand, the number of variational parameters γ_{ui} grows quickly with the number of users and items ($n \times m$). This will pose over-fitting and efficiency problems. On the other hand, Independent assumption of users' exposure is not practical in real world. Typically, users with directly social relations, co-purchased items or common communities will exhibit correlations in their exposure.

Thus, we are interested in, and propose a solution to two related problems:

1) A novel variational model of user's exposure that can both capture users' correlation and employ fewer variational parameters to speed up inference and alleviate overfitting.
2) A sampling strategy that can draw informative training instances to speed up gradient estimation and reduce sampling variance.

4 SAMWALKER

To solve the above problems, as illustrated in Figure 3, we first consider the correlations between socially connected users and propose a new social network-based recommendation method SamWalker, that replaces individual variational parameters with a social context-aware function: $\gamma_{ui} = g_\varphi(X,T)$. Specifically, we design a transformation function g_φ with parameters φ that map the local social context of the user, i.e. whether his direct or indirect social friends have consumed the item, into the probability of his exposure to the item. It is reasonable since users usually get item information from social network and their exposure to items depend on their local social contexts. An idea of modeling transformation function g_φ is to iteratively simulate the information spread via the social network. Similar to the PageRank algorithm [58], the label of user's exposure is initially set according to his consumption. Then, all users spread their item information to their connected friends via the social network as illustrated in Figure 3. The spread process is repeated until a global stable state is achieved. In each step, users collect information from the connected social friends (neighbors) and reconstruct their exposure as follows:

$$\gamma_{ui}^{(t+1)} = (1-c)x_{ui} + \sum_{k\in\mathcal{T}_u} c\varphi_{uk}\gamma_{ki}^{(t)} \quad (7)$$

The parameter c ($0 \le c \le 1$) specifies the relative contribution from the social friends and the initial label. φ_{uk} is defined as the edge weight, which balances the heterogenous effect from different graph neighbors ($k \in \mathcal{T}_u$) and meets $\sum_{k\in\mathcal{T}_u}\varphi_{ik} = 1$. Overall, SamWalker replaces γ_{ui} with a social context-aware function g parameterized by φ, to which the equation (7) converges:

$$Y = g_\varphi(X,T) \equiv \lim_{t\to\infty} Y^{(t)} = (I - cW)^{-1}(1-c)X \quad (8)$$

where we collect variables $\gamma_{ui}^{(t)}$ for every user-item pairs (u,i) as a matrix $Y^{(t)}$. Also, we collect φ_{uk} as a matrix W, in which $W_{uk} = \varphi_{uk}$ for connected user pairs and $W_{uk} = 0$ for others. As we can see from equation (8), SamWalker replaces the posterior expectation of user's exposure with a weighted combination of the users' consumption in his social network. The weight matrix $(I - cW)^{-1}$ is a graph or diffusion kernel [59], which has been widely adopted to measure node proximity in the network and depends on the edge weight parameters φ for every social ties. The inference of user's exposure can benefit from the knowledge of his similar social friends. Overall, SamWalker is capable of capturing social correlations between users and reduces the number of variational parameters from $O(n \times m)$ to $O(|E|)$, where $|E|$ denotes the number of edges in the social network. By iteratively learning the transformation function and the preference model, SamWalker can adaptively specify different weights to different data based on users' social contexts.

5 SAMWALKER++

A key limitation of SamWalker is that it requires additional social network data to model users' correlation, which may not be easy to collect in many recommender systems. To deal with this problem, we further propose SamWalker++, which does not use any side information. SamWalker++ models data confidence with a constructed pseudo-social network. As illustrated in Figure 2, in the pseudo-social network similar users are connected with specific additional nodes so that the learning of a user's exposure can benefit from the information of his similar users. We introduce two kinds of nodes:

- **Item nodes**: Note that the users who have interacted with common items tend to have similar exposure.

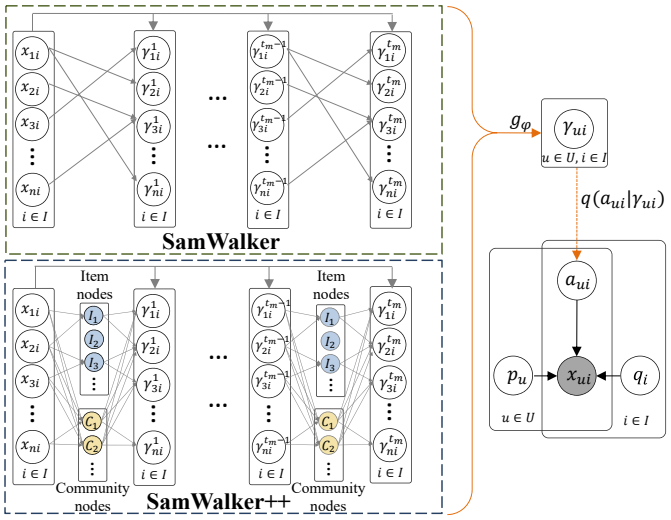


Fig. 3. A schematic view of the proposed SamWalker (Left-upper) and SamWalker++ (Left-bottem). SamWalker directly simulates the information spread via the social network, while SamWalker++ transfers the knowledge between users by introducing specific item and communities nodes as bridges.

Thus, we introduce item nodes as bridges and link the user-item pairs with positive feedback, so that the inference of one's exposure can benefit from the knowledge of his similar users with co-purchased items.

- **Community nodes:** Motivated by the social psychology statement [13], [14] that users are clustered into some content-sharing communities, we deduce the latent communities among users and introduce community nodes as medium to transfer the knowledge of users' exposure along the community. That is, as shown in Figure 2, we give links for the user-community pairs if the user belongs to the community. Note that pre-computing users' community with community discovering algorithms is not-optimal, as the rich supervised signals from users' exposure have not been exploited. Thus, we prefer end-to-end models. We first connect each user-item pairs and then adaptively learn users' community distribution as training process going on.

Given a pseudo-social network, we devise a novel exposure model on the network and specify data confidence weights with a network-aware function $\gamma_{ui} = h_\varphi(X, G)$, where G denotes the constructed pseudo-social network encoding similarity among users. That is, we design a transformation function h with parameters φ that maps the behaviors of the target user and his similar users into the probability of his exposure to the item. This way, the rich knowledge from these similar users can be transferred to learn the target user, which mitigates over-fitting problem and boosts inference accuracy. Similar to SamWalker, a promising way of modeling transformation function h_φ is to iteratively simulate the knowledge flowing along the pseudo-social network. We initially set the label of users' exposure with his consumption and then reconstruct exposure with the information from their connected users. We model two kinds of information propagation.

