

ImWalkMF: Joint Matrix Factorization and Implicit Walk Integrative Learning for Recommendation

Chuxu Zhang[†], Lu Yu[‡], Xiangliang Zhang[‡] and Nitesh Chawla[†]

[†]University of Notre Dame, Notre Dame, IN, USA

Email: {czhang11, nchawla}@nd.edu

[‡]King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

Email: {lu.yu, xiangliang.zhang}@kaust.edu.sa

Abstract—Data sparsity and cold-start problems are prevalent in recommender systems. To address such problems, both the observable explicit social information (e.g., user-user trust connections) and the inferable implicit correlations (e.g., implicit neighbors computed by similarity measurement) have been introduced to complement user-item ratings data for improving the performances of traditional model-based recommendation algorithms such as matrix factorization. Although effective, (1) the utilization of the explicit user-user social relationships suffers from the weakness of unavailability in real systems such as Netflix or the issue of sparse observable content like 0.03% trust density in Epinions, thus there is no or little explicit social information that can be employed to improve baseline model in real applications; (2) the current similarity measurement approaches focus on inferring implicit correlations between a user (item) and their direct neighbors or top-k similar neighbors based on user-item ratings bipartite network, so that they fail to comprehensively unfold the indirect potential relationships among users and items. To solve these issues regarding both explicit/implicit social recommendation algorithms, we design a joint model of matrix factorization and implicit walk integrative learning, i.e., ImWalkMF, which only uses explicit ratings information yet models both direct rating feedbacks and multiple direct/indirect implicit correlations among users and items from a random walk perspective. We further propose a combined strategy for training two independent components in the proposed model based on sampling. The experimental results on two real-world sparse datasets demonstrate that ImWalkMF outperforms the traditional regularized/probabilistic matrix factorization models as well as other competitive baselines that utilize explicit/implicit social information.

I. INTRODUCTION

Recommender systems have become an indispensable technique for filtering and recommending information or items to personalize to users' preferences or needs, such as product recommendation at Amazon and movie recommendation at Netflix or music recommendation at Pandora or even disease prediction systems [1]. Various approaches [2]–[8] based on matrix factorization (MF) have been proposed to solve the problem of ratings prediction and make recommendations by only using user-item ratings information. To improve the recommendation performance, recent works [9]–[16] employ the observable explicit social relationships (e.g., trust links among online users) to enhance MF framework and build *social recommender systems*. Besides, implicit correlations information (e.g., top-k similar neighbors) induced by similarity measurement based approaches [17], [18] is utilized to im-

prove MF and build the so-called *implicit social recommender systems*.

Social recommender systems make usage of the trustable social relationships among users to address the sparsity issue of ratings data, and thus improve the user preference prediction by considering not only a users' rating behavior, but also the tastes of a user's trustable social neighbors. For example, in [12], a user social regularization term is integrated into the objective function of MF to help shape the users' latent space. However, the utilization of the explicit user-user social connections suffers from two main weaknesses: (a) there is no available indication about reliable social relationship in most real-life systems such as Netflix or Ebay or (even there is) the explicit relationship indication is usually very sparse (e.g., the trust density in Epinions is 0.03%), thus most of the social recommendation algorithms can not be applied to real systems; (b) an active user can be socially connected with others who have different taste/preference [18] so that social relationship fails to encode the comprehensive correlation between the diverse tastes of two users toward different kinds of items. As for the implicit social recommender systems, they infer and incorporate implicit correlations information into MF based on the explicit rating feedbacks. For instance, in [18], an implicit network embedding method CUNE is proposed to compute similarities among users and generate top-k similar neighbors of each user and further incorporate them into MF. Although enhancing MF with inferred correlations, current implicit social recommendation approaches have two main limitations: (a) the rating-based similarity measurements (e.g., Pearson correlation coefficient) are easy to find direct neighbors yet provide no correlation information for non-neighbor nodes on user-item ratings bipartite network; (b) the methods (e.g., CUNE) generating top-k implicit neighbors ignore correlations between a user and their non-top-k neighbors which may still contain some potential useful information, so that they fail to explore implicit information comprehensively.

To resolve the above issues regarding both explicit and implicit social recommender systems, we propose to extract multiple implicit and reliable correlations among users and items by only using ratings information. Specifically, we manage users' positive feedbacks (relatively large ratings) on items as a user-item implicit bipartite network (U-I-Net) and utilize random walk sampling on U-I-Net to generate a

set of node sequences. Each random walk sequence implies multiple direct/indirect correlations among users and items within the walk. Next, we design a joint model ImWalkMF of MF and implicit walk integrative learning (IWIL) based on the collected random walk set. The MF component of ImWalkMF formalizes users' direct rating feedbacks on items by using standard square loss. Besides, the IWIL component of ImWalkMF formalizes multiple direct/indirect correlations among users and items from both user and item levels by introducing a user-user (item-item) pull loss function and a user-item (item-user) push loss function. Thus ImWalkMF comprehensively models both direct rating feedbacks and useful implicit information. In order to solve the challenge of training ImWalkMF made of two independent components using different data samples, we propose a combined strategy based on sampling to train the joint model and optimize the latent factors of users and items. Further evaluated experiments verify the effectiveness of ImWalkMF in recommendation. In summary, our main contributions are as follows:

- We innovatively introduce random walk sampling to collect a set of node sequences based on user-item implicit bipartite network that implies multiple implicit and reliable correlations. Unlike previous work that focus on computing similarities and inferring limited implicit relationship among users (items), it captures comprehensive information to complement user-item ratings data.
- Based on the set of random walk sequences, we propose a joint recommendation model ImWalkMF for modeling both rating feedbacks and multiple implicit correlations among users and items, and further design a combined strategy for training ImWalkMF based on sampling.
- We conduct extensive experiments to evaluate the performance of ImWalkMF. The results show that ImWalkMF largely improves the traditional regularized/probabilistic matrix factorization models, and outperforms the competitive baselines that utilize explicit/implicit social information.

The rest of the paper is organized as follows. Section II reviews the background and related work of this paper. Section III describes the proposed method and algorithm. The results from extensive experiments are presented in Section IV, followed by the conclusion in Section V.

II. BACKGROUND

In this section, we briefly review the background and related work including recommender systems, metric learning and random walk sampling on networks.

A. Matrix Factorization for Collaborative Filtering

Traditional collaborative filtering (CF) techniques [19], [20] focus on computing rating-based similarities (e.g., cosine similarity) among users (items) and find the most similar neighbors to the query user (item). Over the past decade, the matrix factorization (MF) [2], [4], [6], [8] has perhaps become the most popular CF approach due to its superior performance. The ratings of m users on n items are denoted by an $m \times n$

matrix \mathbf{r} . MF tends to approximate \mathbf{r} by a multiplication of d -rank user and item latent factor vectors, i.e. $\mathbf{r} \approx \mathbf{z}^T \mathbf{q}$, where $\mathbf{z} \in \mathbb{R}^{d \times m}$ and $\mathbf{q} \in \mathbb{R}^{d \times n}$ with $d \ll \min(m, n)$. Thus the rating of user u_i on item v_j can be predicted by the inner product of the specific user latent factor \mathbf{z}_i (i -th column vector of \mathbf{z}) and item latent factor \mathbf{q}_j (j -th column vector of \mathbf{q}), i.e. $\hat{r}_{ij} = \mathbf{z}_i^T \mathbf{q}_j$. The formulation leads to the optimization problem that minimizes the square loss on the set of known ratings:

$$\min_{\mathbf{z}^*, \mathbf{q}^*} \sum_{r_{ij} \in \tilde{R}} (r_{ij} - \mathbf{z}_i^T \mathbf{q}_j)^2 + \frac{\lambda}{2} (\|\mathbf{z}_i\|^2 + \|\mathbf{q}_j\|^2),$$

where \tilde{R} is the set of known ratings, i.e., training data, and λ is the regularization parameter for controlling model complexity to avoid over-fitting. Gradient based approaches can be applied to model training since the objective function is derivable. Besides ratings prediction, MF has been proved to be effective in top-N recommendations [21]–[24].

