SamWalker: Social Recommendation with Informative Sampling Strategy

ABSTRACT

Recommendation from implicit feedback is a highly challenging task due to the lack of reliable negative feedback data. Only positive feedback are observed and the unobserved feedback can be attributed to two reasons: *unknow* or *dislike*. 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 lacks flexible and may create empirical bias in 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 users are 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 a new recommendation method SamWalker that leverage social information to infer data confidence and guide the sampling process. By modeling data confidence with a social context-aware function, SamWalker can adaptively specify different weights to different data based on users' social contexts. Further, a personalized random-walk-based sampling strategy is developed to adaptively draw informative training instances, which can speed up gradient estimation and reduce sampling variance. Extensive experiments on three real-world datasets demonstrate the superiority of the proposed SamWalker method and its sampling strategy.

ACM Reference Format:

. 2019. SamWalker: Social Recommendation with Informative Sampling Strategy. In *Proceedings of The web conference (WWW'19)*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

With the exponential growth of information on electronic commerce websites such as Amazon and Taobao, recommender systems are drawing more attention from both academia and industry [5, 7, 15, 16, 25, 27, 40, 55]. Collaborative filtering (CF), as the prevalent recommendation model in these systems, infers user's preference and produces recommendations based on user's historical behaviors. Early work on CF mainly focused on explicit feedback, where the numerical ratings which directly reflect users' preference are provided.

WWW'19, 2019, San Francisco

ACM ISBN 123-4567-24-567/08/06...$15.00

https://doi.org/10.475/123_4

However, explicit feedback may not be available in many applications. We usually only have access to implicit feedback derived from user actions, e.g. users' video viewing and product purchasing history. Although implicit feedback is abundant in practice, recommendation from implicit feedback is more difficult than from explicit feedback. The reason is that learning from implicit feedback lacks reliable negative feedback data, i.e. only the positive feedback are observed in the implicit feedback data. The un-observed user-item feedback data (e.g. a user has not bought an item yet) are a mixture of real negative feedback (a user does not like it) and missing values (a user just does not know it). While the positive feedback suggests that the user likes the item, the un-observed feedback does not necessarily mean the user dislikes the item. In most cases, users may just not know the items that they have not consumed.

Existing methods address this problem by treating all the un-observed data as negative (dislike) but downweight the confidence of these data. Most of these methods rely on the careful assignment of confidence weights to the data. Although it may be effective, choosing these weights usually involves heuristic alterations to the data and thus need expensive exhaustive grid search via cross-validation. Furthermore, it is unrealistic for researchers to manually set flexible and diverse weights for millions of data. In practical scenarios, different data may have different confidence on estimating users' preference. Some unobserved feedback data may be attributed to users' preference, while others are the result of users' limited awareness. Coarse-grained manual confidence weights will create empirical bias on estimating user's preference.

Another problem in recommendation from implicit feedback is learning efficiency. Due to the lack of reliable negative feedback data, all the un-observed data need to be considered to learn users' preference [4]. However, this degrades the learning efficiency because of the large number of unobserved data. For efficient recommendation, most of the existing methods employ stochastic gradient descent solvers and corresponding data sampling strategies (e.g.uniform, item popularity-based). However, in real world, users typically are only aware of a relatively small fraction of the potential items [31]. In such cases, existing samplers usually select uninformative data with low confidence weight in which the user just does not know the item rather than dislikes it. This will affect the convergence and recommendation performance of the methods.

To deal with these problems, we propose a novel recommendation method SamWalker that leverages social information to simultaneously learn the personalized data confidence and draw informative training instances. With the development of online social websites, social relations have become a major information resource when users select items to consume [37]. Users usually get item information from social friends (neighbors) [10] and their *exposure* to items (i.e. whether a user knows the items) will inevitable be

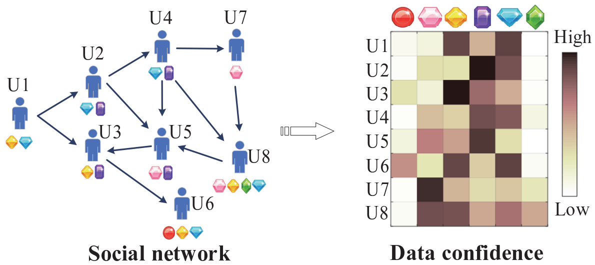


Figure 1: SamWalker estimates data confidence based on user's social relations. The left part of the figure illustrates a social network including implicit feedback expressed by users. The consumed items (i.e. the items with positive feedback) are shown below the users. The right part shows our inferred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

dominated by their *social contexts* (i.e. whether their direct or indirect social neighbors have consumed the items). Thus, users' social contexts can affect how users are exposed to the items and suggest the confidence of the data. It is consistently with our intuitions. Note that there exists two reasons for negative feedback: unknown or dislike. When an item is more popular among the user's social neighbors (e.g. the purple gem comparing with the green gem for user U1 in Figure 1), the user is more likely to know the item and his feedback is more 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 prorogation 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 [30]), SamWalker can adaptively specify different weights to different data based on user's social contexts.

Due to the large number of the unobserved data, developing an efficient informative sampling strategy is challenging. It is apparently inefficient to estimate and rank the current learned confidence weights for every data to select informative data. Instead, we propose an efficient sampling strategy based on the random walk along the social network. Intuitively, if our more and closer social friends (neighbors) have consumed the item, we are more likely to know it due to the information sharing in the social network. Our feedback on this item can thus be attributed to our preference with more confidence and the corresponding user-item feedback data is more informative. Consequently, we conduct personalized random walk for each user to explore his local social contexts and pick out items consumed by these close direct or indirect social neighbors. theoretical analysis proves that the distribution of the proposed sampling strategy is proportional to the data confidence, which can reduce the sampling variance and speed up gradient estimation.

It is worthwhile to highlight the following contributions:

- We propose a new recommendation method SamWalker that adaptively learns the data confidence based on users' social context.
- We propose an efficient social-based sampling strategy to draw informative training instances, which can both reduce sampling variance and speed up gradient estimation.
- Our experimental evaluation on three well-known benchmark datasets demonstrates that SamWalker outperforms a range of state-of-the-art methods and analyzes the superiority of the proposed social-based 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. The experimental results and discussions are presented in section 5. Finally, we conclude the paper and present some directions for future work in section 6.

2 RELATED WORK

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

Recommendation from implicit feedback data. Due to the lack of reliable negative feedback data, existing recommendation methods for implicit feedback data treat all the un-observed data as negative but downweight the confidence of these data. However, most of the existing methods manually assign coarse-grained confidence weights. For example, the classic weighted factorization matrix model (WMF) [22] and many neural-based collaborative methods (e.g. CDAE[48], NCF[17]) used a simple heuristic where all negative feedback data are equally downweighted vis-a-vis the positive feedback data. [18] and [53] also manually assign the confidence weights based on item popularity.

More recently, a new probabilistic model EXMF[30] was porposed 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, this method suffers from the efficiency problem. We will analyze EXMF model in details in section 3.

