Social Recommendation Across Multiple Relational Domains

Meng Jiang[†§], Peng Cui[†§], Fei Wang[‡], Qiang Yang[¶], Wenwu Zhu[†§], Shiqiang Yang[†§]

[†]Beijing Key Laboratory of Networked Multimedia
[§]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[‡]Healthcare Transformation Group, IBM T J Watson Research Center, USA
[¶]Department of Computer Science and Engineering, Hong Kong University of Science and Technology
jm06@mails.tsinghua.edu.cn, cuip@tsinghua.edu.cn, feiwang03@gmail.com
qyang@cse.ust.uk, wwzhu@tsinghua.edu.cn, yangshq@tsinghua.edu.cn

ABSTRACT

Social networks enable users to create different types of personal items. In dealing with serious information overload, the major problems of social recommendation are sparsity and cold start. In existing approaches, relational and heterogeneous domains can not be effectively utilized for social recommendation, which brings a challenge to model users and multiple types of items together on social networks. In this paper, we consider how to represent social networks with multiple relational domains and alleviate the major problems in an individual domain by transferring knowledge from other domains. We propose a novel *Hybrid Random Walk* (HRW), which can integrate multiple heterogeneous domains including directed/undirected links, signed/unsigned links and within-domain/cross-domain links into a star-structured hybrid graph with user graph at the center. We perform random walk until convergence and use the steady state distribution for recommendation. We conduct experiments on a real social network dataset and show that our method can significantly outperform existing social recommendation approaches.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; J.4 [**Computer Applications**]: Social and Behavioral Sciences

General Terms

Algorithms, Experimentation

Keywords

Social Recommendation, Relational Domains, Star-Structured Graph, Hybrid Random Walk, Transfer Learning

1. INTRODUCTION

Social networks enable users to create different types of personal items, including web posts (e.g., tweets on Twitter, articles on Facebook), videos, user labels and interest groups, which form multiple item domains. A huge volume of items from these domains diffuse via user graph and result in serious information overload. Most existing recommender systems deployed for web post re-ranking suffer from the data sparsity and cold start problems. One related solution is to characterize latent features of each user independently by regularizing vectors according to information from auxiliary domains [12, 19]. On social networks, however, multiple types of items are not solely related to the characteristic or interest of one user himself. For example, users read web posts of other people; they edit user labels similar to their friends or colleagues; they watch videos uploaded by one another. It is the user graph that these items rely on to be diffused and adopted. Another approach is to learn individuals' preferences by connecting directly-related item domains such as a track and tags on the track [11], web pages and queries to them [27]. Yet this does not apply to multiple indirectly-related item domains on social networks, eg., web posts like tweets, images [28], videos that one recommends, and user labels that label one's identity and interest. Rather, they are tightly connected with mass users who form the user graph and different domains jointly reflect users' intrinsic preferences and tie strength.

Among multiple domains on social networks, social domain, which consists of users and social relation links (including undirected friendship links in Facebook and directed follow links in Twitter), forms a weighted user graph. It plays a central role and we can treat all item domains as peripherals in a star-structured graph centered at social domain. Consequently, all items in peripheral domains are generated by users and propagated over social relations in social domain, which serves as the cross-domain links. An example of this structure is shown in Figure 1.

When conducting social recommendation, tie strength (often referred to as social influence [16, 3] or social trust [10]) on the central social domain is a very important factor. From the view of *Homophily* [1], users are more likely to have stronger ties if they share similar characteristics. Here cross-domain links to different peripheral domains reflect users' characteristics from different aspects. For example, a cross-domain link from a user in social domain to a web post in
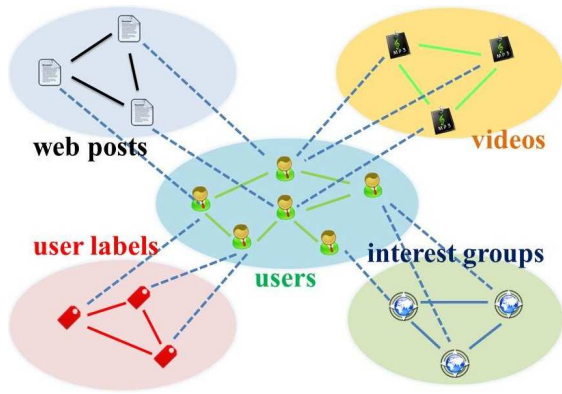
Figure 1: An example of hybrid high-order star-structured graph. 4 different item domains surround social domain.

web-post domain about Iphone shows the user's short-term interest in Iphone, while a cross-domain link to a user label *Iphone Fans* shows the user's long-term interest in Iphone. Clearly the more auxiliary domain knowledge we have, the more we know about the users and the more accurate we can estimate the user tie strength. We can hence get better recommendation performance, especially when the target domain is extremely sparse. A simple example is, when a user and his friend have many common user labels, we can update their tie strength stronger and expect that they will adopt (share the articles, movie or source links, forward or comment the tweet, etc.) similar web posts in web-post domain, and vice versa.