B. Social Recommendation Systems

Recently, due the prevalence of Web 2.0 based social networks, the research of social recommender systems has attracted a lot of attention. Many novel social recommendation algorithms [9]–[13], [15] based on MF have been proposed to improve the recommendation performance. For example, one popular model SoReg [12] incorporates a social regularization into MF and is formally defined as:

$$\min_{\mathbf{z}^*, \mathbf{q}^*} \sum_{r_{ij} \in \tilde{R}} (r_{ij} - \mathbf{z}_i^T \mathbf{q}_j)^2 + \frac{\lambda}{2} (\|\mathbf{z}_i\|^2 + \|\mathbf{q}_j\|^2) + \alpha \sum_{u_k \in F(i)} \text{sim}(i, k) \|\mathbf{z}_i - \mathbf{q}_k\|^2,$$

where $F(i)$ denotes the set of social neighbors of user u_i , parameter α controls the extent of social constraint, and $\text{sim}(i, k)$ represents the precomputed similarity score between u_i and their friend u_k . The extra regularization term tends to reduce the distance between the latent factors of each user and their social neighbors, which helps shape the latent space of users and improve the recommendation performance. More importantly, the framework can be easily modified or extended by other indirect implicit information. For example, the explicit social constraint can be replaced by implicit correlations generated by predefined similarity measurement [17] or network embedding technique [18] when the explicit social connections are not available. In addition, item based constraints can be introduced to further enhance the model if category or tags information [25] is available.

C. Metric Learning for k -Nearest Neighbors Classification

Metric learning algorithms [26]–[29] generate a distance metric that captures the important relationship among data and have been widely applied to many applications such as the nearest neighbors classification or document retrieval. One of the most well known metric learning models for k -nearest neighbors classification is the large margin nearest neighbor (LMNN) [27], which tends to learn a metric that minimizes the number of differently labeled impostors of the given input data point. Specifically, LMNN defines two loss functions, i.e.,

pull loss and *push loss*, and combines them to generate the complete loss function. The *pull loss* of the given input i is formulated as:

$$\varphi_{pull}(i) = \sum_{j \rightsquigarrow i} d(i, j)^2,$$

where $d(\cdot)$ is the distance metric between two input data points and $j \rightsquigarrow i$ denotes that j is i 's target neighbor. Besides, the *push loss* of the given input i is defined as:

$$\varphi_{push}(i) = \sum_{j \rightsquigarrow i} \sum_k (1 - y_{ik}) [1 + d(i, j)^2 - d(i, k)^2]_+,$$

where $y_{ik} = 1$ if two data points i and k belong to the same class, otherwise $y_{ik} = 0$, and $[x]_+ = \max(x, 0)$ is the standard hinge loss. Besides the nearest neighbors classification, the idea of LMNN has been adopted in question answering systems [30], [31], recommender systems [32], author identification [33], etc.

D. Random Walk Based Information Integration

The random walk based approaches were originally used in network sampling [34]–[36] and influential nodes identification [37]–[39]. In recent years, the random walk method has been applied to network embedding [40], [41], question answering system [30], [31], etc. due to its effectiveness in integrating multiple correlations information in networks. Generally, a random walk rooted at a specific node is a stochastic process with a sequence of random variables, such that each variable is a node chosen randomly from the neighbors of its predecessor. In analogy with a sentence in corpus, the random walk node sequences could be further fed to Word2Vec model [42] for embedding learning. Besides, the homogeneous random walk sequence can be easily extended in heterogeneous context and be applied to other applications such as question retrieval (contains user, question, answer nodes) in question answering system or author identification (contains author, paper, venue, keyword nodes) in academic search service.

III. PROPOSED METHOD

In this section, we present the ImWalkMF method that contains three consecutive steps: 1) constructing user-item implicit bipartite network; 2) generating user-item random walk sequences; and 3) designing the joint model of matrix factorization and implicit walk integrative learning. Figure 1 gives an illustration of the whole process, which we detail below.

A. User-Item Implicit Bipartite Network Construction

At the first step of ImWalkMF, we extract user-item implicit feedbacks by using the available ratings information. The user-item online ratings are usually represented by an $m \times n$ rating matrix \mathbf{r} where r_{ij} denotes that user u_i ever gave rating r_{ij} to item v_j . The value of r_{ij} denotes the extent of u_i 's preference on v_j and a relatively large score means a positive indication. Similar to the concept of implicit feedbacks [43] used in previous work, we utilize a user-item implicit bipartite network (U-I-Net) where connections only exist between u_i and v_j if

and only if $r_{ij} \geq \sigma$ ($\sigma = 3$ in this work when ratings are from 1 to 5), to denote the corresponding implicit information. Step 1 of Figure 1 shows an example. We extract all positive feedbacks (in red color, $r_{ij} \geq 3$ in scale 5) from an 8×8 rating matrix and represent them by an 8-8 U-I-Net. The U-I-Net is unweighted since we regard each positive feedback as the same extent of implicit indication and it captures preference relations of each user with respect to different items.

B. User-Item Random Walk Sequences Generation

To conduct the second step of ImWalkMF, we perform random walk samplings over U-I-Net and collect a set of node sequences. Formally, given a root node c_0 , we simulate a random walk (with length L) $\mathbf{w}_{c_0} = \{c_0, c_1, c_2, \dots, c_{L-1}\}$ via conducting alternation between users and items. Denote the i -th node in the walk as c_{i-1} and it is selected from the neighbors of its predecessor c_{i-2} using:

$$p(c_{i-1} = x | c_{i-2} = y) = \frac{1}{|Nb(y)|},$$

where $Nb(y)$ is c_{i-2} 's neighbor set in U-I-Net. Figure 1 (step 2) gives an example: a walk $\mathbf{w}_1 \equiv \{1 \rightarrow b \rightarrow 3 \rightarrow e \rightarrow 6 \rightarrow g \rightarrow 4 \rightarrow c \rightarrow 2 \rightarrow a\}$ is generated by random walk in U-I-Net. The surrounding context of each node in \mathbf{w}_1 implies multiple correlations among users and items within \mathbf{w}_1 . Take sub-sequences $\hat{\mathbf{w}}_b \equiv \{b \rightarrow 3 \rightarrow e \rightarrow 6 \rightarrow g \rightarrow 4 \rightarrow c\}$ and $\hat{\mathbf{w}}_3 \equiv \{3 \rightarrow e \rightarrow 6 \rightarrow g \rightarrow 4 \rightarrow c \rightarrow 2\}$ (window size equals to 3) centered at user 6 and item g as examples. The multiple correlations include:

- user-item direct correlation: user 6 shows preference on item e and g since 6 directly connects to e and g in $\hat{\mathbf{w}}_b$.
- user-user indirect correlation: user 6 has similar taste with user 3 and user 4 in $\hat{\mathbf{w}}_b$ since both 6 and 3 rate item e positively, both 6 and 4 rate item g positively.
- user-item indirect correlation: user 6 has indirect preference on item b and item c in $\hat{\mathbf{w}}_b$ since user 3 and user 4 have direct correlations with b and c respectively, and 3 and 4 have indirect correlations with 6.
- item-user direct correlation: item g receives positive feedback from both user 6 and user 4 since g directly connects to 6 and 4 in $\hat{\mathbf{w}}_3$.
- item-item indirect correlation: item g has similar attribute with item e and item c in $\hat{\mathbf{w}}_3$ since both g and e are rated by user 6 positively, both g and c are rated by user 4 positively.
- item-user indirect correlation: item g receives indirect preference from user 3 and user 2 in $\hat{\mathbf{w}}_3$ since item e and item c have direct correlations with 3 and 2 respectively, and e and c have indirect correlations with g .