Along item nodes. On the one hand, the users with co-purchased items provide knowledge on the target user's exposure. We define the message from this kind of similar users to the target user u along the co-purchased items as:

$$m_{uj}^{u\leftarrow i\leftarrow u} = \sum_{i\in N_u\cap I}\sum_{v\in N_i} \varphi_{ui}^{u\leftarrow i}\varphi_{iv}^{i\leftarrow u}\gamma_{vj}^{(t)} \tag{9}$$

where N_u and N_i denote the neighbor nodes sets of the user u and item i; $\varphi_{ui}^{u\leftarrow i}, \varphi_{iv}^{i\leftarrow u}$ denote edge weights, balancing the contributions of information from different edges and meeting $\sum_{i\in N_u\cap I} \varphi_{ui}^{u\leftarrow i} = 1$, $\sum_{v\in N_i} \varphi_{iv}^{i\leftarrow u} = 1$. The product of $\varphi_{ui}^{u\leftarrow i}$ and $\varphi_{iv}^{i\leftarrow u}$ can be intepreted as path strength for $u \leftarrow i \leftarrow v$, characterizing the strength of information flowing from the user v to the target user u. As we can see, the users with more and stronger paths, suggesting that they exhibit more similarity with the target user, will bring more information on learning.

Along community nodes. On the other hand, users belonging to common communities will also exhibit correlations in their exposure. We define the message from this kind of similar users as:

$$m_{uj}^{u\leftarrow c\leftarrow u} = \sum_{c\in N_u\cap C}\sum_{v\in N_c} \varphi_{uc}^{u\leftarrow c}\varphi_{cv}^{c\leftarrow u}\gamma_{vj}^{(t)} \tag{10}$$

where $\varphi_{uc}^{u\leftarrow c}$ and $\varphi_{cv}^{c\leftarrow u}$ denote edge weights, balancing effect of different information edges and meeting $\sum_{c\in N_u\cap C} \varphi_{uc}^{u\leftarrow c} = 1$, $\sum_{v\in N_c} \varphi_{cv}^{c\leftarrow u} = 1$. Intuitively, $\varphi_{uc}^{u\leftarrow c}$ can be interpreted as user's community distribution and $\varphi_{cv}^{c\leftarrow u}$ as the extent to which the community member v exposes to the community c. The inference of user u's exposure can refer to the exposure of other community members. The knowledge flows along the path $u \leftarrow c \leftarrow v$ with strength $\varphi_{uc}^{u\leftarrow c}\varphi_{cv}^{c\leftarrow u}$.

Aggregation. We now aggregate the two messages to refine the target user u's exposure:

$$\gamma_{uj}^{(t+1)} = c(a_u m_{uj}^{u\leftarrow i\leftarrow u} + (1-a_u)m_{uj}^{u\leftarrow c\leftarrow u}) + (1-c)\gamma_{uj}^{(t)} \tag{11}$$

The parameter c $(0 \le c \le 1)$ specifies the relative contributions from the initial label and the connected[4] similar users. a_u balances the contributions of these two messages. Moreover, with the transitivity of similarity, our similar users of similar users, or even higher-order similar users, are potentially similar to us. These high-order similar users, as abundant information resources, usually provide useful knowledge of the target user's exposure. Thus, referring to SamWalker, we stack multi-stage of refinement as illustrated in Figure 3 so that the inference of a user's exposure can benefit from high-order connected users. Overall, SamWalker++ replace γ_{ui} with a network-aware function h parameterized by $\varphi = \{\varphi^{u\leftarrow i}, \varphi^{i\leftarrow u}, \varphi^{u\leftarrow c}, \varphi^{c\leftarrow u}, a\}$, to which equations (9),(10),(11) converge:

$$Y^{(t+1)} = h_\varphi(X, G) \equiv \lim_{t\to\infty} Y^{(t)} = (I - cW^+)^{-1}(1-c)X \tag{12}$$

where $W^+ = A\Phi^{u\leftarrow i}(\Phi^{i\leftarrow u})^T + (1-A)\Phi^{u\leftarrow c}(\Phi^{c\leftarrow u})^T$. Also, we collect parameters a_u for each user as a diagonal matrix A and collect $\varphi_{ui}^{u\leftarrow i}$ as a matrix $\Phi^{u\leftarrow i}$, in

4. Here we define the connected users as the users with common items or common communities.

which $\Phi_{ui}^{u\leftarrow i} = \varphi_{ui}^{u\leftarrow i}$ for connected user-item pair (u, i) and $\Phi_{ui}^{u\leftarrow i} = 0$ for others. Similar treatments are used for parameters $\varphi^{i\leftarrow u}, \varphi^{u\leftarrow c}, \varphi^{c\leftarrow u}$. SamWalker++ models user's exposure with a weighted combination of other users' consumption. Also, the weight matrix $(I - cW^{+})^{-1}$ is a graph or diffusion kernel [59] characterizing users proximity in the pseudo-network, which naturally encodes similarity or even high-order similarity between users into the inference procedures, which boosts the inference accuracy.

5.1 Discussion

We show that the proposed SamWalker++ satisfies the following four desirable properties:

Side-information free. As we can see, SamWalker++ only uses implicit feedback data and does not require any side information (e.g. social network, item contents, tags). As side information are not available in many recommender system, SamWalker++ can be applied in more situations comparing with the methods using side information(e.g. SamWalker).

Mitigate over-fitting. Another advantage of SamWalker++ is its ability to mitigate over-fitting problem. One evidence supporting this point can be seen from the less parameters of SamWalker++ comparing with EXMF. SamWalker++ reduces the number of variational parameters from $O(nm)$ to $O(|X^{(1)}| + nK)$, where $|X^{(1)}|$ denotes the number of observed positive feedback in the dataset and K denotes the number of inferred communities. Due to the sparsity of the implicit feedback data, the number of positive data ($|X^{(1)}|$) is much less than the the number of all data (nm).

How SamWalker++ mitigates over-fitting can be intepreted from another perspective. Referring to the analyses presented in [6], let us draw an analogy with the floating balls in the water, as illustrated in Figure 4. Learning exposure-based recommendation model according to equation (5) will give a force to pull up these positive balls (data) and push down these unobserved balls (data). For the vanilla EXMF model, the data confidence weights will easily achieve extreme values ($\gamma_{ui} \approx 1$ for the positive data and $\gamma_{ui} \approx 0$ for the unobserved data), where the unobserved data make little contribution to training the recommendation model and the model will suffer from over-fitting. But in SamWalker++, users' exposure (data confidence) are connected with additional items or communities, which can be analogies as additional balls with elastic links connecting the data. Naturally, the unobserved data with more and stronger connections with positive data, will be pulled up higher due to the force from the links. This way, when the model has well fitted the data, the positive and the unobserved ball(data) will get stable at different depth in water, as the knowledge (force) will prorogation among the data. The over-fitting effect will be mitigated.

Adaption. The data confidence is defined with a parameterized function instead of pre-defined values. Thus, the data confidence will adaptively evolve with training process going on, which is more flexible and does not require manual tuning of confidence weights. Moreover, as SamWalker++ integrates other user information, some irrelevant factors and even data noises may be injected into the inference. Fortunately, SamWalker++ is trained in a supervised manner by maximizing data likelihood so that the model can adaptively recognize important edges and extract useful information from the network.