Some other pair-wise methods treat negative feedback data in a different way. Bayesian personalized ranking (BPR) [39] focuses on the learning of relative preferences and aims at maximizing the AUC objective function. They believe that user's preference on his consumed items is above un-consumed items. Similarly, the weighted approximated ranking pairwise (WARP) loss proposed in [47] optimizes Precision@K.

Efficient Recommendation. For efficient recommendation, most of the existing methods rely on the stochastic optimization and corresponding data sampling strategy. The most popular sampling strategy is to draw un-observed feedback data uniformly, which has been widely adopted by WMF [22], BPR [39],CDAE [48], NCF [17], etc. Also, [53] and [19] further propose item popular-based and item-user co-bias sampling strategy to reduce sampling variance. The detailed comparison of these sampling methods are presented in Table 4. However, as users are only exposed to very small fraction of the items, these samplers usually select uninformative data to estimate user's preference, which affects convergence and recommendation performance.

Some other sampling strategies are proposed to improve convergence and accuracy from different perspectives. [38, 54, 56] propose subtle dynamic sampling strategies to oversample the "difficult" negative examples in which the prediction is much different from the ground-truth. Although effective, sampling these "difficult" data for advanced preference model (e.g. BMF [19] or neural-based CF) still suffer from low efficiency [12]. Besides, the stochastic gradient estimator is biased and may amplify the natural noise in user-item feedback data [29]. [54] also proposes to draw positive data based on random walk along user-item bipartite graph. Our sampling strategy differs [54] in that we pay more attention to sample informative negative data. [9] proposes several sampling strategies to balance backward computation of the item-dependent neural network and the user-item interaction function. [12] proposes an effective sampling strategy by leveraging view data.

Also, [4, 18, 22] propose memoization strategies (e.g. ALS, eALS) to speed up inference. However, these strategies are just suitable for the model with K-separable property and L2 loss function (Guassian likelihood). In fact, cross-entropy loss function (Bernoulli likelihood) is more natural for the binary implicit feedback data and has been validated better performance than L2 [26].

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 [2, 13, 14, 23, 34, 51, 52, 59]: Sorec [33], TrustMF [50], PSLF [41], jointly factorize rating matrix and trust (social) matrix by sharing a common latent user space; In [3, 6, 32, 42, 45, 49], users' feedback is considered as synthetic results of their preference and social influence; [24, 45] utilize a social regularization term to constrain user's latent preference close to his trusted friends; [46, 57] 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 believe that comparing with users' preference, users' exposure is more influenced by their social friends (neighbors). Thus, [44] and [26] integrate social influence on user's exposure into the generative process of EXMF model. [44] extends EXMF with social regularization (SERec-Re) and social boosting (SERec-Bo). [26] integrates social knowledge influence and social consumption influence into EXMF model. However, these two methods need to infer $n \times m$ parameters of user's exposure, which will suffer from overfitting and become the efficiency bottleneck for practise application. SamWalker employs social-based variational posterior of user's exposure, which requires much fewer parameters and better captures heterogenous influence from both direct and indirect social neighbors. Further, SamWalker uses a social-based sampler to speed up inference and reduce sampling variance.

3 PRELIMINARIES

In this section, we first give the problem definition of implicit recommendation. Then, we introduce exposure-based matrix factorization (EXMF) [30] framework from variational perspective to provide usual insight about 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_{ij} denoting whether or not the user i has consumed the item j. Social information indicates the connections between users, represented as $n \times n$ matrix T. Also, $\mathcal{T}_i$ denotes the set of the connected social friends (direct neighbors) of user i. 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.

3.2 Exposure-based matrix factorization (EXMF)

EXMF [30] directly incorporates user's exposure into collaborative filtering. Firstly EXMF generates the latent variable a_{ij}, which indicates whether user i has been exposed to item j. Then, EXMF models user's consumption x_{ij} based on a_{ij},

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

$$x_{ij}|a_{ij} = 1 \sim Bernoulli(\sigma(u_i^\top v_j)) \tag{2}$$

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

where δ_0 denotes $p(x_{ij} = 0|a_{ij} = 0) = 1$; η_{ij} 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=1e-3). When $a_{ij} = 0$, we have $x_{ij} \approx 0$, since the user does not know the item and he can not consume it. When $a_{ij} = 1$, when the user has learned the item, he will decide whether or not to consume the item based on his preference. Thus, x_{ij} is modeled as the classic preference model[1] and factorized by the latent vectors u_i and v_j, which respectively characterize the latent preferences of user i and latent attributes of item j. To facilitate the description, here we collect the parameters of the preference model as $\theta = \{u_i, v_j\}_{i\in U, j\in I}$.

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 $\log p(X) = \sum_{ij} \log p(x_{ij})$, which can each be rewritten as:

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

The second term is the KL divergence of the approximate variational posterior from the true posterior. Since the KL-divergence is non-negative, the first term $L(\theta, q; x_{ij})$ is the evidence lower bound on (ELBO) the margin likelihood. Thus, optimizing marginal likelihood can be translated as optimizing the lower bound $L(\theta, q; x_{ij})$ w.r.t. both the variational posterior and the preference parameters θ. Classic variational methods [20] usually employ conjugate variational distribution and individual variational parameters[2], i.e. $q(a_{ij}|x_{ij}) = Bernoulli(\gamma_{ij})$. For convenient we collect variational parameters γ_{ij} for every user-item pairs (i, j) as matrix Y. Then, the ELBO can be transferred into:

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

[1]Here the preference model is slightly different from the original model presented in work [30] in that we employ Bernoulli likelihood instead of Gaussian likelihood on x_{ij}. In fact, Bernoulli likelihood is more natural for the binary variable [26].

[2]Note that the EM algorithm presented in [30] is a special case of the classic variational inference.

The EBLO is composed of the two terms, the first term is a weighted cross-Entropy loss for the predicted preference, where $\ell(a, b) = alog(b) + (1 - a)log(1 - b)$. The second term is a loss function w.r.t γ_{ij}:

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

Exposure probability as the data confidence. From equation (5), a good property is observed that the variational parameters γ_{ij}, which characterize the probability that user i is exposed to item j, act as the confidence of the corresponding data to infer the preference parameters θ ($\theta = \{u_i, v_j\}_{i \in U, j \in I}$). This is clear by considering the following fact: when γ_{ij} becomes larger (or smaller), the inferred user and item factors θ 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, he can decide whether or not to consume the items based on his preference. Thus, the data with larger exposure is more reliable in deriving user's preference.

Weaknesses. However, although EXMF can adaptively derive the confidence of the data, we emphasize two latent weaknesses of the EXMF model: (1) EXMF is costly in accounting for all unobserved data ($O(n \times m)$) and is thus unrealistic for practical use. 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 or be aware of a relatively small fraction of the potential items that they could interact with. That is, the γ_{ij} of most data are small and these data make limited contribution on updating parameter θ. Existing coarse-grained sampling strategy will usually draw uninformative data with small γ_{ij}, which affects the convergence and the recommended performance of the methods. (2) EXMF assumes user-independent posteriors of user's exposure. On the one hand, The number of variational parameters γ_{ij} grows quickly with the number of users and items ($n \times m$). This will suffer from serious overfitting and become the efficiency bottleneck for practise recommender systems. On the other hand, social network enable effective information sharing [1, 10]. Users' exposure to items will be influenced by their social contexts. Independent assumption of users' exposure is not practise in real world.