Hence we clearly declare that such knowledge transfer procedures among multiple relational domains should focus on updating user tie strength on social domain together. We transfer knowledge from auxiliary domains to work on a sparse target domain through the principal medium of user graph, but it still remains a difficult problem because both within and cross domain correlations are complex. How to collaboratively and deeply integrate multiple domains to discover the common knowledge and accordingly alleviate the sparsity and cold start problems in each individual domain is rather challenging, because:

(1) These domains are relational and have complex link structures. Besides cross-domain links from social domain to other peripheral domains, there are rich within-domain links in each relational domain. For example, in social domain, users are linked by social relations; in web-post domain and user-label domain, web posts and user labels are linked by semantic similarity; in interest-group domain, groups are linked by common members, etc. How to effectively utilize rich links poses a challenge to method capability.

(2) These domains are heterogeneous. Some within-domain links are directed, e.g., the follow links in social domain from Twitter-style microblogging platforms. Some of these links are undirected, for example, the semantic similarity links within the web-post domain. Some cross-domain links are signed, such as the cross-domain links to web-post domain where we know which posts are adopted and rejected by users. Some are unsigned, e.g., the cross-domain links to the user-label domain where we cannot observe and infer the rejection behavior, because people edit their own user

labels instead of choosing from recommended user labels. How to transfer knowledge across these different domains despite their information heterogeneity poses a challenge to method comprehensibility.

(3) These domains are sparse, although the sparsity of them may be different. The sparsity problem is essentially caused by the fact that we have a very large volume of users, a limited level of attention and time for each user. It is a major challenge to make the best use of limited information available despite the fact that it comes from different domains.

In order to address with the above problems, we propose a novel *Hybrid Random Walk* (HRW) method for transferring knowledge on a star-structured graph and apply it for social recommendation, which is a fundamental and practical problem that needs a reasonable solution. It estimates a proper relevance score between any two nodes within one domain or across the social domain and an item domain. Such scores can represent user tie strength between two users on the social domain, item similarity between two items on the item domain and cross-domain correlations that represent to what degree a user adopts or rejects an item. Our method integrates the knowledge from all the multiple relational domains, which alleviates sparsity and cold-start problems for social recommendation.

It is worthwhile to highlight our contribution as follows:

(1) We propose an effective method to transfer knowledge across multiple relational domains on social networks, which is capable of incorporating heterogeneous graphs with directed/undirected links, within-domain/cross-domain links and signed/unsigned links. This method can be widely and naturally applied in graph-based scenarios, such as social networks and biology networks.

(2) We use the proposed *Hybrid Random Walk* method on a star-structured graph to recommend items on social networks, conduct intensive experiments on a large real social network dataset, and demonstrate that it can greatly boost the performance. This shows a promising way to solve the sparsity problem of multiple relational domains.

(3) The experiments show the importance of user labels for providing recommendation service to inactive or even new users on social networks, who are the most fragile ones that the system should pay more attention to.

The rest of this paper is organized as follows. In Section 2 we review related works and provide necessary background for traditional recommender systems and random walk models. Then we introduce our *Hybrid Random Walk* method in Section 3. In Section 4 we evaluate this method with comparative experiments on social network data and analyze the performances. Finally, we conclude the paper in Section 5.

2. RELATED WORKS

As the problem we focus on is social recommendation, one closely related area is Collaborative Filtering (CF), which is the most popular approach for building recommender systems and has been successfully employed in many applications. CF techniques are divided into two categories: memory-based [23, 4] and model-based [17, 9]. Memory-based algorithms are heuristics to recommend by aggregating the preference of similar users. Model-based algorithms learn models to recommend based on patterns recognized in the rating behaviors of users such as clustering methods, bayesian models and matrix factorization algorithms. However, CF

methods solely rely on the historical user-item interaction behaviors. For a typical recommendation service, there are not sufficient behaviors of users and items in most cases. Thus CF is fragile to perform recommendation on sparse domains.