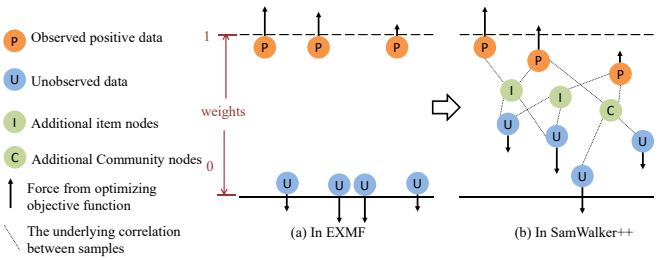


Fig. 4. Illustration of how SamWalker++ mitigates over-fitting

Fast informative sampling. Also, SamWalker++ supports fast informative sampling, which will be discussed in the next section.

6 Inference with Informative Sampler

6.1 Random walker-based sampler

Stochastic gradient descent (SGD), as a promising solution to speed up training procedures, has been widely applied in recommendation. The sampling strategy plays an important role in SGD, as it determines which data are used to update parameters and how often. However, since the informative instances with larger confidence γ_{ui} are usually buried in a large pile of uninformative ones, existing samplers usually fail to pick out informative data, leading to poor convergence and non-optimal performance. Thus, in this section, we develop a novel informative sampling strategy and address the following two key research questions: (Q1) Given the current learned data confidence γ_{ui}, how to define the informative sampling distribution? (Q2) Given informative sampling distribution, how to draw instances efficiently?

For the question (Q1), intuitively, the informative data with larger confidence γ_{ui} should be sampled with larger probability, since these terms make more contribution to the objective function. In fact, we have the following lemmas:

Lemma 1. *To evaluate the unbiased gradient of L w.r.t θ, the sampling strategy with distribution $p_{ui} \propto \gamma_{ui}$ can reduce sampling variance.*

The proof is presented in appendix.

Lemma 2. *To evaluate the unbiased gradient of L w.r.t θ, the sampling strategy with distribution $p_{ui} \propto \gamma_{ui}$ can speed up gradient calculation.*

Proof. If the sampling distribution is proportional to the data confidence ($p_{ui} = \gamma_{ui}/Z$, where $Z = \sum_{u\in U, i\in I} \gamma_{ui}$), we have:

$$\frac{\partial L}{\partial \theta} = \sum_{u\in U, i\in I} \gamma_{ui} \frac{\partial \ell_{ui}}{\partial \theta} = \sum_{u\in U, i\in I} \frac{p_{ui}}{Z} \frac{\partial \ell_{ui}}{\partial \theta} = E_p[\sum_{(a,b)\in p} \frac{1}{Z} \frac{\partial \ell_{ab}}{\partial \theta}] \tag{13}$$

where the confidence weights γ_{ui} have been absorbed into the sampling bias and does not need calculating in each iteration, which saves much time. □

The question (Q2) is more challenging, as the sampling distribution will evolve over a large data space as training process going on. A naive implementation of informative sampler is to estimate and rank the current learned confidence weights γ_{ui} for every user-item pairs and then pick out the informative data based on γ_{ui}. It is apparently inefficient and can not satisfy practical requirement. To avoid estimating confidence weights, we propose the following sampling strategy for our SamWalker and SamWalker++:

Random Walk-based sampling strategy. For the target user u, we perform the random walk along the network from user node u to sample the informative feedback data of user u. At each step t of random walk, supposing we are at a certain user v, we have two options:

(1) With probability c, we terminate the random walk. We stay at user v and randomly (uniformly) select a portion of (N_v/β) items that have been consumed by the user u, where N_v denotes the number of items consumed by the user v. Then we add the feedback data of user u on these selected items into sampled set S.

(2) With probability $(1-c)$, we continue our random walk. For the SamWalker model, we randomly select one of v's connected friends r based on personalized tie strength φ_{vr} and walk to r for the next walk step; For the SamWalker++ model, we first flip a coin based on a_v to decide which kinds of nodes (items or communities) we would like to walk along. If we choose item nodes (or community nodes) as a medium, then we randomly walk to one of v's connected items i (or communities c) based on edge weights $\varphi_{vi}^{u\leftarrow i}$ ($\varphi_{vc}^{u\leftarrow c}$). After that, we randomly walk from i (or c) to its connected user node r based on $\varphi_{ir}^{i\leftarrow u}$ (or $\varphi_{cr}^{c\leftarrow u}$) for the next walk step.

Our random walk-based sampling strategy satisfies the following desirable property:

Lemma 3. *The sampling probability of the above random walk-based strategy is proportional to the data confidence.*

Proof. It is easy to check that transformation probability from one user u to another user r in the step (2) is consistent with the (u,r)-th element of the matrix W for SamWalker or W^+ for SamWalker++. Further we can find that the (u,r)-th element of matrix $(cW^+)^t(1-c)$ (or $(cW)^t(1-c)$) is the probability of starting from the source user u and terminating at the user r in step t. Correspondingly, the (u,i)-th element of matrix $(c\Phi)^t(1-c)X$ represents the sampled probability of the user-item feedback data x_{ui} in the step t. Sum over the probability in different steps, we have the sampled probability of the data as follow:

$$P = \sum_{t=0}^{\infty} (c\Phi)^t (1-c)X/\beta \propto Y \tag{14}$$

which is proportional to the data confidence. □

Here we give more intuitive explanation of our proposed random walk-based sampling strategy. The random walk from the target user u will explore user's social contexts and finally randomly arrive at a specific user v based on their graph proximity. In fact, the network is constructed with social relations or users' similarity. Higher edge weights or shorter distance between the user v to the target user u, indicates more similarity between the two users and thus v will be selected with higher probability. The corresponding items, which are exposed (consumed) to u's similar users, are more likely exposed to the user u. Our random walk-based sampling strategy encodes the knowledge from other similar users and thus is capable of selecting informative instances.

In practice, we usually conduct α times random walk for each user to achieve more reliable mini-batch stochastic optimization. The parameters α, β control the batch size. Note that there is a chance for a single random walk to continue forever. In fact, we pay more attention to user's local social context and thus terminate the random walk when the number of steps exceeds a certain threshold ($t > t_m$). Concretely, when $t > t_m$, we uniformly walk to random user in the system and sample the data as option (1).