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

(1) A social-based variational distribution of user's exposure that can both model the social influence between users 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

For the purpose of solving the above problems, as illustrated in Figure 2, we propose a new recommendation method SamWalker that replaces individual variational parameter with a social context-aware function: $\gamma_{ij} = g_\varphi(X, T)$, where φ are replaced variational parameters. That is, we design an transformation function g_φ that maps the local social context of the user, i.e. whether his direct or indirect social friends (neighbors) have consumed the item, into the probability of user's 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 PageRank algorithm [35], we initially set the label of user's exposure according to his consumption. Then, all users spread their item information to their connected friends via the social network. The spread process is repeated until a global stable state is achieved. Concretely, in each step, users collect information from the connected social friends (neighbors) and reconstruct their exposure as follows:

$$\gamma_{ij}^{(t+1)} = (1 - c)x_{ij} + \sum_{k \in \mathcal{T}_i} c\varphi_{ik}\gamma_{kj}^{(t)} \tag{7}$$

The parameter c ($0 \le c \le 1$) specifies the relative contributions from the social influence and the initial label. φ_{ik} is defined as tie strength, which balances the heterogenous influence from different neighborhoods ($k \in \mathcal{T}_i$) and meets $\sum_{k \in \mathcal{T}_i} \varphi_{ik} = 1$. Overall, SamWalker replaces γ_{ij} with a social context-aware function g parameterized by φ, to which the equation (7) converges:

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

where we collect variables $\gamma_{ij}^{(t)}$ for every user-item pairs (i, j) as a matrix $Y^{(t)}$. Also, we collect φ_{ik} as a matrix Φ, in which $\Phi_{ij} = \varphi_{ij}$ for connected user pairs and $\Phi_{ij} = 0$ 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 $(I - c\Phi)^{-1}$ is a graph or diffusion kernel [58], which has been widely adopted to measure nodes proximity in the network and depends on the tie strength parameters φ for every social ties. Overall, SamWalker can capture the heterogeneous social influence between users and reduce the number of variational parameters from $O(n \times m)$ to $O(|E|)$, where $|E|$ denotes the number of ties. By iteratively learning the transformation function and user's preference, SamWalker can adaptively specify different weights to different data based on users' social contexts.

4.1 Informative Sampler based on personalized random walk

As mentioned above, inferring user's preference from implicit feedback data is time-consuming due to the massive volume of the unobserved data. Thus, for efficient recommendation, SamWalker employs stochastic inference based on data sampling. That is, in each iteration, we update the preference parameters θ based on the estimated gradient from only a subset of the data. Note that different data may have different confidence weights. This makes the data sampling strategy important because it determines which data are used to update parameters and how often. Intuitively, the informative data with larger confidence γ_{ij} should be sampled with larger probability, since these terms make more contribution to the objective function. In fact, we have the following lemma:

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

The proof are presented in appendix.

Also, the sampler with distribution $p_{ij} \propto \gamma_{ij}$ can speed up gradient estimation. As we can see from the following equation (9) (here ℓ_{ij} as shorthand for $\ell(x_{ij}, \sigma(u_i^T v_j))$), when the data sampling probability is proportional to the data confidence ($p_{ij} = \gamma_{ij}/G$), the confidence weights γ_{ij} can be absorbed into the sampling bias. Thus, in each iteration the gradient estimator does not need to calculate current learned confidence weights γ_{ij} to scale down the contribution from different data, which saves much time.

$$\frac{\partial L}{\partial \theta} = \sum_{i \in U, j \in I} \gamma_{ij} \frac{\partial \ell_{ij}}{\partial \theta} = \sum_{i \in U, j \in I} \frac{p_{ij}}{G} \frac{\partial \ell_{ij}}{\partial \theta} = E_p[\sum_{(a,b) \in p} \frac{1}{G} \frac{\partial \ell_{ab}}{\partial \theta}] \quad (9)$$

A naive implement of informative sampler with distribution $p_{ij} \propto \gamma_{ij}$ is to estimate and rank the current learned confidence weights γ_{ij} for every user-item pairs and then pick out the informative data based on γ_{ij}. It is apparently inefficient and can't satisfy practical requirement. To avoid having to estimate $n \times m$ confidence weights, we propose a subtle sampling strategy based on random walk along the network. Here we do some transformation of equation (8) to bring some insights:

$$\begin{aligned} Y &= (I - c\Phi)^{-1}(1 - c)X \\ &= (1 + c\Phi + (c\Phi)^2 + (c\Phi)^3 \ldots)(1 - c)X \end{aligned} \quad (10)$$

We can simulate the equation (10) and develop following sampling strategy:

For target user i, we perform the personalized random walk from user i to sample the informative feedback data of user i. At each step t of random walk, we are at a certain user u. We have two options: (1) With probability c, we do not continue the random walk. We stay at user u and randomly (uniformly) select a portion of (N_u/β) items that have been consumed by user u, where N_u denotes the number of items consumed by user u. Then we add the feedback data of user i on these selected items into sampled set S. (2) With probability $(1 - c)$, we continue our random walk. We randomly select one of u's connected friends v ($v \in \mathcal{T}_u$) based on personalized tie strength (transition probability) φ_{uv} and walk to v for the next walk step.

It is easy to check that the (i, v)-th element of matrix $(c\Phi)^t(1-c)$ is the probability of starting from source user i and terminating at user v in step t. Further, the (i, j)-th element of matrix $(c\Phi)^t(1-c)X$ represents the sampled probability of the user-item feedback data x_{ij} in step t. Sum over the probability in different step, we have the sampled probability of the data as follow:

$$P = \sum_{t=0}^{\infty} (c\Phi)^t (1 - c)X/\beta \propto Y \quad (11)$$

which is proportional to the data confidence. Overall, our sampling strategy can efficiently draw informative training instances to speed up inference and reduce sampling variance.

In practice, we usually conduct α times random walk for each user to achieve more reliable mini-batch stochastic optimization. The paremeters α and 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 we go very far from the source user ($t > t_m$). Concretely, when $t > t_m$, we walk to random user in the system (uniformly) and generated sampled data as option (1).

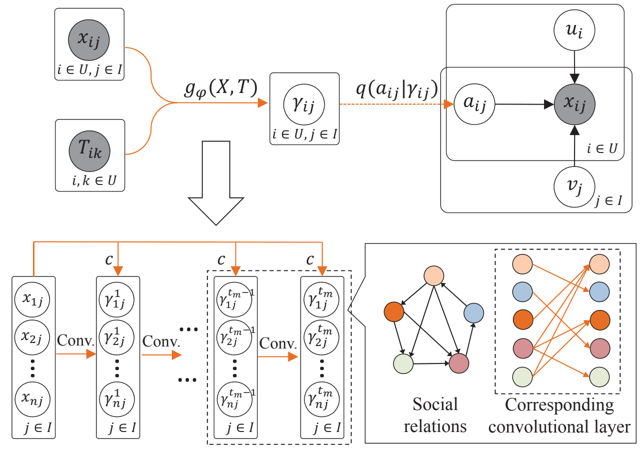


Figure 2: A schematic view of proposed model SamWalker (upper), the transformation function $g_\varphi(X, T)$ based on multi-layer convolutions (bottom-left), and the convolutional layer for the specific social network (bottom-right).