According to the above discussion, we can find that each random walk contains both direct correlations between neighboring nodes and transitive correlations among non-neighboring nodes. In addition, the correlations exist not only between two nodes of the same type but also two nodes of different types, which makes the extracted information comprehensive. Moreover, the co-occurrence frequencies of any two nodes in the same walks reflect their correlations if

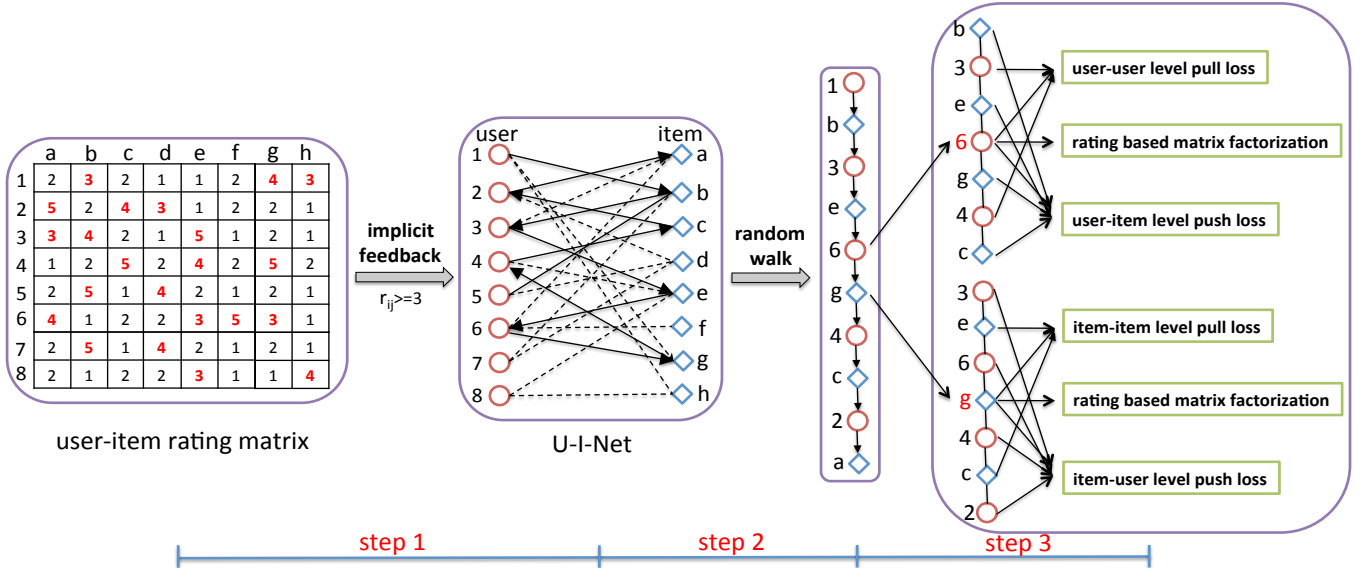


Fig. 1. The illustration of ImWalkMF model. The first step is to construct user-item implicit bipartite network (U-I-Net) via extracting users' positive feedbacks on items. The second step is to generate random walk sequences (e.g., w_1 labeled by solid lines with arrow) by random walk sampling on U-I-Net. The third step is to design the joint model as well as the learning algorithm.

we generate plenty of walks rooted at each node. In the next subsection, we design a joint model to formalize these implicit correlations as well as the explicit ratings information.

C. Joint Model of Matrix Factorization and Implicit Walk Integrative Learning

The previous two steps generate a set of random walk sequences that capture multiple reliable correlations among users and items. As the last and the core step of ImWalkMF, we design a joint model to formalize these correlations in this section. The high level idea of ImWalkMF model is as follows: we use matrix factorization (MF) to model the observable user-item ratings. In addition, we introduce an implicit walk integrative learning (IWIL) component to model multiple direct/indirect correlations among users and items by following two assumptions: (1) the distance of latent factors between indirect correlated neighbors (i.e. user-user pair and item-item pair) should be as small as possible since they share similar preferences or attributes; and (2) the preference score of direct/indirect correlated user-item (item-user) pair should be larger than the corresponding value of negative user-item (item-user) pair since users are likely to choose correlated items than negative items. Next, we formulate the IWIL in details.

1) *User-Level Integrative Learning:* Denote the set of user and the set of item as U and V , respectively. As discussed in Section II, user u_i and item v_j are represented by a user latent factor vector $z_i \in \mathbb{R}^{d \times 1}$ and an item latent factor vector $q_j \in \mathbb{R}^{d \times 1}$, respectively. Inspired by SoReg [12], we measure the distance between user u_i and user $u_{i'}$ by:

$$d_u(i, i') = \|z_i - z_{i'}\|^2 \quad (1)$$

The above equation obeys assumption (1), namely a user should be closer to neighbors who have similar preference than the others. Thus we define a user-level pull loss in each random walk w as:

$$\mathcal{L}_{pull}^u(w) = \sum_{\substack{u_{h(t)} \in w \\ u_{h(t)} \in U}} \sum_{\substack{-\tau \leq \Delta t \leq \tau, \Delta t \neq 0 \\ u_{h(t+\Delta t)} \in U}} \|z_{h(t)} - z_{h(t+\Delta t)}\|^2 \quad (2)$$

where τ is the window size of $u_{h(t)}$'s surrounding context $w[t - \tau : t + \tau]$, and $h(\cdot)$ maps the position of surrounding context to the node index. Thus constraining \mathcal{L}_{pull}^u keeps user-user indirect correlations. In addition, inspired by the metric learning work [27], we introduce the hinge loss to capture the difference of u_i 's preference scores on an indirect correlated item v_k and a negative item $v_{k'}$ (rating is smaller than the threshold σ in implicit feedbacks extraction):

$$H_{uv}(i, k, k') = \left\{ \xi + r_{ik'} - z_i q_k \right\}_+ \quad (3)$$

where $\{x\}_+ = \max(x, 0)$ and ξ is a positive margin value. The hinge loss obeys assumption (2) and a loss penalty will incur if the preference score (quantified by the multiplication of two related latent factor vectors) of positive pair (u_i, v_k) is not at least ξ larger than the corresponding value (explicit rating) of negative pair $(u_i, v_{k'})$. Denote the set of negative items which receive negative indications from u_i as $N(u_i)$ and a user-level push loss in each random walk w is formally defined as:

$$\mathcal{L}_{push}^u(w) = \sum_{\substack{u_{h(t)} \in w \\ u_{h(t)} \in U}} \sum_{\substack{-\tau \leq \Delta t \leq \tau, \Delta t \neq 0 \\ v_{h(t+\Delta t)} \in V}} \sum_{v_{t'} \in N(u_{h(t)})} \left\{ \xi + r_{h(t)t'} - z_{h(t)} q_{h(t+\Delta t)} \right\}_+ \quad (4)$$

Thus constraining \mathcal{L}_{push}^u is to capture user-item direct/indirect correlations.