Algorithm 1 Inference of SamWalker and SamWalker++

```
Initialize parameters randomly;
while not converge do
    Sample a set of data S based on the random walk strategy as mentioned in section 6.
    Update parameters θ of the preference model based on the estimated gradient on the sampled data [equation (13)].
    Randomly select a portion of N_SI items.
    Update parameters φ based on backward propagation along the neural network for the selected items [equations (7,9,10,11)].
end while
```

6.2 Inference of the edge weights φ

Note that knowledge transfer between the data may also inject some irrelevant factors or even noises. Thus, we would like to train SamWalker and SamWalker++ in a supervised manner so that the model can adaptively recognize important edges and extract useful information along the network. We achieve this by optimizing the lower bound of margin likelihood (equation 5) w.r.t parameters φ with stochastic gradient methods. However, directly deriving gradient from transformation function g_φ (equation (8)) involves matrix inversion and suffers from low efficiency. Alteratively, as illustrated in Figure 3, we iteratively simulates information spread as equation (7), and stacks multi-layers neural network [60], [61] to infer the personalized edge weights. We also reparameterize $\varphi_{uv}, \varphi_{ui}^{u\leftarrow i}, \varphi_{iv}^{i\leftarrow u}, \varphi_{uc}^{u\leftarrow c}, \varphi_{cv}^{c\leftarrow u}$ with a Softmax transformation to deal with sum-to-one constraints. Backward prorogation can be easily conducted to infer tie strength φ, without requiring time-consuming matrix inversion. Further, mini-batch-based stochastic gradient methods can be employed to speed up the inference. Note that the data set S from random walk strategy is sampled for updating the recommendation model, and may not be suitable for φ. Thus, we choose the uniform sampler. In each step, we randomly (uniformly) select a portion of N_{SI} items and update tie strength based on users' exposure on these selected items. Overall, the inference of our SamWalker and SamWalker++ is presented in Algorithm 1.

6.3 Complexity Analyses

The time complexity of the inference of SamWalker and SamWalker++ is attributed to the following three parts: (1)

In sampling step, we will conduct α times random walk for each user to generate sampled data set S. The time for this part is $O(\alpha n t_m + |E| + |S|)$, where n denotes the number of users in the system and $|E|$ denotes the number of edges in the network; t_m denotes the max depth of random walk and $|S|$ denotes the number of data in the set S. (2) When inferring the preference parameters θ, we just estimate gradients on the sampled data S. The time for this step is $O(|S|d)$. (3) When inferring the parameters φ of the transformation function, we conduct the gradient back propagation along the network for the selected N_{SI} items. The complexity for this part is $O(N_{SI}|E|)$. Hence, the overall computational complexity is $O(\alpha n t_m + |S|d + N_{SI}|E|)$. Due to the sparsity of the recommendation data, users usually have limited social friends, interactions and communities. Note that $|E| = |T^+|$ for SamWalker and $|E| = nD + |X^+|$ for SamWalker++, where $|T^+|$ denotes the number of social relations and $|X^+|$ denotes the number of observed positive feedback. Also, similar to many recent works [16], [20] we usually let $|S|$ be five times as large as the number of observed data and let the number of selected items N_{SI} be 100. Thus, our algorithm is efficient on sparse implicit feedback data.

7 Experiments and Analyses

In this section, we conduct experiments to evaluate the performance of SamWalker and SamWalker++. Our experiments are intended to address the following questions:

(Q1) Do SamWalker and SamWalker++ outperform state-of-the-art recommendation methods?
(Q2) How does SamWalker++ compare with SamWalker?
(Q3) How does the proposed sampling strategy perform?
(Q4) Is it beneficial to introduce item nodes and community nodes to infer the data confidence?
(Q5) How does the parameter t_m (the max depth of prorogation) affect the recommendation performance?

7.1 Experimental protocol

Datasets. Five datasets Epinions[5], Ciao[6], LastFM[7], Moivelens-1M[8] and BookCrossings[9] are used in our experiments. These datasets contain users' feedback on the items. The datasets Epinions, Ciao and LastFM also contain users' social relations. The dataset statistics are presented in Table 1. Similar to [5], [48], we preprocess the datasets so that all items have at least three interactions and "binarize" user's feedback into implicit feedback. That is, as long as there exists some user-item interactions (ratings or clicks), the corresponding implicit feedback is assigned a value of 1. Also, we drop out items that have been consumed by too many (larger than 100) or too few (smaller than 3) users to moderate the popularity biases [62] in estimation. Grid search and 5-fold cross validation are used to find the best parameters. In our SamWalker and SamWalker++, we set $\eta_{ij} = 0.5$, $\alpha = 100$, $\beta = 20$, $c = 0.9$, $t_m = 5$. All experiments are conducted on a server with 2 TiTanX GPUs, Intel E5-2620 CPUs and 256G RAM[10].

TABLE 1
Statistics of five datasets.

Datasets	Number of users	Number of items	Number of interactions	Sparsity of interactions	Number of social relations
LastFM	1,892	4,489	52,668	0.62%	25,434
Ciao	5,298	19,301	138,840	0.14%	106,640
Epinions	21,290	34,075	333,916	0.05%	414,549
Moivelens-1M	6,040	3,678	1,000,177	4.50%	None
BookCrossing	13,097	37,075	473,846	0.10%	None

TABLE 2
The characteristics of the compared methods.

Methods	Social?	Exposure -based?	Sampling?	Complexity
WMF(ALS)	\	\	\	$O((n+m)d^3)$
BPR	\	\	√	$O((n+m+\|S\|)d)$
EXMF	\	√	\	$O(nmd)$
FAWMF	\	√	\	$O(\|\mathbf{X}^+\|(K+D))$
LightGCN	\	\	√	$O((\|S\|+\|\mathbf{X}^+\|)d)$
SBPR	√	\	√	$O((n+m+\|S\|)d)$
SERec-Bo	√	√	\	$O(nmd)$
SoEXBMF	√	√	\	$O(nmd^2)$
SamWalker	√	√	√	$O(\alpha n t_m + \|S\|d + N_{SI}\|E\|)$
SamWalker++	\	√	√	$O(\alpha n t_m + \|S\|d + N_{SI}\|E\|)$

Compared methods. We compare SamWalker and SamWalker++ with following baseline methods. Table 2 also summarizes their characteristics.

- WMF(ALS) [4], [63]: The classic weighted matrix factorization model for implicit feedback data. The corresponding ALS-based [4] algorithm can reduce inference complexity.
- BPR [20]: The classic pair-wise method for recommendation, coupled with matrix factorization. For efficient recommendation, BPR employs uniform sampling strategy to draw the training instances.
- EXMF [12]: A probabilistic model that directly incorporates user's exposure to items into traditional matrix factorization. EXMF does not utilize social information and chooses an item-dependent prior of user's exposure.
- FAWMF [6]: A fast matrix factorization model with adaptive confidence weights and memorization-based learning algorithm.
- LightGCN [17]: State-of-the-art recommendation model with graph neural network on the user-item interaction graph.
- SBPR [51]: SBPR integrates social information into BPR by assuming that the items consumed by connected friends are ranked higher than those not.
- SERec-Bo [52]: A probabilistic model that extends the EXMF model with social influence on user's exposure. Note that in [52] the authors reported that the performance of SERec-Bo is consistently better than their other model SERec-Re. Thus, here we choose SERec-Bo as a comparison.
- SoEXBMF [8]: A probabilistic model that further extends the EXMF model with both social knowledge influence and social consumption influence on user's exposure.