4.2 Inference of personalized tie strength φ

Chaney et al. [6] and Wang et al. [45] claim that different social ties may have different influence strength. For example, we are more likely to get information from our close friends than from our acquaintances. Thus, the parameters φ of tie strength in transformation function tend to be diverse and is thus hard to be assigned manually. To deal with this problem, SamWalker fits transformation function and learns its parameters φ from the data.

We achieve this by optimizing the lower bound of margin likelihood (equation 5) w.r.t tie strength φ_{ij} based on gradient methods. However, directly deriving gradient from transformation function g_φ (equation (8)) involves matrix inversion and suffers from low efficiency. Alterativelly, as illustrated in Figure 2, SamWalker iteratively simulates information spread as equation (7), and constructs equivalent social-based convolutional neural network [28, 55] to infer the personalized tie strength. In fact, using convolutional neural network to model information spread in social network is natural: The nodes in t-th layer can represent users' exposure $\gamma_{ij}^{(t)}$ in t-th iteration. The forward process from layer t to $t + 1$ models t-th information spread. That is, nodes/users collect the information from their social connected friends (neighbors) and generate new representation in $t + 1$ layer:

$$\gamma_{ij}^{(t+1)} = ac(\sum_{k \in N_i} c\varphi_{ik}\gamma_{kj}^{(t)} + (1 - c)x_{ij}) \quad (12)$$

where $ac(.)$ is an activation function. In SamWalker, we use identity activation function $ac(x) = x$. In fact, other activation function can be employed for different prorogation pattern (e.g. sigmoid function for Linear Threshold [11]). Further, attentive mechanism has been employed to balance the social influence from different social

Algorithm 1 Inference of SamWalker

1: Initialize parameters randomly;
2: **while** not converge **do**
3: Sample a set of data S based on the random walk as mentioned in subsection 4.1.
4: Update parameters θ of the preference model based on the estimated gradient from the sampled data (equation (9)).
5: randomly select a portion of N_{SI} items.
6: update personalized tie strength φ based on backward propagation along the neural network for the selected items.
7: **end while**

friends (neighbors). Here we simply use independent attention and reparameterize $\varphi_{ik} = \frac{\exp(w_{ik})}{\sum\limits_{k \in N_i} \exp(w_{ik})}$ using a Softmax transformation. In fact, more sophisticated attentive network can be employed to capture complex and context-aware tie strength [8, 21]. We will leave it in the future work.

With above neural network, backward prorogation can be 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. That is, 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 is presented in Algorithm 1.

4.3 Discussion of the SamWalker

Global influence Vs. local influence. If a user has no connections with others, the method will face degeneration when estimating user's exposure or conducting random walk. To deal with this problem, we borrow the idea from Pagerank and consider a user may also get information from a random user. That is, when estimating user's exposure, users are assumed to link out to all other users but with very weak common tie strength. Thus, we revise equation (7) as follow:

$$\gamma_{ij}^{(t+1)} = (1-c)X_{ij} + c\Big(\sum_{k \in \mathcal{T}_i} \varphi_{ik}\gamma_{kj}^{(t)} + \frac{\varphi_{i0}}{n}\sum_{k \in U}\gamma_{kj}^{(t)}\Big) \tag{13}$$

where we introduce the influence from all other users to iteratively reconstruct user's exposure. Also, the parameters $\varphi_{i0}, \varphi_{ik}$ have been employed to balance the local social influence from connected users and the global social influence from all other users (i.e. $\varphi_{i0} + \sum\limits_{k \in \mathcal{T}_i} \varphi_{ik} = 1$). From another perspective, the additional global term can be considered as the mean priori of users' exposure. As claimed by [37], users' selection behaviour is mainly driven by the local social influence among friends while the global popularity plays a supplementary role driving the behaviour only when there is little local information for the user to refer to. Thus, On the one hand, we introduce the global influence to pull user's exposure closer to the global average. On the other hand, users have their personalized local social contexts and thus are exposed to diversity item information. Correspondingly, local social influence term has been introduced to push users' exposure away from average and towards diversity. Overall, we combine these two effects to better capture user's exposure.

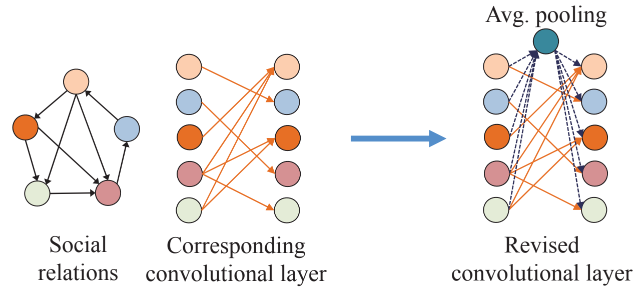


Figure 3: The revised convolutional layer for modeling both local influence and global influence.

Correspondingly, to let our sampling strategy adapt to the revised data confidence, we add random transition into random walk. That is, in each step of random walk, we add another option (3): With probability $c\varphi_{i0}$, we walk to another user ($v \in U$) in the network by random (uniformly) and continue the random walk on v.

Also, as illustrated in Figure 3, with additional pooling units, global influence can be easily integrated into the social-based convolutional neural network to infer the tie strength $\varphi_{i0}, \varphi_{ik}$. Formally, we revise equation (12) as:

$$\begin{aligned} \bar{\gamma}_j^{(t)} &= \frac{1}{n}\sum_{k \in U}\gamma_{kj}^{(t)} \\ \gamma_{ij}^{(t+1)} &= ac\Big(c\Big(\sum_{k \in N_i}\varphi_{ik}\gamma_{kj}^{(t)} + \varphi_{i0}\bar{\gamma}_j^{(t)}\Big) + (1-c)x_{ij}\Big) \end{aligned} \tag{14}$$

Complexity Analysis. The computational time of the inference of SamWalker consists of three parts: (1) In sampling, we need to conduct α times random walk for each user to get sampled data set S. The time for sampling is $O(\alpha n t_m + |S|)$, where n denotes the number of users in the system, 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 loss and gradient for the sampled data based on equation (9). The time to update these parameters is $O(|S|d)$. (3) We infer the tie strength φ based on the gradient back propagation along the neural network for the selected N_{SI} items. The complexity for this part is $O(N_{SI}|E|)$. Hence, the overall computational complexity is $O(\alpha n t_m + |S|d + N_{SI}|E|)$. In practise, 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.

5 EXPERIMENTS AND ANALYSES

In this section, we conduct experiments to evaluate the running time and the recommendation quality of SamWalker. Our experiments are intended to address the following questions:

(Q1) Does SamWalker outperform state-of-the-art recommendation methods?
(Q2) How does proposed social-based sampling strategy perform?
(Q3) Is it beneficial to model the personalized tie strength? Is it beneficial to model the both local social influence and global popularity influence?
(Q4) How does the parameter t_m (the max depth of random walk) affect the recommendation performance?