2) *Item-Level Integrative Learning*: Similar to user-level integrative learning, the item-level pull loss is formally defined as:

$$\mathcal{L}_{pull}^v(\mathbf{w}) = \sum_{\substack{v_h(t) \in \mathbf{w} \\ v_h(t) \in V}} \sum_{\substack{-\tau \leq \Delta t \leq \tau, \Delta t \neq 0 \\ v_{h(t+\Delta t)} \in V}} \|\mathbf{q}_{h(t)} - \mathbf{q}_{h(t+\Delta t)}\|^2 \quad (5)$$

Similarly, constraining \mathcal{L}_{pull}^v keeps item-item indirect correlations. Besides, denote the set of users who give negative feedbacks to v_j as $N(v_j)$ and the item-level push loss is formulated as:

$$\mathcal{L}_{push}^v(\mathbf{w}) = \sum_{\substack{v_h(t) \in \mathbf{w} \\ v_h(t) \in V}} \sum_{\substack{-\tau \leq \Delta t \leq \tau, \Delta t \neq 0 \\ u_{t'} \in N(v_{h(t)}) \\ u_{h(t+\Delta t)} \in U}} \sum_{u_{t'} \in N(v_{h(t)})} \left\{ \xi + r_{t'h(t)} - \mathbf{z}_{h(t+\Delta t)} \mathbf{q}_{h(t)} \right\}_+ \quad (6)$$

In the same way, the loss penalty of \mathcal{L}_{push}^v captures item-user direct/indirect correlations.

3) *The Joint Objective*: Denote the set of all random walk sequences as C , the joint objective function is defined as a combination of the MF and the IWIL in C :

$$\mathcal{L} = \mathcal{L}_{MF} + \sum_{\mathbf{w} \in C} \left\{ \mathcal{L}_{pull}^u(\mathbf{w}) + \mathcal{L}_{push}^u(\mathbf{w}) + \mathcal{L}_{pull}^v(\mathbf{w}) + \mathcal{L}_{push}^v(\mathbf{w}) \right\} + \mathcal{L}_{reg} \quad (7)$$

where \mathcal{L}_{reg} is the regularization term for avoiding over-fitting. Let C_{uu} , C_{uvv} , C_{vv} and C_{vuu} be the sets of user-user pairs of \mathcal{L}_{pull}^u , user-item-item triples of \mathcal{L}_{push}^u , item-item pairs of \mathcal{L}_{pull}^v and item-user-user triples of \mathcal{L}_{push}^v in C , respectively. Thus we can rewrite the joint objective function in terms of an expectation formula:

$$\begin{aligned} \mathcal{L} = & \underbrace{\mathbb{E}_{\{r_{ij} \in \tilde{R}\}} (r_{ij} - \mathbf{z}_i^T \mathbf{q}_j)^2}_{\mathcal{L}_{MF}} + \underbrace{\lambda (\|\mathbf{z}\|^2 + \|\mathbf{q}\|^2)}_{\mathcal{L}_{reg}} \\ & + \underbrace{\mathbb{E}_{\{(a, a') \in C_{uu}\}} \|\mathbf{z}_a - \mathbf{z}_{a'}\|^2}_{\mathcal{L}_{pull}^u} \\ & + \underbrace{\mathbb{E}_{\{(l, k, k') \in C_{uvv}\}} \left\{ \xi + r_{lk'} - \mathbf{z}_l \mathbf{q}_k \right\}_+}_{\mathcal{L}_{push}^u} \\ & + \underbrace{\mathbb{E}_{\{(b, b') \in C_{vv}\}} \|\mathbf{q}_b - \mathbf{q}_{b'}\|^2}_{\mathcal{L}_{pull}^v} \\ & + \underbrace{\mathbb{E}_{\{(g, h, h') \in C_{vuu}\}} \left\{ \xi + r_{h'g} - \mathbf{z}_h \mathbf{q}_g \right\}_+}_{\mathcal{L}_{push}^v} \end{aligned} \quad (8)$$

where \tilde{R} is the set of observable ratings and λ is the regularization parameter. To minimize the joint objective function, we utilize the stochastic gradient descent to update model parameters. The main challenge is: we have two different components MF and IWIL that learn from two different data

sources. To resolve this issue, we design a combined strategy based on sampling to learn from random walk sequences. Specifically, for each node (user/item) in a walk, it first draws samples from ratings data and updates parameters of the MF component. Then it collects correlated neighbors sets of this node and draws pair/triple samples for each loss function of IWIL component and further updates the corresponding parameters. In order to reduce the training epochs and sampling overhead, each component will be trained on a mini-batch of data samples instead of on a single sample. The detail of learning algorithm is summarized in Algorithm 1.

D. Model Inference

The stochastic gradient descent is applied to update model parameters for both MF and IWIL components in the joint objective function, i.e., Eq. (8). The update rules for different training samples are:

- For rating sample $r_{ij} \in \tilde{R}$, the gradients of \mathbf{z}_i and \mathbf{q}_j are calculated as follows:

$$\begin{aligned} \Delta_{\mathbf{z}_i} &= -2 \cdot (r_{ij} - \mathbf{z}_i^T \mathbf{q}_j) \mathbf{q}_j + 2 \cdot \lambda \mathbf{z}_i \\ \Delta_{\mathbf{q}_j} &= -2 \cdot (r_{ij} - \mathbf{z}_i^T \mathbf{q}_j) \mathbf{z}_i + 2 \cdot \lambda \mathbf{q}_j \end{aligned} \quad (9)$$

- For user-user pair sample $(a, a') \in C_{uu}$, the gradients of \mathbf{z}_a and $\mathbf{z}_{a'}$ are calculated as follows:

$$\begin{aligned} \Delta_{\mathbf{z}_a} &= 2 \cdot (\mathbf{z}_a - \mathbf{z}_{a'}) + 2 \cdot \lambda \mathbf{z}_a \\ \Delta_{\mathbf{z}_{a'}} &= -2 \cdot (\mathbf{z}_a - \mathbf{z}_{a'}) + 2 \cdot \lambda \mathbf{z}_{a'} \end{aligned} \quad (10)$$

- For user-item-item triple sample $(l, k, k') \in C_{uvv}$, the gradients of \mathbf{z}_l and \mathbf{q}_k are formulated as follows:

$$\begin{aligned} \Delta_{\mathbf{z}_l} &= -\delta(\xi + r_{lk'} - \mathbf{z}_l \mathbf{q}_k) \cdot \mathbf{q}_k + 2 \cdot \lambda \mathbf{z}_l \\ \Delta_{\mathbf{q}_k} &= -\delta(\xi + r_{lk'} - \mathbf{z}_l \mathbf{q}_k) \cdot \mathbf{z}_l + 2 \cdot \lambda \mathbf{q}_k \end{aligned} \quad (11)$$

where $\delta(x)$ is an indicator function, which equals to 1 if and only if x is larger than 0 otherwise equals to 0.