To alleviate the data sparsity problem, researchers have introduced methods which incorporate information from several types of items [15]. Wang et al. [27] propose a reinforcement clustering method taking into account the similarities between related domains. Transfer learning is presented as a common cross-domain collaborative method for recommender systems which can transfer knowledge from auxiliary data to remedy the sparse target matrix [2, 21, 22]. Multiple rating matrices are shared as useful knowledge in related domains through relational learning methods to alleviate the sparsity problem [24, 25]. Koren et al. [12] introduce matrix factorization techniques allowing the incorporation of additional information with his Netflix Prize approach. Another work named Matrix Co-factorization [5] incorporates rich user and item information into recommendation with implicit feedback. These approaches can not be effectively applied on social network services, because social recommendation is directly driven by the natural and complex link structures of user graph, which is not considered in traditional recommender systems.

To solve social recommendation problems, Ma et al. [18, 19] propose a method integrating social network structure and the user-item rating matrix based on probabilistic factor analysis. However, user relationship should be considered for its power on the information transmission. Leskovec et al. [14] present information cascades in the context of recommendation. Trust-aware systems [20] take trust worthiness of users as an important consideration of social relations. The intractable problem is that tie strength on the user graph is shaped by multiple domains of *User Generated Content* (UGC) instead of an individual one. Implemented with a Random Walk with Restart (RWR) method, our method incorporates relational and heterogeneous domains together with the user graph at the center.

Several studies exist in the field of applying RWR on item recommendation. Konstas et al. [11] use RWR in order to provide a more natural and efficient way to represent social networks. Random Walk with Restarts model can consider both the explicit user-provided annotations and friendships inherent in the user graph. TrustWalker [10] finds a good trade-off combining the trust-based and the CF approach for recommendation and outperforms both methods in precision. Earlier works on heterogeneous data co-clustering [6] inspire us to develop a new social network representation in which a central type of objects connects the other types to represent inter-relationships. In this paper, we consider the specific structure of multiple social relational domains, model these domains into a star-structure graph, and incorporate more complex link structures both within and across domains.

3. THE ALGORITHM

In this section we introduce details of our method on a star-structured graph. First let's start with hybrid second-order graph.

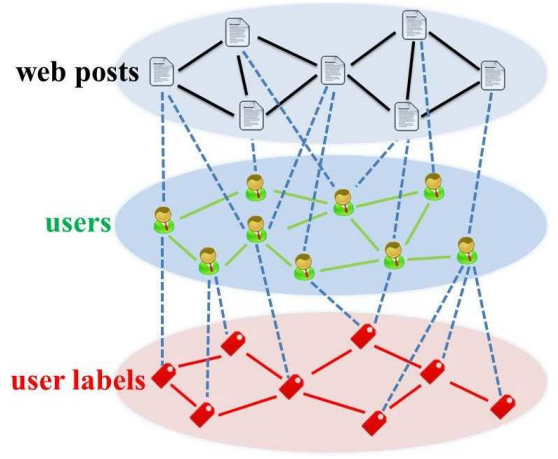### 3.1 Random Walk on Hybrid Second-Order Star-Structured Graph

Figure 2: An example of the second-order star-structured graph.

Figure 2 shows a real world example of the second-order hybrid star-structured graph, where on the microblogging system each user has his/her own user labels and can send web posts. Those web posts have semantic relationships according to their contents, and so do the user labels. Note that the graph we are interested in is different from the traditional star-structured graph [7]. The traditional graph does not consider the entity relationships within each domain, while the hybrid graph considers both within-domain and cross-domain entity relationships.

3.1.1 Notations and Symbols

Table 1 summarizes some notations that we use to derive the algorithm, and we will also use the following symbols to denote the five subgraphs contained in Figure 2.

Table 1: Notations of Our Scenario

u_i	The i-th user
$\mathcal{U} = \{u_1, u_2, \cdots, u_m\}$	The set of users
p_i	The i-th web post
$\mathcal{P} = \{p_1, p_2, \cdots, p_n\}$	The set of web posts
t_i	The i-th user label
$\mathcal{T} = \{t_1, t_2, \cdots, t_l\}$	The set of user labels

- $\mathcal{G}^{(\mathcal{U})} = \{\mathcal{U}, \mathcal{E}^{(\mathcal{U})}\}$, where $\mathcal{E}^{(\mathcal{U})}$ represents the edge set linking the nodes in $\mathcal{U}$

- $\mathcal{G}^{(\mathcal{P})} = \{\mathcal{P}, \mathcal{E}^{(\mathcal{P})}\}$, where $\mathcal{E}^{(\mathcal{P})}$ represents the edge set linking the nodes in $\mathcal{P}$