Table 1: Statistics of three datasets.

Datasets	#Users	#Items	#Links	#User-item interactions
LastFM	1,892	4,489	25,434	52,668
Ciao	5,298	19,301	106,640	138,840
Epinions	21,290	34,075	414,549	333,916

Table 2: The characteristics of the compared methods.

Methods	Social?	Exposure -based?	Sample?	Complexity
WMF(ALS)	\	\	\	$O((n+m)d^3)$
BPR	\	\	√	$O((n+m+\lvert S\rvert)d)$
SBPR	√	\	√	$O((n+m+\lvert S\rvert)d)$
SPF	√	\	\	$O((n+m))d$
CDAE	\	\	√	$O((n+m+\lvert S\rvert)d)$
EXMF	\	√	\	$O(nmd)$
SERec-Bo	√	√	\	$O(nmd)$
SoEXBMF	√	√	\	$O(nmd^2)$
SamWalker	√	√	√	$O(\alpha n t_m + \lvert S\rvert d + N_{SI}\lvert E\rvert)$

5.1 Experimental protocol

Datasets. Three datasets Epinions[3], Ciao[4], LastFM[5] are used in our experiments. These datasets contain users' feedback and social relations. Specifically, the datasets Epinions and Ciao contain users' ratings on movies, while the dataset LastFM contains users' clicks on music. The dataset statistics are presented in Table 1. Similar to [18, 49], 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. Grid search and 5-fold cross validation are used to find the best parameters. In our experiments, we set $\eta_{ij} = 0.5$, $\alpha = 100$, $\beta = 20$, $c = 0.8$, $t_m = 5$ and the decay parameter to 0.1. All experiments are conducted on a server with 2 Intel E5-2620 CPUs and 256G RAM. Our source codes will be available at Github.

Compared methods. We compare SamWalker with following baseline methods. Table 2 concludes the characteristics of them.

- WMF(ALS) [22, 36]: The classic weighted matrix factorization model for implicit feedback data. The corresponding ALS-based [22] algorithm can reduce inference complexity.
- BPR [39]: The classic pair-wise method for recommendation, coupled with matrix factorization. For efficient recommendation, BPR employs uniform sampling strategy to draw the training instances.
- SBPR[57]: SBPR integrates social information into BPR by assuming that the items consumed by connected friends are ranked higher than those not.
- SPF [6]: A social recommendation model that incorporates social influence with users' latent preference based on poisson factorization.
- CDAE [48]: The advanced recommendation method based on Auto-Encoders, which is a generalization of WMF with more flexible components. However, efficient ALS-based inference algorithm are no longer suitable. Thus CDAE employs uniform sampling strategy to draw the training instances.
- EXMF [30]: 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.
- SERec-Bo[44]: A probabilistic model that extends the EXMF model with social influence on user's exposure. Note that in [44] 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 [26]: A probabilistic model that further extends the EXMF model with both social knowledge influence and social consumption influence on user's exposure.

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:

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

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

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

- 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:

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

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

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

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:

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

MRR can be interpreted as the ease of finding all consumed items, as higher numbers indicate the consumed items are higher in the list.

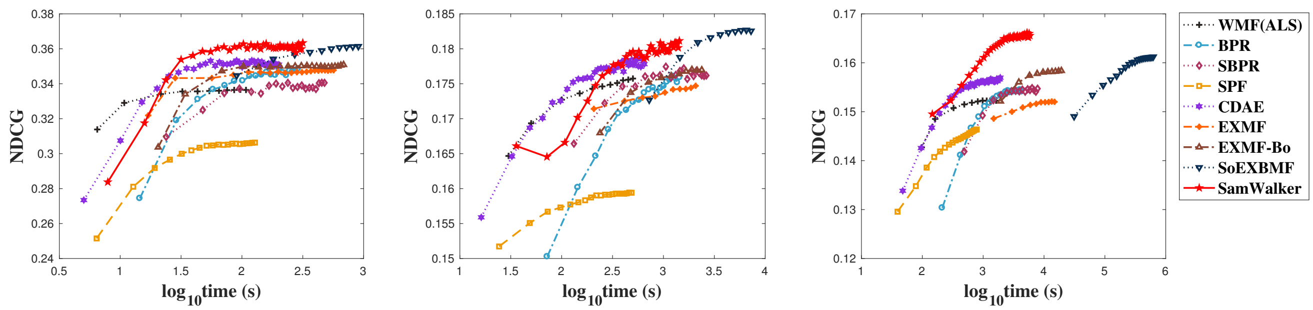
5.2 Performance comparison (Q1)

Table 3 presents the performance of the compared methods in terms of three evaluation metrics. The boldface font denotes the winner in that column. Overall, with few exceptions, SamWalker outperforms all compared methods on all datasets for all metrics. For the sake of

[3] http://www.trustlet.org/epinions
[4] http://www.cse.msu.edu/~tangjili/trust
[5] https://grouplens.org/datasets/hetrec-2011/

Table 3: The performance metrics of the compared methods. The boldface font denotes the winner in that column. The row 'Impv-e' indicates the relative performance gain of our SamWalker compared to the best results among these efficient baselines including WMF(ALS), BPR, CDAE, SPF, SBPR. The row 'Impv-a' indicates the relative performance gain of our SamWalker compared to the best results among all the baselines.

Methods		LastFM				Ciao				Epinions			
		Pre@5	Rec@5	NDCG	MRR	Pre@5	Rec@5	NDCG	MRR	Pre@5	Rec@5	NDCG	MRR
Efficient	WMF(ALS)	0.0928	0.0841	0.3364	0.3552	0.0172	0.0123	0.1757	0.0738	0.0095	0.0087	0.1522	0.0446
	BPR	0.1004	0.0888	0.3485	0.3704	0.0144	0.0125	0.1759	0.0649	0.0087	0.0096	0.1541	0.0406
	SBPR	0.0956	0.0851	0.3405	0.3556	0.0156	0.0124	0.1774	0.0678	0.0088	0.0089	0.1542	0.0411
	SPF	0.0714	0.0622	0.3132	0.2909	0.0081	0.0061	0.1603	0.0379	0.0052	0.0063	0.1464	0.0276
	CDAE	0.1008	0.0890	0.3509	0.3784	0.0144	0.0116	0.1778	0.0647	0.0092	0.0112	0.1570	0.0423
Inefficient	EXMF	0.0957	0.0859	0.3477	0.3665	0.0095	0.0105	0.1747	0.0485	0.0051	0.0082	0.1513	0.0275
	SERec-Bo	0.1018	0.0907	0.3509	0.3787	0.0118	0.0124	0.1770	0.0532	0.0073	0.0101	0.1577	0.0365
	SoEXBMF	0.1108	0.1014	0.3617	0.4140	0.0181	0.0152	**0.1827**	0.0775	0.0120	0.0130	0.1612	0.0537
Proposed	SamWalker	**0.1177**	**0.1072**	**0.3634**	**0.4250**	**0.0182**	**0.0167**	0.1811	**0.0775**	**0.0149**	**0.0184**	**0.1656**	**0.0628**
	Impv-e%	16.8%	20.4%	3.6%	12.3%	5.9%	34.0%	1.9%	5.0%	56.0%	64.2%	5.5%	40.9%
	Impv-a%	6.2%	5.7%	0.5%	2.7%	0.2%	10.4%	-0.8%	0.0%	23.9%	41.1%	2.7%	16.9%