5. http://www.trustlet.org/epinions
6. http://www.cse.msu.edu/~tangjili/trust
7. https://grouplens.org/datasets/hetrec-2011/
8. https://grouplens.org/datasets/movielens/
9. https://grouplens.org/datasets/book-crossing/

10. Source code will be available at github https://github.com/jiawei-chen/SamWalker

Evaluation Metrics. We adopt the following metrics to evaluate recommendation performance:

- Recall@K (Rec@K): This metric quantifies the fraction of consumed items that are in the top-K ranking list sorted by their estimated rankings. For each user i, we define $Rec(i)$ as the set of recommended items in top-K and $Con(i)$ as the set of consumed items in test data for user i. Then we have:

$$Recall@K = \frac{1}{|U|}\sum_{i\in U}\frac{|Rec(i)\cap Con(i)|}{|Con(i)|} \quad (15)$$

- Precision@K (Pre@K): This measures the fraction of the top-K items that are indeed consumed by the user:

$$Precision@K = \frac{1}{|U|}\sum_{i\in U}\frac{|Rec(i)\cap Con(i)|}{|Rec(i)|} \quad (16)$$

- Normalized Discounted Cumulative Gain (NDCG): This is widely used in information retrieval and it measures the quality of ranking through discounted importance based on positions. In recommendation, NDCG is computed as follow:

$$NDCG = \frac{1}{|U|}\sum_{i\in U}\frac{DCG_i}{IDCG_i} \quad (17)$$

where DCG_i is defined as follow and $IDCG_i$ is the ideal value of DCG_i coming from the best ranking.

$$DCG_i = \sum_{j\in Con(i)}\frac{1}{\log_2(rank_{ij}+1)} \quad (18)$$

where $rank_{ij}$ represents the rank of the item j in the recommended list of the user i.

- Mean Reciprocal Rank (MRR): Given the ranking lists, MRR is defined as follow:

$$MRR = \frac{1}{|U|}\sum_{i\in U}\sum_{j\in Con(i)}\frac{1}{rank_{ij}} \quad (19)$$

7.2 Performance comparison (Q1)(Q2)

Table 3 presents the performance of the compared methods in terms of four evaluation metrics. The mark '*' denotes the winner in that row, while the boldface font denotes the winner among the non-social recommendation methods (WMF(ALS), BPR, EXMF, FAWMF, LightGCN, SamWalker++). Overall, except the results in the dataset Movielens, SamWalker or SamWalker++ outperform all compared baselines on all datasets for all metrics. For the sake of clarity, the columns 'impv1' and 'impv2' also show the relative improvement achieved by SamWalker over the all baselines and SamWalker++ over the non-social baselines respectively. The improvements are significant. Especially on the large dataset Epinions, the improvements of SamWalker over all baselines is 24.43%, 46.26%, 3.45%, 16,90% in terms of Pre@5, Rec@5, NDCG, MRR respectively, while the improvements of SamWalker++ over the non-social baselines are 67.73%, 60.37%, 6.24%, 50.27%.

Effect of modeling user's exposure. In the real world, users usually have personalized social contexts and thus are exposed to diverse information. Correspondingly, different data will have different confidence for estimating user's preference. That is, some un-observed feedback are more likely attributed to user's preference while others are the results of users' awareness. The exposure-based methods, which is capable of adaptively learning fine-grained data confidence weights, usually achieve better performance than the methods with manually assigned confidence weights. It can be seen from the experimental results that the best results are always achieved by the exposure-based methods.

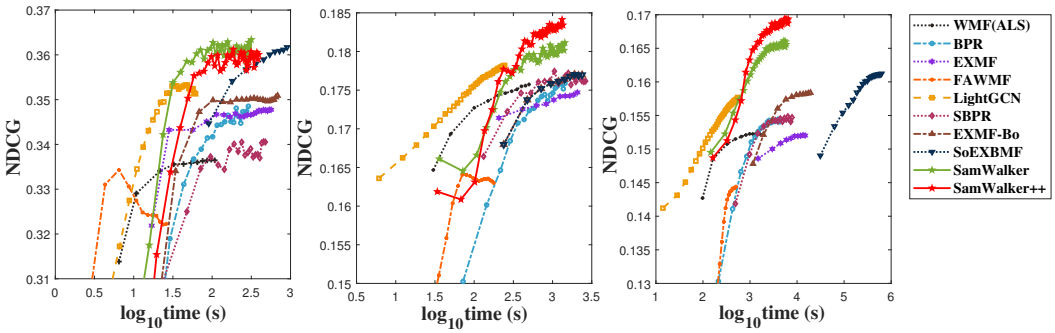
Comparing with exposure-based methods. Generally, our proposed SamWalker and SamWalker++ achieve better performance than existing exposure-based methods. The superiority can be attributed to two reasons: (1) The vanilla exposure-based method will easily suffer from over-fitting problem which deteriorates the recommendation accuracy. As discussed in subsection 5.1, Our SamWalker and SamWalker++ leverage (peusdo)-social network as bridges to transfer the information between the data, which mitigates the over-fitting effect. (2) For efficient recommendation, our methods employ random walk-based sampler to adaptively draw informative data, which can reduce sampling variance and achieve much better performance than heuristic samplers (e.g. Uniform). Thus, with minor sacrifices on accuracy, the stochastic learning strategy of our methods achieves comparable performance with the full-batch-based learning strategy, which either suffer from low efficiency (e.g. EXMF, SERec-Bo and SoEXBMF) or sacrifice models' flexibility with memorization mechanism (FAWMF). Specifically, FAWMF achieves excellent performance in the dense dataset Movielens, even slightly better than SamWalker++, while it has really poor performance in the sparse datasets such as BookCrossing and Epinions, which is even worse than the simple baselines.

SamWalker++ Vs. SamWalker. Although SamWalker++ does not use social information, SamWalker++ still achieves comparable performance with SamWalker. Especially in the dataset Epinions, where the social information is not as abundant as LastFM, SamWalker++ even outperforms SamWalker. These results validate the effectiveness of the constructed pseudo-social network, which encodes rich similarity information between users as the real social network. Moreover, SamWalker++ is more flexible, which can handle the situation where the social information is very sparse or absent.

Runtime vs. NDCG. Figure 5 depicts running time (X-axis) vs. NDCG (Y-axis) of the ten compared recommendation methods. As we can see, generally, SamWalker or SamWalker++ achieve best performance. The powerful competitor is LightGCN, which models user's preference based on light graph neural network. Although LightGCN has better NDCG than SamWalker (SamWalker++) at the beginning, SamWalker (SamWalker++) overtakes LightGCN soon with few iterations and achieves much better performance than LightGCN finally. Also, we observe that these exposure-based methods (EXMF,SERec,SoEXBMF) achieve good performance but are computational expensive. Although memorization strategy can speed up the training procedure, it limits the capacity of the model and deteriorates model's performance. It can be seen from the

TABLE 3
The performance metrics of the compared methods. The mark '*' denotes the winner in that row, while the boldface font denotes the winner among the non-social recommendation methods. The column 'Impv1' indicates the relative performance gain of our SamWalker compared to the best results among all baselines, while the column 'Impv2' indicates the relative performance gain of our SamWalker++ compared to the best results among the non-social methods.