- For item-item pair sample $(b, b') \in C_{vv}$, the gradients of \mathbf{q}_b and $\mathbf{q}_{b'}$ are calculated as follows:

$$\begin{aligned} \Delta_{\mathbf{q}_b} &= 2 \cdot (\mathbf{q}_b - \mathbf{q}_{b'}) + 2 \cdot \lambda \mathbf{q}_b \\ \Delta_{\mathbf{q}_{b'}} &= -2 \cdot (\mathbf{q}_b - \mathbf{q}_{b'}) + 2 \cdot \lambda \mathbf{q}_{b'} \end{aligned} \quad (12)$$

- For item-user-user triple sample $(g, h, h') \in C_{vuu}$, the gradients of \mathbf{z}_h and \mathbf{q}_g are formulated as follows:

$$\begin{aligned} \Delta_{\mathbf{z}_h} &= -\delta(\xi + r_{h'g} - \mathbf{z}_h \mathbf{q}_g) \cdot \mathbf{q}_g + 2 \cdot \lambda \mathbf{z}_h \\ \Delta_{\mathbf{q}_g} &= -\delta(\xi + r_{h'g} - \mathbf{z}_h \mathbf{q}_g) \cdot \mathbf{z}_h + 2 \cdot \lambda \mathbf{q}_g \end{aligned} \quad (13)$$

With the above gradient formulas, we repeat the process of Algorithm 1 to train the model until the parameter updates meet the convergence criterion.

IV. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the performances of our model as well as the baselines.

Algorithm 1: ImWalkMF

Data:

1 \tilde{R} , U-I-Net

Result:

2 $[z, q]$

3 Initialize $z \in \mathbb{R}^{d \times |U|}$, $q \in \mathbb{R}^{|V| \times d}$ with random values within $[0,1]$

4 **for each** $u^* \in U$ **do**

5 **for** $t = 0$ **to** T **do**

6 $w \leftarrow \text{RandomWalkSampling}(\text{U-I-Net}, u^*, L)$

6 $[z, q] \leftarrow \text{Inference}([z, q], \tilde{R}, w, \tau)$

7 Return $[z, q]$

8

9 $\text{RandomWalkSampling}(\text{U-I-Net}, u^*, L)$

10 Initialize walk w with $[u^*]$

11 currNode = u^*

12 **for** $l = 1$ **to** L **do**

13 nextNode = randomSelect(currNode.neighbor)

14 $w.add(\text{nextNode})$

15 currNode = nextNode

16 return w

18 $\text{Inference}([z, q], \tilde{R}, w, \tau)$

19 // user-level

20 **for each** $u_i \in w$ **do**

21 initialize C_{uu}^{w, u_i} and C_{uv}^{w, u_i} with empty sets

22 // MF component

23 sample a mini-batch of rating pair (i, f) from $\tilde{R}[i, :]$

24 update z_i, q_f via Eq. (9)

25 // IWIL component

26 **for each** $v_k \in w[i - \tau : i + \tau]$ **do**

27 **for each** $v_{k'} \in N(u_i)$ **do**

28 add triple (i, k, k') to C_{uv}^{w, u_i}

29 **for each** $u_{i'} \in w[i - \tau : i + \tau]$ **do**

30 add pair (i, i') to C_{uu}^{w, u_i}

31 sample a mini-batch of triple (i, k, k') from C_{uv}^{w, u_i}

32 update z_i, q_k via Eq. (11)

33 sample a mini-batch of triple (i, i') from C_{uu}^{w, u_i}

34 update $z_i, z_{i'}$ via Eq. (10)

35 // item-level

36 **for each** $v_j \in w$ **do**

37 initialize C_{vv}^{w, v_j} and C_{vu}^{w, v_j} with empty sets

38 sample a mini-batch of rating pair (g, j) from $\tilde{R}[:, j]$

39 update z_g, q_j via Eq. (9)

40 **for each** $u_h \in w[j - \tau : j + \tau]$ **do**

41 **for each** $u_{h'} \in N(v_j)$ **do**

42 add triple (j, h, h') to C_{vu}^{w, v_j}

43 **for each** $v_{j'} \in w[j - \tau : j + \tau]$ **do**

44 add pair (j, j') to C_{vv}^{w, v_j}

45 sample a mini-batch of triple (j, h, h') from C_{vu}^{w, v_j}

46 update z_h, q_j via Eq. (13)

47 sample a mini-batch of triple (j, j') from C_{vv}^{w, v_j}

48 update $q_j, q_{j'}$ via Eq. (12)

49 Return $[z, q]$

TABLE I
STATISTICS OF DATASETS USED IN THIS PAPER.

Statistics	CiaoDVD	Epinions
users #	17,615	40,163
items #	16,121	139,738
ratings #	72,665	664,824
rating density	0.026%	0.012%
rating range	[1, 5]	[1, 5]
trusts #	111,781	487,183
trust density	0.036%	0.030%

A. Experimental Design

1) *Datasets*: We use two real-world datasets CiaoDVD and Epinions. Both of them contain sparse rating feedbacks and sparse social trusts information. The trust connection is unweighted and each trust value equals to 1. Note that the proposed ImWalkMF uses only the rating feedbacks, not the social trusts information that will be considered in other baseline methods for comparison. The main statistics of the datasets are summarized in Table I.

2) *Evaluation Metric*: We use two most popular metrics, i.e., the mean absolute error (MAE) and the root mean square error (RMSE), to evaluate the recommendation accuracy of each method. MAE is defined as:

$$MAE = \frac{\sum_{r_{ij} \in \hat{R}} |r_{ij} - \hat{r}_{ij}|}{|\hat{R}|} \quad (14)$$

where r_{ij} is the rating user u_i gave to item v_j in test data set \hat{R} , and \hat{r}_{ij} represents the corresponding prediction score by a specific method. RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{r_{ij} \in \hat{R}} (r_{ij} - \hat{r}_{ij})^2}{|\hat{R}|}} \quad (15)$$

A smaller MAE or RMSE value means a better performance.

3) *Baselines*: To validate the effectiveness of our method, we select seven competitive comparison methods. The proposed ImWalkMF is under the framework of matrix factorization. Thus most of the baselines are based on matrix factorization for the fair comparison. The baselines include:

- **RMF [6]**: this is the matrix factorization approach with L2 regularization on latent factors.
- **PMF [8]**: this is the probabilistic matrix factorization algorithm which models latent factors of users and items via Gaussian distributions.
- **SocReg [12]**: this model incorporates the explicit social connections among users into matrix factorization as a regularization constraint.
- **SocRec [9]**: this is the social recommendation approach which models user preference by considering not only user's rating behavior, but also user's social connections, based on the probabilistic matrix factorization.

- **ISMF [17]**: this is the implicit social recommendation algorithm which extends the matrix factorization with implicit correlations computed by the predefined rating based similarity function, e.g., Pearson correlation coefficient.
- **GPLSA [44]**: this method utilizes the Gaussian probabilistic latent semantic analysis to model the users’ preferences on different items.
- **CUNEMF [18]**: this model extends the matrix factorization with implicit constraint generated by random walk based network embedding technique.