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

Figure 4: NDCG for each method in different steps versus time

clarity, the last two rows of Table 3 also show the relative improvements achieved by SamWalker over the efficient baselines (Impv-e) and over all baselines (Impv-a) respectively. The improvement of SamWalker over the efficient baselines is apparent. Especially on the large dataset Epinions, the improvement is 56.0%, 64.2%,5.5%,40.9% in terms of Pre@5, Rec@5, NDCG, MRR respectively. The improvement of SamWalker over these efficient baselines can be attributed to two aspects: (1) In the real world, users have personalized social contexts and thus are exposed to diverse information. Correspondingly, data have diverse 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. By adaptively fitting fine-grained data confidence weights based on users' social contexts, SamWalker achieves better performance than those baselines with manually coarse-grained confidence weights. (2) Instead of employing pre-defined coarse-grained sampling methods, SamWalker employs random walk-based sampler to adaptively draw informative data, which can reduce sampling variance and achieve better performance. We also conduct specific experiments for different samplers in subsection 5.3 to validate this point.

Also, even although SamWalker employs stochastic optimization to reduce time complexity, SamWalker still beat these exposure-based methods (EXMF, SERec-Bo, SoEXBMF). This result validate the effectiveness of employing social-based variational posterior of user's exposure, which models the heterogenous influence from both direct and indirect social friends (neighbors). Although SoEXBMF can also capture such influence, it costs much more running time and requires pre-defined community distribution. In fact, accurate community distribution is not easy to get, which affects the performance of SoEXBMF model.

Runtime vs. NDCG. Figure 4 depicts running time (X-axis) vs. NDCG (Y-axis) of the nine compared recommendation methods. As we can see, generally, SamWalker achieves best performance. The powerful competitor is CDAE, which models user's preference based on collaborative auto-encoder. Although CDAE has better NDCG than SamWalker at first, SamWalker overtakes CDAE soon and before CDAE gets convergence. Especially on large dataset Epinions, SamWalker outperforms CDAE almost in all time and achieves much better performance than CDAE when they get convergence. Also, we observe that these exposure-based methods (EXMF,SERec,SoEXBMF) achieve good performance but cost a lot of time. Although these methods can learn the personalized confidence of the data, they need infer $n \times m$ parameters of user's exposure, which will suffer from low efficiency and overfitting. By employing social-based posterior structure of exposure and random walk-based sampler, our method SamWalker can beat EXMF,SERec,SoEXBMF in both running time and recommendation performance.

Table 4: Average empirical variance of the estimated gradients using different sampling strategy. 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					
		Batchsize=5$\|X^{(1)}\|$			Batchsize=$\|X^{(1)}\|$		
		50 It.	100 It.	500 It.	50 It.	100 It.	500 It.
S-allunion	$p_{ij} = 1/(n \times m)$	2.3793	2.5640	2.6663	11.8566	13.2071	13.2475
S-balunion	$p_{ij} = 1/(2\|X^{(x_{ij})}\|)$	0.4154	0.4941	0.5368	2.0834	2.4732	2.6779
S-itempop	$p_{ij} = I[x_{ij} = 1]/(2\|X^{(1)}\|) + I[x_{ij} = 0]c_j^{(1)}/(2 \sum_{1 \le b \le m} c_b^{(1)})$	0.1720	0.1938	0.2190	0.8702	0.9890	1.1052
S-cobias	$p_{ij} = r_i^{(1-x_{ij})} c_j^{(1-x_{ij})}/(2 \sum_{1 \le a \le n} \sum_{1 \le b \le m} I[x_{ab} = x_{ij}] r_i^{(1-x_{ij})} c_j^{(1-x_{ij})})$	0.1617	0.1874	0.2031	0.8074	0.9663	1.0647
Random Walk	$p_{ij} \propto \gamma_{ij}$	0.1358	0.1464	0.1510	0.6469	0.6977	0.7038

5.3 Performance of the different samplers (Q2)

In this subsection, we compare our sampler with other sampling strategies including: (1) S-allunion, the uniform sampling strategy; (2) S-balunion [19, 36], which samples un-observed data (zeros) and observed data (ones) with equal probability to deal with unbalance data problem; (3) S-itempop [53], which samples data based on item popularity; (4) S-cobias [19], whose probability of sampling a un-observed data (zero) at location (i, j) is proportional to the number of ones in the i-th row (the number of user's consumption) and the number of zeros in the j-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 gradient of our objective w.r.t θ by using different sampling strategy. To do this, we train our model for 50,100 or 500 epochs on dataset LastFM. After training, we generate mini-batch with various sampling strategy and calculate un-biased estimated gradients of our objective w.r.t θ. We repeat this proceeding for 1000 times and calculate the variance of the estimated gradients for different sampling strategy. 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 coincide with our theory analysis in section 4.1. 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 models the diverse data confidence and the personalized tie strength. At first, SamWalker keeps relative similar data confidence. With training proceeding, driven by the data, the confidence weights and the tie strength in SamWalker exhibit more and more heterogeneity. Thus, the variance of the estimated gradients will become larger.

Performance. Figure 5 presents the NDCG of SamWalker on dataset LastFM with different sampling strategies versus the number of iteration and running time. As we can see, our adaptive random walk-based sampler performs better than others in all convergence, speed and recommended performance. One reason is that our sampler has low variance, as presented in Table 4. Another reason is that in our sampler the confidence weights γ_{ij} are absorbed into the bias of the sampling distribution. Thus, the gradient estimator does not need to calculate γ_{ij} in each iteration to scale down the contribution from the different data, which saves much time.

5.4 Heterogenous social influence (Q3)

To show the effect of modeling heterogenous social influence, we compare SamWalker with its three special case: (1) SamWalker-ho, the special case of SamWalker with homogeneous tie strength. (2) SamWalker-g (item-pop): the special case which leave out heterogenous local social influence and models user's exposure just based on item popularity. (3) SamWalker-l: the special case which leaves out global influence. The performance of these special cases comparing with SamWalker is presented in Figure 6.

Personalized tie strength. As shown in Figure 6, SamWalker with personalized tie strength consistently achieves better performance than its special case SamWalker-ho with the homogeneous tie strength in all three datasets. These results support our intuition that different ties may have different influence strength. For example, the items consumed by our close friends that have high frequency of interactions, are more likely to come to our attention than those items consumed by our acquaintances.

Global influence Vs. local influence. Figure 6 also shows that SamWalker outperforms its special cases SamWalker-g and SamWalker-l. The result validates our point that user's exposure will be affected by both local social influence and global popularity influence. Especially, as claimed by some sociologic related works [37], user's social relations have become a major information resource when he select items to consume. Thus, we observe the apparent improvement of SamWalker over the special case SamWalker-g (item-pop) which leaves out local social influence.