Data-sets	Metrics	WMF	BPR	EXMF	FAWMF	Light-GCN	SBPR	SeRec	So-EXBMF	Sam-Walker	Impv1	Sam-Walker++	Impv2
LastFM	Pre@5	0.0928	0.1004	0.0957	0.1011	0.1051	0.0956	0.1018	0.1108	0.1177*	6.22%	**0.1099**	4.61%
	Rec@5	0.0841	0.0888	0.0859	0.0906	0.0934	0.0851	0.0907	0.1014	0.1072*	5.68%	**0.0983**	5.20%
	NDCG	0.3364	0.3485	0.3477	0.3242	0.3533	0.3405	0.3509	0.3617	0.3634*	0.48%	**0.3601**	1.93%
	MRR	0.3552	0.3704	0.3665	0.3604	0.3835	0.3556	0.3787	0.4140	0.4250*	2.67%	**0.4102**	6.96%
Ciao	Pre@5	0.0172	0.0144	0.0095	0.0143	0.0151	0.0156	0.0118	0.0181	0.0182*	0.25%	**0.0184**	6.98%
	Rec@5	0.0123	0.0125	0.0105	0.0092	0.0136	0.0124	0.0124	0.0152	0.0167*	10.38%	**0.0161**	18.25%
	NDCG	0.1757	0.1759	0.1747	0.1641	0.1800	0.1774	0.1770	0.1827	0.1811	-0.84%	**0.1834***	1.89%
	MRR	0.0738	0.0649	0.0485	0.0598	0.0675	0.0678	0.0532	0.0775	0.0775	0.02%	**0.0794***	7.57%
Epinions	Pre@5	0.0095	0.0087	0.0051	0.0090	0.0098	0.0088	0.0073	0.0119	0.0149	24.43%	**0.0165***	67.72%
	Rec@5	0.0087	0.0096	0.0082	0.0071	0.0116	0.0089	0.0101	0.0126	0.0184	46.26%	**0.0186***	60.37%
	NDCG	0.1522	0.1541	0.1513	0.1444	0.1593	0.1542	0.1577	0.1600	0.1656	3.45%	**0.1693***	6.24%
	MRR	0.0446	0.0406	0.0275	0.0396	0.0453	0.0411	0.0365	0.0537	0.0628	16.90%	**0.0681***	50.27%
Moive-lens-1M	Pre@5	0.3841	0.3613	0.3871	**0.4054***	0.3704	\	\	\	\	\	0.3929	-3.08%
	Rec@5	0.0924	0.0798	0.0936	**0.0949***	0.0845	\	\	\	\	\	0.0943	-0.63%
	NDCG	0.5971	0.5814	**0.5963***	0.5911	0.5733	\	\	\	\	\	0.5920	-0.86%
	MRR	1.5751	1.5023	1.5720	**1.6270***	1.5021	\	\	\	\	\	1.5810	-2.83%
Book-Crossing	Pre@5	0.0145	0.0109	0.0127	0.0091	0.0111	\	\	\	\	\	**0.0184***	26.40%
	Rec@5	0.0096	0.0096	0.0108	0.0058	0.0105	\	\	\	\	\	**0.0177***	64.09%
	NDCG	0.1683	0.1712	0.1705	0.1506	0.1705	\	\	\	\	\	**0.1829***	6.82%
	MRR	0.0658	0.0538	0.0582	0.0415	0.0541	\	\	\	\	\	**0.0811***	23.31%



(a) On dataset LastFM (b) On dataset Ciao (c) On dataset Epinions

Fig. 5. NDCG for each method in different steps versus time.

TABLE 4
Average empirical variance of the estimated gradients using different sampling strategies. Here $X^{(1)}$, $X^{(0)}$ denote the number of ones or zeros in the matrix X. Similarly, $r_i^{(1)}$, $r_i^{(0)}$ denote the number of ones or zeros in the i-th row of matrix X and $c_j^{(1)}$, $c_j^{(0)}$ denote the number of ones or zeros in the j-th column of matrix X.

Sampling strategy	Distribution	Average Variance					
		For SamWalker			For SamWalker++		
		50 It.	100 It.	500 It.	50 It.	100 It.	500 It.
S-allunion	$p_{ui} = 1/(n \times m)$	2.3793	2.5640	2.6663	2.6411	2.8059	2.8003
S-balunion	$p_{ui} = 1/(2\|X^{(x_{ui})}\|)$	0.4154	0.4941	0.5368	0.3691	0.4381	0.4639
S-itempop	$p_{ui} = I[x_{ui}=1]/(2\|X^{(1)}\|) + I[x_{ui}=0]c_i^{(1)}/(2\sum_{1\le b\le m} c_b^{(1)})$	0.1720	0.1938	0.2190	0.1382	0.1537	0.1583
S-cobias	$p_{ui} = r_u^{(1-x_{ui})} c_i^{(1-x_{ui})}/(2\sum_{1\le a\le n}\sum_{1\le b\le m} I[x_{ab}=x_{ui}] r_a^{(1-x_{ui})} c_b^{(1-x_{ui})})$	0.1617	0.1874	0.2031	0.1562	0.1762	0.1859
Random Walk	$p_{ui} \propto \gamma_{ui}$	0.1358	0.1464	0.1510	0.1296	0.1395	0.1394

efficiency but poor performance of FAWMF in these sparse datasets. Instead, our SamWalker and SamWalker++ employ network-based exposure model and sampler, which has both great efficiency and accuracy.

7.3 Sampler comparisons (Q3)

In this subsection, we compare our random-walk-based sampler with other sampling strategies including: (1) S-allunion, the global uniform sampling strategy; (2) S-balunion [23], [63], which samples un-observed data (zeros) and observed data (ones) with equal probability to deal with unbalance data problem; (3) S-itempop [18], which samples instances based on item popularity; (4) S-cobias [23], whose probability of sampling a un-observed data (zero) at location (u, i) is proportional to the number of ones in the u-th row (the number of user's consumption) and the number of zeros in the i-th colunm (item popularity). Also, an equivalent bias is introduced for the observed data. The detailed distributions of these sampling strategies are presented in Table 4.

Variance. We first empirically compare the variance of the estimated gradients of our SamWalker or SamWalker++ by using different sampling strategies. To do this, we train

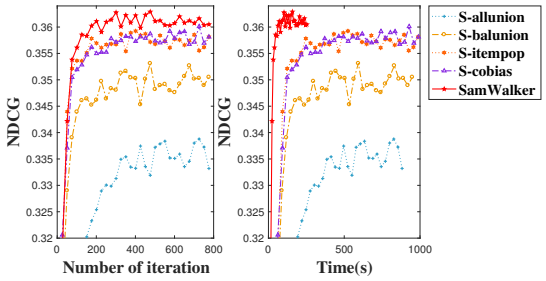


Fig. 6. NDCG for different sampling strategy versus the number of iterations and running time.

our models for 50, 100 or 500 epochs on the dataset LastFM. After training, we generate mini-batch with various sampling strategies and calculate un-biased estimated gradients of our objective w.r.t θ. We repeat this precedure for 1000 times and calculate the variance of the estimated gradients for different sampling strategies. The final results presented in Table 4 are averaged over various preference parameters. Table 4 shows that our random walk-based sampling strategy achieves the lowest average variance among various samplers for all conditions. This result is coincident with our Lemma 1. Our sampler, whose sampling probability is proportional to the data confidence, can indeed reduce the sampling variance. Also, we observe the following interesting phenomenon: With more training epoches, the variance will become larger, not smaller as usual. It may be explained as follow: SamWalker and SamWalker++ initialize with relative similar edge weights. With training proceeding, driven by the data, the edge weights and data confidence exhibit more and more heterogeneity. Thus, the variance of the estimated gradients will become larger.