4) *Experimental Settings*: We randomly select 60% or 80% of the dataset as a training set to train the model, and further predict the remaining 40% or 20% of the dataset. To achieve a fair comparison, we use grid search to find the best values for the dimension of latent factor vector from {5, 10, 15, 20}, and the regularization parameter from {0.1, 0.01, 0.001, 0.0001} for each evaluated method. The latent factor vector dimension d of RMF, PMF, SocReg, SocRec, ISMF and CUNEMF is set as 20, 20, 10, 20, 20, 20, respectively. Besides, the regularization parameter λ of these six models equals to 0.1, 0.001, 0.1, 0.001, 0.01, 0.1, respectively. For GPLSA, we similarly use grid search to find the best value for the number of topic from {5, 10, 15, 20} and set the value to 10. For ImWalkMF, $\sigma = 3$ (in scale 5) is used to construct user-item implicit bipartite network, then we select $L = 40$, $T = 10$ to generate the set of random walk sequences. In model training, we set $d = 15$, $\lambda = 0.001$, $\xi = 1.0$ for both datasets, and $\tau = 6$, 4 for CiaoDVD and Epinions, respectively. Due to the sparsity of datasets, we use global average rating (mean value of all ratings in training data) to predict the test instances whose user or item never appears in training process. Furthermore, we set up two categories of test data: category *A* contains all test samples, and category *B* contains all test instances whose user and item appear in model training.

B. Result Comparison

The performances of our method and the baselines on all kinds of settings are reported in Table II, where the best performances are shown by bold text. In addition, the last row of each dataset reports the average prediction accuracy of each method. The main takeaways from the result table are summarized as follows:

- All models perform better on CiaoDVD dataset than on Epinions dataset. It is mainly because Epinions has sparser rating density than CiaoDVD, i.e. 0.012% v.s. 0.026%. In addition, SocReg and SocRec, ISMF and CUNEMF have better performances than RMF, which indicates the positive influence of incorporating the explicit social relationship or implicit correlations into MF.
- The proposed ImWalkMF has the best performance. It is much better than RMF and PMF, with 5.6% and 10.8% average improvements, respectively. In addition, ImWalkMF reduces 4.6% and 4.7% prediction errors of explicit social recommendation algorithms SocReg and SocRec, respectively. Moreover, ImWalkMF outperforms the implicit social recommendation models ISMF and CUNEMF by

on average 4.6% and 1.9% respectively, showing that the implicit random walk sequences capture informative and useful information and the joint model effectively incorporates the comprehensive correlations among users and items to further improve the recommendation accuracy. Although the improvements of ImWalkMF over some baselines are small, the small improvements in MAE or RMSE may lead to significant difference of recommendation in real applications [45].

- Comparing with the best baseline, we find that the improvements of ImWalkMF on category *B* test data is larger than the corresponding value on category *A* test data. That is because test instances (whose user or item never appears in training process) predicted by global average rating reduce the overall prediction accuracy in category *A*, which further validates the effectiveness of model training of ImWalkMF. In the further analysis of next subsection, we choose category *B* as the primary test data.

C. Analysis and Discussion

1) *Performances of Variant Proposed Models*: As described in Section III, ImWalkMF is a joint model of matrix factorization (MF) and implicit walk integrative learning (IWIL) from both user and item levels. Whether each component plays a role on the joint model and to what extent does each component influence the recommendation performance? To answer these questions, we conduct experiments to evaluate the performances of three kinds of ImWalkMF model including:

- U_{MF} : the model of MF from only user level, i.e., implementing line 20 to line 24 in the inference function of Algorithm 1.
- U_{MF}^{IWIL} : the joint model of MF and IWIL from only user level, i.e., implementing line 20 to line 34 in the inference function of Algorithm 1.
- UV_{MF}^{IWIL} : the proposed joint model of MF and IWIL from both user and item levels.

The performances of three models of on category *B* test data of various settings are reported by Table III. As it can be seen from the table:

- U_{MF} has close performance to RMF since it only relies on ratings based matrix factorization.
- U_{MF}^{IWIL} largely improves U_{MF} ’s performance. It verifies that random walk sequences capture multiple useful correlations among users and items, and the IWIL component is necessary for model learning.
- The joint model UV_{MF}^{IWIL} outperforms U_{MF}^{IWIL} , especially for the prediction accuracy in terms of RMSE. It shows that the item-level correlations, especially for item-item indirected correlation, further enhance the model’s ability in shaping latent spaces of users and items.

Therefore, each component of the joint model helps improve the performance of recommendation.

2) *Parameters Sensitivity*: The hyper-parameters play important roles in ImWalkMF, as they determine how the model

TABLE II

PERFORMANCE COMPARISON OF IMWALKMF AND THE BASELINES. THE BEST PERFORMANCES ARE SHOWN IN BOLD TEXT. CATEGORY A SET CONTAINS ALL TEST RATING SAMPLES AND CATEGORY B SET CONTAINS ALL TEST RATING SAMPLES WHOSE USER AND ITEM APPEAR IN MODEL TRAINING. IMWALKMF HAS THE BEST PERFORMANCE IN ALL SETTINGS.

Dataset	Train%	Test Category	Metric	RMF	PMF	SocReg	SocRec	ISMF	GPLSA	CUNEMF	ImWalkMF	
CiaoDVD	60%	A	MAE	0.806	0.868	0.804	0.789	0.801	0.763	0.769	0.754	
			RMSE	1.039	1.131	1.035	1.027	1.031	1.026	1.001	0.989	
		B	MAE	0.796	0.869	0.788	0.784	0.786	0.733	0.741	0.719	
			RMSE	1.029	1.141	1.016	1.022	1.025	1.005	0.970	0.953	
	80%	A	MAE	0.791	0.827	0.783	0.769	0.776	0.744	0.754	0.737	
			RMSE	1.019	1.084	1.013	1.001	1.011	1.003	0.983	0.971	
		B	MAE	0.771	0.832	0.764	0.765	0.762	0.715	0.727	0.706	
			RMSE	0.997	1.097	0.994	0.999	0.995	0.980	0.953	0.935	
	Average over all				0.906	0.981	0.900	0.894	0.898	0.871	0.862	0.845
	Epinions	60%	A	MAE	0.864	0.887	0.847	0.841	0.836	0.821	0.830	0.814
RMSE				1.109	1.154	1.095	1.109	1.097	1.099	1.082	1.070	
B			MAE	0.863	0.901	0.846	0.840	0.835	0.818	0.828	0.809	
			RMSE	1.106	1.164	1.090	1.108	1.093	1.096	1.074	1.061	
80%		A	MAE	0.849	0.869	0.838	0.834	0.836	0.812	0.825	0.806	
			RMSE	1.095	1.128	1.083	1.098	1.095	1.089	1.076	1.062	
		B	MAE	0.850	0.873	0.837	0.835	0.838	0.809	0.823	0.801	
			RMSE	1.091	1.131	1.076	1.099	1.094	1.084	1.068	1.051	
Average over all				0.978	1.013	0.964	0.971	0.966	0.954	0.951	0.934	

TABLE III

PERFORMANCE COMPARISON OF THREE KINDS OF IMWALKMF MODEL. U_{MF}^{IWIL} LARGELY IMPROVES U_{MF} AND UV_{MF}^{IWIL} FURTHER ENHANCES U_{MF}^{IWIL} .