5.5 Effect of parameter t_m (Q4)

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 7, as t_m become larger, with few exception, the performance will become better first. The reason is that the information about items propagates along the social network. User's exposure will also be influenced by his high order neighbors. However, when t_m surpasses a threshold ($t_m > 5$),

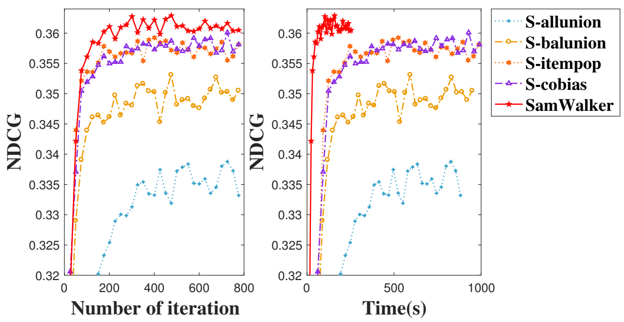
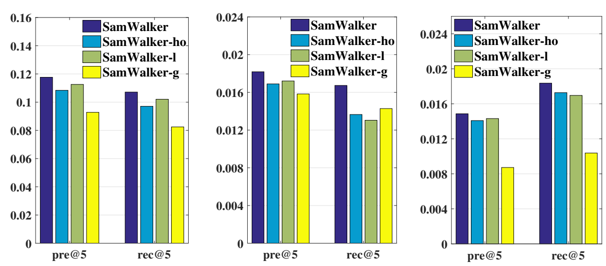
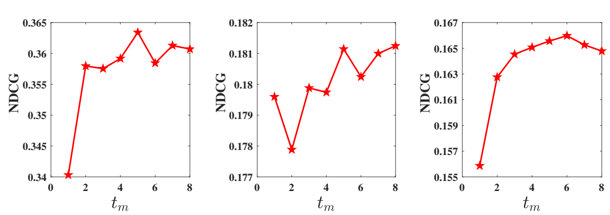


Figure 5: NDCG for different sampling strategy versus the number of iterations and running time



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

Figure 6: Performance comparison of SamWalker with its three special cases



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

Figure 7: Performance comparison with varying t_m

the performance becomes unaffected or even worse with further increase of t_m. In fact, based on the idea of "six-degrees of separation [43]", the random walk with length 5 can visit most of the nodes in the social network. Too deep random walk will not bring more usual information and sometimes bring some noises.

6 CONCLUSIONS

In this paper, we present a novel recommendation method SamWalker that leverages social information to simultaneously learn the personalized data confidence and draw informative training instances. On the one hand, SamWalker models the posterior of user's exposure with a social context-aware function, which can reduce the number of learned parameters and adaptively specify different confidence weights to different data. On the other hand, we propose a personalized random walker-based sampling strategy to draw informative training instances to speed up inference and reduce sampling variance. The personalized transition probability (tie strength) can be inferred efficiently based on our attentive neural network. The experimental results on three real-world datasets demonstrate the superiority of SamWalker over existing methods.

One interesting direction for future work is to explore dynamic exposure-based recommendation. In the real world, users' preference, exposure and relations may evolve over time. Further, we did not examine other attentive structure when modeling tie strength. It will be also interesting to develop more complex attentive model and explore context-aware social influence.

A PROOF OF LEMMA 1

Given the confidence weights γ_{ij}, we have the following un-biased gradient estimator of the objection (equation 5) based on the specific sampler p_{ij}:

$$\nabla_\theta L = \sum_{i\in U, j\in I} \gamma_{ij}\nabla_\theta \ell_{ij} = E_p\Big[\frac{nm}{|S|}\sum_{a,b\in p}\frac{\gamma_{ab}\nabla_\theta \ell_{ab}}{p_{ab}}\Big] \tag{20}$$

where ℓ_{ij} is shorthand for $\ell(x_{ij}, \sigma(u_i^T v_j))$ and $|S|$ denotes mini-batch size. Then, we have following variance of the gradient estimator:

$$\begin{aligned}
&Var_{(a,b)\sim p}\Big[\frac{nm}{|S|}\sum_{a,b\in p}\frac{\gamma_{ab}\nabla_\theta \ell_{ab}}{p_{ab}}\Big] \\
&= \frac{n^2m^2}{|S|} Var_{(a,b)\sim p}\Big[\frac{\gamma_{ab}\nabla_\theta \ell_{ab}}{p_{ab}}\Big] \\
&= \frac{n^2m^2}{|S|}\Big(E_p\Big[\Big(\frac{\gamma_{ab}\nabla_\theta \ell_{ab}}{p_{ab}}\Big)^2\Big] - {E_p}^2\Big[\frac{\gamma_{ab}\nabla_\theta \ell_{ab}}{p_{ab}}\Big]\Big) \\
&= \frac{n^2m^2}{|S|} E_p\Big[\Big(\frac{\gamma_{ab}^2}{p_{ab}^2}\nabla_\theta^2 \ell_{ab}\Big] - \Big(\sum_{i\in U, j\in I}\gamma_{ij}\nabla_\theta \ell_{ab}\Big)^2
\end{aligned} \tag{21}$$

It is untractable to find optimized p_{ij} based on equation (21) due to the unknown of the terms $\nabla_\theta^2 \ell_{ab}$. But we can derive the tight upper-bound of the variance based on Cauchy inequality as follow:

$$\begin{aligned}
&Var_{(a,b)\sim p}\Big[\frac{nm}{|S|}\sum_{a,b\in p}\frac{\gamma_{ab}\nabla_\theta \ell_{ab}}{p_{ab}}\Big] \\
&= \frac{n^2m^2}{|S|} E_p\Big[\Big(\frac{\gamma_{ab}^2}{p_{ab}^2}\nabla_\theta^2 \ell_{ab}\Big] - \Big(\sum_{i\in U, j\in I}\gamma_{ij}\nabla_\theta \ell_{ab}\Big)^2 \\
&\le \frac{n^2m^2}{|S|}\sum_{ij}\frac{\gamma_{ij}^2}{p_{ij}} \times E_{p,\ell_{ab}}[\nabla_\theta^2 \ell_{ab}] - \Big(\sum_{i\in U, j\in I}\gamma_{ij}\nabla_\theta \ell_{ab}\Big)^2
\end{aligned} \tag{22}$$

Also, we can find the minimum of $\sum_i \frac{\gamma_{ij}^2}{p_{ij}}$ as follow:

$$\begin{aligned}
\sum_i \frac{\gamma_{ij}^2}{p_{ij}} &= \sum_i \frac{\gamma_{ij}^2}{p_{ij}}\sum_i p_{ij} \\
&\ge \Big(\sum_i \frac{\gamma_{ij}}{\sqrt{p_{ij}}}\sqrt{p_{ij}}\Big)^2 = \Big(\sum_i \gamma_{ij}\Big)^2
\end{aligned} \tag{23}$$

The equation holds if and only if $p_{ij} \propto \gamma_{ij}$. Now, the sampling strategy has the lowest upper-bound of the variance.