Performance. Figure 6 presents the NDCG of SamWalker on the dataset LastFM with different sampling strategies versus the number of iterations and running time. As we can see, our adaptive random walk-based sampler performs better than others in all convergence, speed and accuracy. One reason is that our sampler has low variance, as showed in Table 4. Another reason is that in our sampler the confidence weights γ_{ui} are absorbed into the bias of the sampling distribution (referring to Lemma 2). Thus, the gradient estimator does not need to calculate γ_{ui} in each iteration to scale down the contribution from the different data, which saves much time.

7.4 Ablation study (Q4)

We further design a detailed ablation study to answer question Q4. That is, we remove different components at a time and compare CoSam with its two special cases: (1) SamWalker++noc, the special case of SamWalker++ which leaves out community nodes; (2) SamWalker++noi: the special case without item nodes. The performance of these special cases comparing with SamWalker++ is presented in Figure 7. We observe that SamWalker++ consistently outperforms its two special cases. This result validates the effectiveness of introducing both item nodes and community nodes to transfer the knowledge.

7.5 Effect of parameter t_m (Q5)

It will be interesting to explore how parameter t_m affects the performance of SamWalker++, where t_m indicates the max depth of the random walk. As we can see from Figure 8, the performance of SamWalker++ with $t_m \geq 2$ achieves significant improvement over the SamWalker++ with $t_m = 1$. This result validates the effectiveness of leveraging similar users to transfer the knowledge. The reason is that the inference from one's exposure can benefit from the information from other similar users. As t_m becomes larger, with few exception, the performance will improve first, since the knowledge from high-order similar users is also beneficial to learn the user's exposure. However, when t_m surpasses a threshold, the performance becomes unaffected or even experiences some degradation with further increase of t_m. Too deep random walk will bring more irrelevant factors and even data noises, which deteriorates recommendation accuracy.

8 Conclusion

Data confidence assignment and efficient model learning are two key problems in the implicit recommendation task. In this paper, we present two novel recommendation methods SamWalker and SamWalker++ to address these problems. SamWalker models the data confidence with a social context-aware function, which can reduce the number of learned parameters and adaptively specify personalized confidence weights for implicit feedback data. As the real social network data is not easily obtained, we instead construct pseudo-social network where similar users are connected with specific item nodes or community nodes. SamWalker++ models data confidence on the pseudo-social network, so that the inference of a user's exposure can benefit from other similar users. We further propose a random walker-based sampling strategy to draw informative training instances to speed up inference and reduce sampling variance. Extensive experimental results on five real-world datasets demonstrate the superiority of SamWalker and SamWalker++ over existing methods.

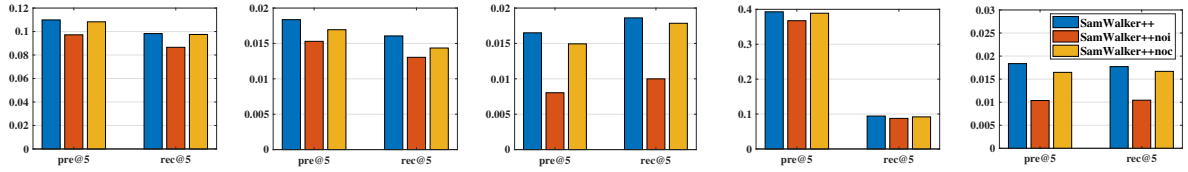
One interesting direction for future work is to leverage sophisticated graph neural network in the exposure model, which is capable of capturing more complex affinity between users and items along the interaction graph. Also, it will be interesting to explore dynamic exposure-based recommendation, as in the real world, users' preference, exposure and relations may evolve over time.

Acknowledgments

This work is supported by National Key R&D Program of China (Grant No: 2019YFB1600700, 2018AAA0101505) and National Natural Science Foundation of China (Grant No: U1866602).

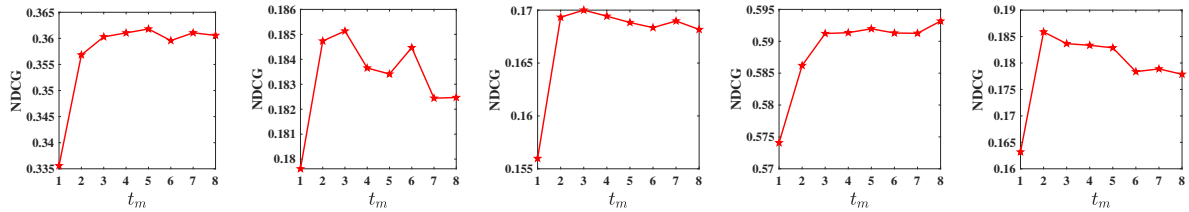
References

[1] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender systems handbook*. Springer, 2015.
[2] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender systems: an introduction*. Cambridge University Press, 2010.



(a) On LastFM (b) On Ciao (c) On Epinions (d) On Moivelens (e) On BookCrossing

Fig. 7. Performance comparison of SamWalker++ with its two special cases in terms of Pre@5 and Rec@5.



(a) On LastFM (b) On Ciao (c) On Epinions (d) On Moivelens (e) On BookCrossing

Fig. 8. Performance comparison with varying t_m.

[3] M. Karimi, D. Jannach, and M. Jugovac, "News recommender systems–survey and roads ahead," *Information Processing & Management*, 2018.

[4] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. Ieee, 2008, pp. 263–272.

[5] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 549–558.

[6] J. Chen, C. Wang, S. Zhou, Q. Shi, J. Chen, Y. Feng, and C. Chen, "Fast adaptively weighted matrix factorization for recommendation with implicit feedback." in *AAAI*, 2020, pp. 3470–3477.

[7] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*. ACM, 2017, pp. 173–182.

[8] C. Jiawei, F. Yan, E. Martin, Z. Sheng, C. Chun, and C. Wang, "Modeling users' exposure with social knowledge influence and consumption influence for recommendation," in *Proceedings of the 27th ACM International on Conference on Information and Knowledge Management*. ACM, 2018, pp. 953–962.

[9] M. Lichman and P. Smyth, "Prediction of sparse user-item consumption rates with zero-inflated poisson regression," in *The World Wide Web Conference*. IW3C2, 2018, pp. 719–728.

[10] X. Pan, L. Hou, and K. Liu, "Social influence on selection behaviour: Distinguishing local-and global-driven preferential attachment," *PloS one*, vol. 12, no. 4, p. e0175761, 2017.