- $\mathcal{G}^{(\mathcal{T})} = \{\mathcal{T}, \mathcal{E}^{(\mathcal{T})}\}$, where $\mathcal{E}^{(\mathcal{T})}$ represents the edge set linking the nodes in $\mathcal{T}$

- $\mathcal{G}^{(\mathcal{U}\mathcal{P})} = \{\mathcal{U}\bigcup\mathcal{P}, \mathcal{E}^{(\mathcal{U}\mathcal{P})}\}$, where $\mathcal{E}^{(\mathcal{U}\mathcal{P})}$ represents the edges linking between the nodes in $\mathcal{U}$ and $\mathcal{P}$

- $\mathcal{G}^{(\mathcal{U}\mathcal{T})} = \{\mathcal{U}\bigcup\mathcal{T}, \mathcal{E}^{(\mathcal{U}\mathcal{T})}\}$, where $\mathcal{E}^{(\mathcal{U}\mathcal{T})}$ represents the edges linking between the nodes in $\mathcal{U}$ and $\mathcal{T}$

User relationships in $\mathcal{G}^{\mathcal{U}}$ can be captured with different ways in different scenarios. For online social networks

such as Facebook, such relationships can be user friendships, which are undirected. For microblogging services such as Twitter, such relationships can be reflected by user followings, which are directed. The relevance from user u_i to u_j is

$$w_{ij}^{(\mathcal{U})} = \begin{cases} 1 & \text{if user } u_i \text{ is a friend of } u_j \text{ or follows } u_j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

For relationships among web posts in $\mathcal{P}$, we construct the TF-IDF representation vector for each post as $b_i = [b_{i1}, \cdots, b_{ik}, \cdots, b_{iK}]^\top$ in matrix $\mathbf{B}$ (K is the size of vocabulary), then adopt the cosine similarity to measure the semantic similarity between post b_i and b_j as:

$$w_{ij}^{(\mathcal{P})} = \frac{\sum_k b_{ik} b_{jk}}{\sqrt{\sum_k {b_{ik}}^2} \sqrt{\sum_k {b_{jk}}^2}} \tag{2}$$

For user labels, we use co-occurrence frequency to measure their relationships. Assume that labels t_i and t_j appear in c_i and c_j tweets as a word, and co-appear in c_{ij} tweets. Then the semantic similarity between them is computed as

$$w_{ij}^{(\mathcal{T})} = \frac{c_{ij}}{c_i + c_j - c_{ij}} \tag{3}$$

Till now we have constructed three similarity matrices $\mathbf{W}^{(\mathcal{U})} = \{w_{ij}^{(\mathcal{U})}\}$, $\mathbf{W}^{(\mathcal{P})} = \{w_{ij}^{(\mathcal{P})}\}$ and $\mathbf{W}^{(\mathcal{T})} = \{w_{ij}^{(\mathcal{T})}\}$ to encode edge weights for three within-domain subgraphs. We denote their corresponding degree matrices as $\mathbf{D}^{(\mathcal{U})}$, $\mathbf{D}^{(\mathcal{P})}$ and $\mathbf{D}^{(\mathcal{T})}$ respectively.

There are also two cross-domain subgraphs $\mathcal{G}^{(\mathcal{UP})}$ and $\mathcal{G}^{(\mathcal{UT})}$ whose edge weights need to be estimated. Since web posts can be adopted or rejected but user labels are edited by users themselves (regarded as adoption), both positive and negative user-post links exist but only positive user-label links exist. These links are presented as undirected edges $e_{ij}^{(\mathcal{UP})}$ and $e_{ij}^{(\mathcal{UT})}$, whose weights are calculated as

$$
\begin{aligned}
w_{ij}^{(\mathcal{UP})^+} &= \begin{cases} 1 & \text{if user } u_i \text{ adopts post } \rho_j \\ 0 & \text{otherwise} \end{cases} \\
w_{ij}^{(\mathcal{UP})^-} &= \begin{cases} 1 & \text{if user } u_i \text{ rejects post } \rho_j \\ 0 & \text{otherwise} \end{cases} \\
w_{ij}^{(\mathcal{UT})^+} &= \begin{cases} 1 & \text{if user } u_i \text{ adopts label } t_j \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{4}
$$

Finally we obtain three weight matrices $\mathbf{W}^{(\mathcal{UP})^+} = \{w_{ij}^{(\mathcal{UP})^+}\}$, $\mathbf{W}^{(\mathcal{UP})^-} = \{w_{ij}^{(\mathcal{UP})^-}\}$, $\mathbf{W}^{(\mathcal{UT})^+} = \{w_{ij}^{(\mathcal{UT})^+}\}$ and we denote their corresponding degree matrices as $\mathbf{D}^{(\mathcal{UP})^+}$, $\mathbf{D}^{(\mathcal{UP})^-}$, and $\mathbf{D}^{(\mathcal{UT})^+}$.