Dataset	Train%	Metric	U_{MF}	U_{MF}^{IWIL}	UV_{MF}^{IWIL}
CiaoDVD	60%	MAE	0.794	0.723	0.719
		RMSE	1.035	0.962	0.953
	80%	MAE	0.774	0.708	0.706
		RMSE	1.006	0.942	0.935
Epinions	60%	MAE	0.869	0.812	0.809
		RMSE	1.115	1.067	1.061
	80%	MAE	0.854	0.804	0.801
		RMSE	1.108	1.056	1.051

will be trained. In order to evaluate how changes to the hyper-parameters of ImWalkMF affect its performance on recommendation, we conduct experiments to evaluate the impacts of window size τ , latent factor vector dimension d , regularization parameter λ and hinge loss margin ξ . The recommendation accuracy (in terms of MAE and RMSE on category B test set) of ImWalkMF on various settings of τ , d , λ and ξ are shown in Figure 2. According to the figure:

- With the increment of window size τ , MAE (RMSE) decreases at first since a larger window means more useful implicit correlations among users and items. But when τ goes beyond a certain value, MAE (RMSE) increases with the further increment of τ due to the possible involvement of non-correlated information. The best values of τ for CiaoDVD and Epinions are 6 and 4, respectively.
- Similar to τ , appropriate values should be set for latent factor vector dimension d and regularization parameter λ such that the best embeddings of users (items) are learned for making accurate recommendations. The optimal d and

λ equal to 15 and 0.001, respectively.

- The margin ξ controls user’s preference gap between positive correlated items and negative items. The larger ξ is, the less penalty of hinge loss for negative samples will be incurred. Thus the performance, especially for accuracy in RMSE, decreases with the increment of ξ and the value of ξ for optimal performance equals to 1.0.

Therefore, the certain setting of the hyper-parameters lead to the best performance of ImWalkMF.

3) *Performances on Cold Start Users:* The cold start problem is prevalent in recommender systems, especially for new users or new items with limited available ratings. Thus it would be more difficult to learn preferences of cold start users than general users. In order to validate the effectiveness of ImWalkMF for improving the recommendation for cold start users, we evaluate the improvements of ImWalkMF over RMF (in terms of the reduction in MAE and RMSE) on different user groups. First we classify all users into 6 groups (i.e., 1~5, 6~10, 11~15, 16~20, 21~25 and >25) based on the number of observable ratings they have in training data, and then measure the improvement in each group. The results evaluated on category B test data are reported in Figure 3. It is easy to find that ImWalkMF consistently outperforms RMF in all user groups. More importantly, our method achieves relatively larger improvements in cold start user groups (the first few groups in the plots) as opposed to the other groups. It indicates that ImWalkMF well addresses the cold start problem. Besides, the reduction in MAE and RMSE of 60% training data is larger than that of 80% training data, which further verifies the strength of ImWalkMF in modeling cold start users’ preferences since less training data means sparser available data and more cold start users.

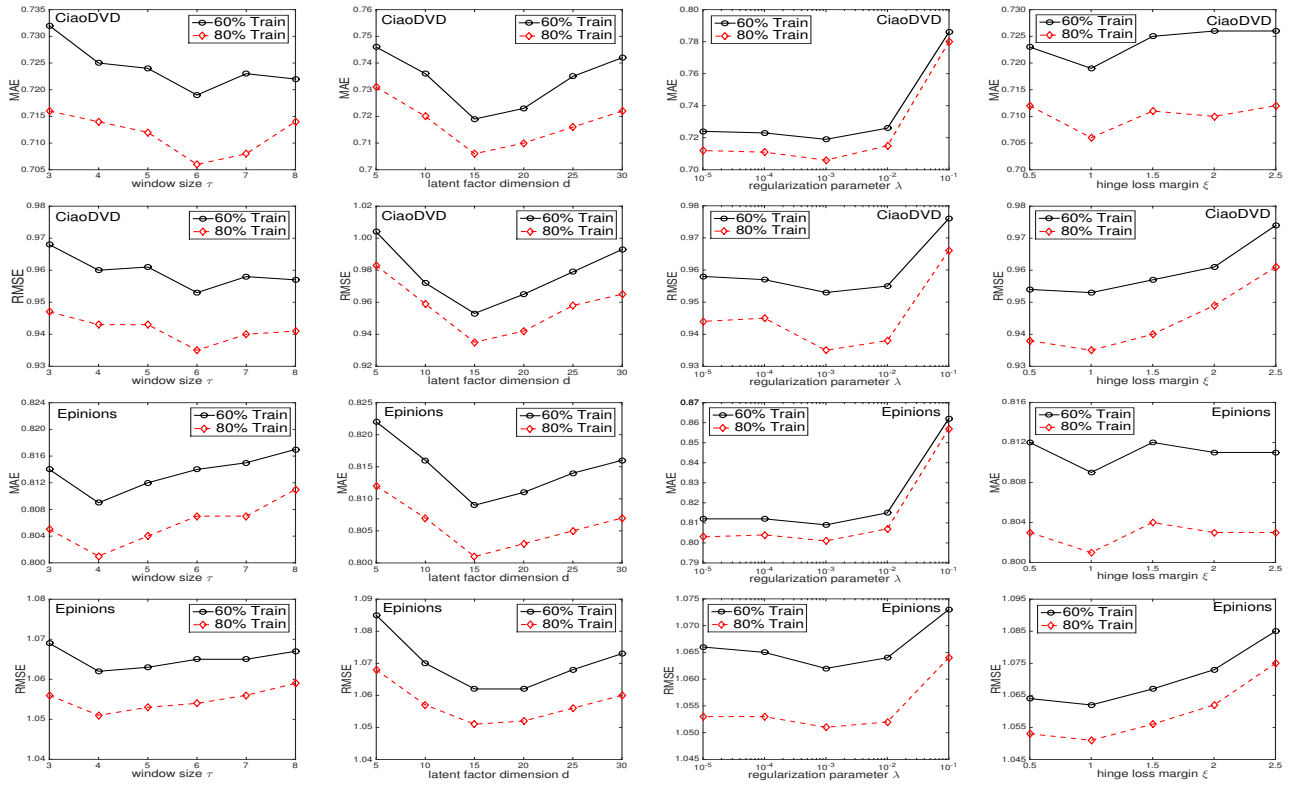


Fig. 2. The impacts of different hyper-parameters on the performance of ImWalkMF, window size τ , latent factor vector dimension d , regularization parameter λ and hinge loss margin ξ .

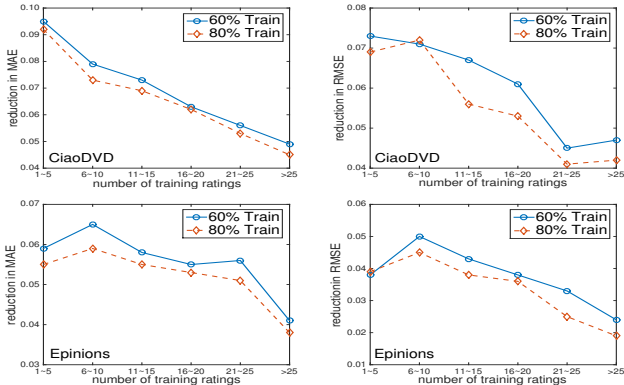


Fig. 3. The improvements (in terms of reduction in MAE and RMSE) of ImWalkMF over RMF in different users groups. ImWalkMF reaches larger improvements in the first few (cold start) groups of users than the other groups.