[11] W. Chen, L. V. Lakshmanan, and C. Castillo, "Information and influence propagation in social networks," *Synthesis Lectures on Data Management*, vol. 5, no. 4, pp. 1–177, 2013.

[12] D. Liang, L. Charlin, J. McInerney, and D. M. Blei, "Modeling user exposure in recommendation," in *Proceedings of the 25th International Conference on World Wide Web*. ACM, 2016, pp. 951–961.

[13] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[14] T. Zhou, "Understanding online community user participation: a social influence perspective," *Internet research*, vol. 21, no. 1, pp. 67–81, 2011.

[15] J. Chen, C. Wang, S. Zhou, Q. Shi, Y. Feng, and C. Chen, "Samwalker: Social recommendation with informative sampling strategy," in *The World Wide Web Conference*. ACM, 2019, pp. 228–239.

[16] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 2016, pp. 153–162.

[17] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," *arXiv preprint arXiv:2002.02126*, 2020.

[18] H.-F. Yu, M. Bilenko, and C.-J. Lin, "Selection of negative samples for one-class matrix factorization," in *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 2017, pp. 363–371.

[19] C. C. Johnson, "Logistic matrix factorization for implicit feedback data," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[20] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 2009, pp. 452–461.

[21] B. Song, X. Yang, Y. Cao, and C. Xu, "Neural collaborative ranking," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 1353–1362.

[22] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2015, pp. 1235–1244.

[23] J. M. Hernández-Lobato, N. Houlsby, and Z. Ghahramani, "Stochastic inference for scalable probabilistic modeling of binary matrices," in *International Conference on Machine Learning*, 2014, pp. 379–387.

[24] T. Chen, Y. Sun, Y. Shi, and L. Hong, "On sampling strategies for neural network-based collaborative filtering," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 767–776.

[25] L. Yu, C. Zhang, S. Pei, G. Sun, and X. Zhang, "Walkranker: A unified pairwise ranking model with multiple relations for item recommendation," *AAAI*, 2018.

[26] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 273–282.

[27] W. Zhang, T. Chen, J. Wang, and Y. Yu, "Optimizing top-n collaborative filtering via dynamic negative item sampling," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2013, pp. 785–788.

[28] X. Wang, Y. Xu, X. He, Y. Cao, M. Wang, and T. Chua, "Reinforced negative sampling over knowledge graph for recommendation," in *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. ACM / IW3C2, 2020, pp. 99–109.

[29] D. Li, C. Chen, Q. Lv, H. Gu, T. Lu, L. Shang, N. Gu, and S. M. Chu, "Adaerror: An adaptive learning rate method for matrix approximation-based collaborative filtering," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International

World Wide Web Conferences Steering Committee, 2018, pp. 741–751.

[30] J. Ding, F. Feng, X. He, G. Yu, Y. Li, and D. Jin, "An improved sampler for bayesian personalized ranking by leveraging view data," in *Companion of the The Web Conference 2018 on The Web Conference 2018*. IW3C2, 2018, pp. 13–14.

[31] J. Ding, Y. Quan, X. He, Y. Li, and D. Jin, "Reinforced negative sampling for recommendation with exposure data," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 2230–2236.

[32] I. Bayer, X. He, B. Kanagal, and S. Rendle, "A generic coordinate descent framework for learning from implicit feedback," in *Proceedings of the 26th International Conference on World Wide Web*. IW3C2, 2017, pp. 1341–1350.

[33] C. Chen, M. Zhang, C. Wang, W. Ma, M. Li, Y. Liu, and S. Ma, "An efficient adaptive transfer neural network for social-aware recommendation," *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 225–234, 2019.

[34] C. Chen, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Efficient neural matrix factorization without sampling for recommendation," *ACM Transactions on Information Systems (TOIS)*, pp. 1–28, 2020.

[35] J. Golbeck, "Trust and nuanced profile similarity in online social networks," *ACM Transactions on the Web (TWEB)*, vol. 3, no. 4, p. 12, 2009.

[36] ——, "Generating predictive movie recommendations from trust in social networks," in *International Conference on Trust Management*. Springer, 2006, pp. 93–104.

[37] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1267–1275.

[38] W. Yao, J. He, G. Huang, and Y. Zhang, "Modeling dual role preferences for trust-aware recommendation," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014, pp. 975–978.

[39] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, "Friendship prediction and homophily in social media," *ACM Transactions on the Web (TWEB)*, vol. 6, no. 2, p. 9, 2012.

[40] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *CIKM*. ACM, 2008, pp. 931–940.

[41] B. Yang, Y. Lei, D. Liu, and J. Liu, "Social collaborative filtering by trust," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 2747–2753.

[42] Y. Shen and R. Jin, "Learning personal+ social latent factor model for social recommendation," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1303–1311.

[43] H. Ma, I. King, and 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.

[44] J. Tang, H. Gao, and H. Liu, "mtrust: discerning multi-faceted trust in a connected world," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 93–102.

[45] A. J. Chaney, D. M. Blei, and T. Eliassi-Rad, "A probabilistic model for using social networks in personalized item recommendation," in *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 2015, pp. 43–50.

[46] Y. Bao, H. Fang, and J. Zhang, "Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, 2014, p. 350.

[47] X. Wang, S. C. Hoi, M. Ester, J. Bu, and C. Chen, "Learning personalized preference of strong and weak ties for social recommendation," in *Proceedings of the 26th International Conference on World Wide Web*. IW3C2, 2017, pp. 1601–1610.

[48] L. Xiao, Z. Min, Z. Yongfeng, L. Yiqun, and M. Shaoping, "Learning and transferring social and item visibilities for personalized recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 337–346.

[49] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 135–142.

[50] X. Wang, W. Lu, M. Ester, C. Wang, and C. Chen, "Social recommendation with strong and weak ties," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 5–14.

[51] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 261–270.

[52] M. Wang, X. Zheng, Y. Yang, and K. Zhang, "Collaborative filtering with social exposure: A modular approach to social recommendation," in *AAAI, New Orleans, Louisiana, USA, February 2-7, 2018*, 2018.

[53] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 397–406.

[54] F. Christoffel, B. Paudel, C. Newell, and A. Bernstein, "Blockbusters and wallflowers: Accurate, diverse, and scalable recommendations with random walks," *Conference on Recommender Systems*, 2015.

[55] F. Vahedian, D. R. Burke, and B. Mobasher, "Weighted random walk sampling for multi-relational recommendation," *UMAP*, 2017.

[56] S. Rendle, "Factorization machines with libfm," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, pp. 1–22, 2012.

[57] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[58] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[59] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in neural information processing systems*, 2004, pp. 321–328.

[60] S. Zhang, L. Yao, and A. Sun, "Deep learning based recommender system: A survey and new perspectives," *arXiv preprint arXiv:1707.07435*, 2017.

[61] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[62] R. Cañamares and P. Castells, "Should i follow the crowd?: A probabilistic analysis of the effectiveness of popularity in recommender systems," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 415–424.

[63] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," *ICDM*, pp. 502–511, 2008.