REFERENCES

[1] Luca Maria Aiello, Alain Barrat, Ciro Cattuto, Rossano Schifanella, and Giancarlo Ruffo. 2012. Link creation and information spreading over social and communication ties in an interest-based online social network. *EPJ Data Science* 1, 1 (2012), 12.

[2] Luca Maria Aiello, Alain Barrat, Rossano Schifanella, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. 2012. Friendship prediction and homophily in social media. *ACM Transactions on the Web (TWEB)* 6, 2 (2012), 9.

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

[4] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. 2017. A generic coordinate descent framework for learning from implicit feedback. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1341–1350.

[5] Pablo Castells, Neil J Hurley, and Saul Vargas. 2015. Novelty and diversity in recommender systems. In *Recommender Systems Handbook*. Springer, 881–918.

[6] Allison JB Chaney, David M Blei, and Tina Eliassi-Rad. 2015. A probabilistic model for using social networks in personalized item recommendation. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 43–50.

[7] Jiawei Chen, Can Wang, Martin Ester, Qihao Shi, Yan Feng, and Chun Chen. 2018. Social recommendation with missing not at random data. In *Data Mining (ICDM), 2018 IEEE International Conference on*. IEEE, 29–39.

[8] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 335–344.

[9] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. 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, 767–776.

[10] Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. 2013. Information and influence propagation in social networks. *Synthesis Lectures on Data Management* 5, 4 (2013), 1–177.

[11] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable influence maximization in social networks under the linear threshold model. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 88–97.

[12] Jingtao Ding, Fuli Feng, Xiangnan He, Guanghui Yu, Yong Li, and Depeng Jin. 2018. An improved sampler for bayesian personalized ranking by leveraging view data. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 13–14.

[13] Jennifer Golbeck. 2006. Generating predictive movie recommendations from trust in social networks. In *International Conference on Trust Management*. Springer, 93–104.

[14] Jennifer Golbeck. 2009. Trust and nuanced profile similarity in online social networks. *ACM Transactions on the Web (TWEB)* 3, 4 (2009), 12.

[15] Ido Guy. 2015. Social recommender systems. In *Recommender Systems Handbook*. Springer, 511–543.

[16] Ido Guy, Alejandro Jaimes, Pau Agulló, Pat Moore, Palash Nandy, Chahab Nastar, and Henrik Schinzel. 2010. Will recommenders kill search?: recommender systems-an industry perspective. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 7–12.

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

[18] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. 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, 549–558.

[19] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Stochastic inference for scalable probabilistic modeling of binary matrices. In *International Conference on Machine Learning*. 379–387.

[20] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *The Journal of Machine Learning Research* 14, 1 (2013), 1303–1347.

[21] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1531–1540.

[22] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. Ieee, 263–272.

[23] Bert Huang, Angelika Kimmig, Lise Getoor, and Jennifer Golbeck. 2013. A flexible framework for probabilistic models of social trust. In *International conference on social computing, behavioral-cultural modeling, and prediction*. Springer, 265–273.

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

[25] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender systems: an introduction*. Cambridge University Press.

[26] Chen Jiawei, Feng Yan, Ester Martin, Zhou Sheng, Chen Chun, and Can Wang. 2018. 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, 953–962.

[27] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. 2018. News recommender systems–Survey and roads ahead. *Information Processing & Management* (2018).

[28] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[29] Dongsheng Li, Chao Chen, Qin Lv, Hansu Gu, Tun Lu, Li Shang, Ning Gu, and Stephen M Chu. 2018. 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, 741–751.

[30] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 951–961.

[31] Moshe Lichman and Padhraic Smyth. 2018. Prediction of Sparse User-Item Consumption Rates with Zero-Inflated Poisson Regression. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 719–728.

[32] Hao Ma, Irwin King, and Michael R Lyu. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 203–210.

[33] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 931–940.

[34] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 287–296.

[35] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[36] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. *Proceedings - IEEE International Conference on Data Mining, ICDM* (2008), 502–511. https://doi.org/10.1109/ICDM.2008.16

[37] Xue Pan, Lei Hou, and Kecheng Liu. 2017. Social influence on selection behaviour: Distinguishing local-and global-driven preferential attachment. *PloS one* 12, 4 (2017), e0175761.

[38] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 273–282.

[39] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[40] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: introduction and challenges. In *Recommender systems handbook*. Springer, 1–34.

[41] Yelong Shen and Ruoming Jin. 2012. 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, 1303–1311.

[42] Jiliang Tang, Huiji Gao, and Huan Liu. 2012. mTrust: discerning multi-faceted trust in a connected world. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 93–102.

[43] Jeffrey Travers and Stanley Milgram. 1967. The small world problem. *Phychology Today* 1, 1 (1967), 61–67.

[44] Menghan Wang, Xiaolin Zheng, Yang Yang, and Kun Zhang. 2018. Collaborative Filtering With Social Exposure: A Modular Approach to Social Recommendation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*.

[45] Xin Wang, Steven CH Hoi, Martin Ester, Jiajun Bu, and Chun Chen. 2017. Learning personalized preference of strong and weak ties for social recommendation. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1601–1610.

[46] Xin Wang, Wei Lu, Martin Ester, Can Wang, and Chun Chen. 2016. Social Recommendation with Strong and Weak Ties. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM,

5–14.
[47] Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, Vol. 11. 2764–2770.
[48] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 153–162.
[49] Lin Xiao, Zhang Min, Zhang Yongfeng, Liu Yiqun, and Ma Shaoping. 2017. Learning and transferring social and item visibilities for personalized recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 337–346.
[50] Bo Yang, Yu Lei, Dayou Liu, and Jiming Liu. 2013. Social collaborative filtering by trust. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2747–2753.
[51] Xiwang Yang, Harald Steck, and Yong Liu. 2012. Circle-based recommendation in online social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1267–1275.
[52] Weilong Yao, Jing He, Guangyan Huang, and Yanchun Zhang. 2014. Modeling dual role preferences for trust-aware recommendation. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 975–978.
[53] Hsiang-Fu Yu, Mikhail Bilenko, and Chih-Jen Lin. 2017. Selection of negative samples for one-class matrix factorization. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 363–371.
[54] Lu Yu, Chuxu Zhang, Shichao Pei, Guolei Sun, and Xiangliang Zhang. 2018. Walkranker: A unified pairwise ranking model with multiple relations for item recommendation. AAAI.
[55] Shuai Zhang, Lina Yao, and Aixin Sun. 2017. Deep learning based recommender system: A survey and new perspectives. *arXiv preprint arXiv:1707.07435* (2017).
[56] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. 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, 785–788.
[57] Tong Zhao, Julian McAuley, and Irwin King. 2014. 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, 261–270.
[58] Denny Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. In *Advances in neural information processing systems*. 321–328.
[59] Cai-Nicolas Ziegler and Jennifer Golbeck. 2007. Investigating interactions of trust and interest similarity. *Decision support systems* 43, 2 (2007), 460–475.