V. CONCLUSION AND FUTURE WORK

In this paper, in order to address the previous work’s limitations in utilizing explicit/implicit social information to improve the matrix factorization recommendation technique, first we utilize random walk sampling on user-item implicit bipartite network to collect a set of node sequences which contain multiple directed/indirected correlations among users and items. Then we design a joint model of matrix fac-

torization and implicit walk integrative learning, based on the random walk sequences. Finally we propose a sampling based combined strategy to train the joint model and learn the latent factors of users and items. The experimental results on the two real-world datasets show that our proposed method outperforms the traditional regularized/probabilistic matrix factorization models as well as the other baselines which employ explicit/implicit social information. Overall, our work provides new insights for capturing and modeling comprehensive implicit information in recommender system.

At last, some potential future work includes: (1) extending the implicit walk integrative learning with more content like items’ tags information; (2) designing heterogenous random walk based integrative learning framework for other applications like citation paper recommendation in academic search systems.

ACKNOWLEDGMENT

This work is supported by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 and the National Science Foundation (NSF) grant IIS-1447795. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute

reprints for Government purposes notwithstanding any copyright notation here on. This publication is jointly supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. 2639.

REFERENCES

- [1] D. A. Davis, N. V. Chawla, N. A. Christakis, and A.-L. Barabási, “Time to care: a collaborative engine for practical disease prediction,” *DMKD*, vol. 20, no. 3, pp. 388–415, 2010.
- [2] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *NIPS*, 2001, pp. 556–562.
- [3] J. D. Rennie and N. Srebro, “Fast maximum margin matrix factorization for collaborative prediction,” in *ICML*, 2005, pp. 713–719.
- [4] Y. Koren, “Factorization meets the neighborhood: a multifaceted collaborative filtering model,” in *KDD*, 2008, pp. 426–434.
- [5] K. Yu, S. Zhu, J. Lafferty, and Y. Gong, “Fast nonparametric matrix factorization for large-scale collaborative filtering,” in *SIGIR*, 2009, pp. 211–218.
- [6] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, 2009.
- [7] Y. Koren, “Collaborative filtering with temporal dynamics,” *Communications of the ACM*, vol. 53, no. 4, pp. 89–97, 2010.
- [8] R. Salakhutdinov and A. Mnih, “Probabilistic matrix factorization,” in *NIPS*, 2007, pp. 1257–1264.
- [9] H. Ma, H. Yang, M. R. Lyu, and I. King, “Sorec: social recommendation using probabilistic matrix factorization,” in *CIKM*, 2008, pp. 931–940.
- [10] H. Ma, I. King, and M. R. Lyu, “Learning to recommend with social trust ensemble,” in *SIGIR*, 2009, pp. 203–210.
- [11] M. Jamali and M. Ester, “A matrix factorization technique with trust propagation for recommendation in social networks,” in *RecSys*, 2010, pp. 135–142.
- [12] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, “Recommender systems with social regularization,” in *WSDM*, 2011, pp. 287–296.
- [13] B. Yang, Y. Lei, D. Liu, and J. Liu, “Social collaborative filtering by trust,” in *IJCAI*, 2013, pp. 2747–2753.
- [14] A. J. Chaney, D. M. Blei, and T. Eliassi-Rad, “A probabilistic model for using social networks in personalized item recommendation,” in *RecSys*, 2015, pp. 43–50.
- [15] G. Guo, J. Zhang, and N. Yorke-Smith, “Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings,” in *AAAI*, 2015, pp. 123–129.
- [16] X. Wang, S. C. Hoi, M. Ester, J. Bu, and C. Chen, “Learning personalized preference of strong and weak ties for social recommendation,” in *WWW*, 2017, pp. 1601–1610.
- [17] H. Ma, “An experimental study on implicit social recommendation,” in *SIGIR*, 2013, pp. 73–82.
- [18] C. Zhang, L. Yu, Y. Wang, C. Shah, and X. Zhang, “Collaborative user network embedding for social recommender systems,” in *SDM*, 2017.
- [19] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: Item-to-item collaborative filtering,” *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.
- [20] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: an open architecture for collaborative filtering of netnews,” in *CSCW*, 1994, pp. 175–186.
- [21] Y. Hu, Y. Koren, and C. Volinsky, “Collaborative filtering for implicit feedback datasets,” in *ICDM*, 2008, pp. 263–272.
- [22] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, “One-class collaborative filtering,” in *ICDM*, 2008, pp. 502–511.
- [23] S. Kabbur, X. Ning, and G. Karypis, “Fism: factored item similarity models for top-n recommender systems,” in *KDD*, 2013, pp. 659–667.
- [24] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, “Fast matrix factorization for online recommendation with implicit feedback,” in *SIGIR*, 2016, pp. 549–558.
- [25] C.-X. Zhang, Z.-K. Zhang, L. Yu, C. Liu, H. Liu, and X.-Y. Yan, “Information filtering via collaborative user clustering modeling,” *Physica A*, vol. 396, pp. 195–203, 2014.
- [26] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, “Distance metric learning with application to clustering with side-information,” in *NIPS*, 2002, pp. 505–512.
- [27] K. Q. Weinberger, J. Blitzer, and L. Saul, “Distance metric learning for large margin nearest neighbor classification,” in *NIPS*, 2006.
- [28] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, “Large scale metric learning from equivalence constraints,” in *CVPR*, 2012, pp. 2288–2295.
- [29] W. Liu and I. W. Tsang, “Large margin metric learning for multi-label prediction,” in *AAAI*, 2015, pp. 2800–2806.
- [30] Z. Zhao, Q. Yang, D. Cai, X. He, and Y. Zhuang, “Expert finding for community-based question answering via ranking metric network learning,” in *IJCAI*, 2016, pp. 3000–3006.
- [31] Z. Zhao, H. Lu, V. W. Zheng, D. Cai, X. He, and Y. Zhuang, “Community-based question answering via asymmetric multi-faceted ranking network learning,” in *AAAI*, 2017.
- [32] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, “Collaborative metric learning,” in *WWW*, 2017, pp. 193–201.
- [33] T. Chen and Y. Sun, “Task-guided and path-augmented heterogeneous network embedding for author identification,” in *WSDM*, 2017.
- [34] C. Gkantsidis, M. Mihail, and A. Saberi, “Random walks in peer-to-peer networks,” in *INFOCOM*, 2004.
- [35] J. Leskovec and C. Faloutsos, “Sampling from large graphs,” in *KDD*, 2006, pp. 631–636.
- [36] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, “Walking in facebook: A case study of unbiased sampling of osns,” in *INFOCOM*, 2010.
- [37] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.” Stanford InfoLab, Tech. Rep., 1999.
- [38] M. E. Newman, “A measure of betweenness centrality based on random walks,” *Social networks*, vol. 27, no. 1, pp. 39–54, 2005.
- [39] X. Song, Y. Chi, K. Hino, and B. Tseng, “Identifying opinion leaders in the blogosphere,” in *CIKM*, 2007, pp. 971–974.
- [40] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *KDD*, 2014, pp. 701–710.
- [41] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *KDD*, 2016, pp. 855–864.
- [42] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *NIPS*, 2013, pp. 3111–3119.
- [43] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “Bpr: Bayesian personalized ranking from implicit feedback,” in *UAI*, 2009, pp. 452–461.
- [44] T. Hofmann, “Collaborative filtering via gaussian probabilistic latent semantic analysis,” in *SIGIR*, 2003, pp. 259–266.
- [45] Y. Koren, “Factor in the neighbors: Scalable and accurate collaborative filtering,” *TKDD*, vol. 4, no. 1, 2010.