3.1.2 Algorithm Derivation

In this section we derive a random walk algorithm to predict the missing links on $\mathcal{G}^{(\mathcal{UP})}$ and $\mathcal{G}^{(\mathcal{UT})}$, which include both within-domain and cross-domain random walk. For $\mathcal{G}^{(\mathcal{U})}$, $\mathcal{G}^{(\mathcal{P})}$ and $\mathcal{G}^{(\mathcal{T})}$, we consider the *Random Walk with Restart* (RWR) [26] model to derive the steady-state distributions, which indicate the intrinsic relevance among users, posts and labels. For a standard RWR-based relevance model, a random walker starts from the i-th vertex and iteratively jumps to other vertices with transition probabilities $\mathbf{p}_i = \{p_{i1}, \cdots, p_{in}\}$, with $p_{ii} = 1 - \alpha$. After reaching the steady-state, the probability of the random walker staying

at the j-th vertex corresponds to the relevance score of vertex j to i. Specifically, the transition probability matrices are computed as the row-normalized weight matrices (whose diagonal line are all zeros):

$$
\begin{aligned}
\mathbf{P}^{(\mathcal{U})} &= (\mathbf{D}^{(\mathcal{U})})^{-1} \mathbf{W}^{(\mathcal{U})} \\
\mathbf{P}^{(\mathcal{P})} &= (\mathbf{D}^{(\mathcal{P})})^{-1} \mathbf{W}^{(\mathcal{P})} \\
\mathbf{P}^{(\mathcal{T})} &= (\mathbf{D}^{(\mathcal{T})})^{-1} \mathbf{W}^{(\mathcal{T})}
\end{aligned} \tag{5}
$$

The final steady-state probability matrices can be obtained by iterating the following updates:

$$
\begin{aligned}
\mathbf{R}^{(\mathcal{U})}(t+1) &= \alpha \mathbf{P}^{(\mathcal{U})} \mathbf{R}^{(\mathcal{U})}(t) + (1-\alpha)\mathbf{I} \\
\mathbf{R}^{(\mathcal{P})}(t+1) &= \beta \mathbf{P}^{(\mathcal{P})} \mathbf{R}^{(\mathcal{P})}(t) + (1-\beta)\mathbf{I} \\
\mathbf{R}^{(\mathcal{T})}(t+1) &= \gamma \mathbf{P}^{(\mathcal{T})} \mathbf{R}^{(\mathcal{T})}(t) + (1-\gamma)\mathbf{I}
\end{aligned} \tag{6}
$$

where $\mathbf{R}^{(\mathcal{U})}(t)$, $\mathbf{R}^{(\mathcal{P})}(t)$, $\mathbf{R}^{(\mathcal{T})}(t)$ and $\mathbf{R}^{(\mathcal{U})}(t+1)$, $\mathbf{R}^{(\mathcal{P})}(t+1)$, $\mathbf{R}^{(\mathcal{T})}(t+1)$ are the state probability matrices at time t and $t + 1$, $0 \leq \alpha, \beta, \gamma \leq 1$ are the prior probabilities that the random walker will leave its current state. It can be easily shown that the above iterations will finally converge to the following steady state matrices when $t \rightarrow \infty$ [29].

$$
\begin{aligned}
\mathbf{R}^{(\mathcal{U})} &= (1-\alpha)(\mathbf{I} - \alpha \mathbf{P}^{(\mathcal{U})})^{-1} \\
\mathbf{R}^{(\mathcal{P})} &= (1-\beta)(\mathbf{I} - \beta \mathbf{P}^{(\mathcal{P})})^{-1} \\
\mathbf{R}^{(\mathcal{T})} &= (1-\gamma)(\mathbf{I} - \gamma \mathbf{P}^{(\mathcal{T})})^{-1}
\end{aligned} \tag{7}
$$

For cross-domain random walk, we compute the transition probability matrices as

$$
\begin{aligned}
\mathbf{P}^{(\mathcal{UP})^+} &= (\mathbf{D}^{(\mathcal{UP})^+})^{-1} \mathbf{W}^{(\mathcal{UP})^+} \\
\mathbf{P}^{(\mathcal{UP})^-} &= (\mathbf{D}^{(\mathcal{UP})^-})^{-1} \mathbf{W}^{(\mathcal{UP})^-} \\
\mathbf{P}^{(\mathcal{UT})^+} &= (\mathbf{D}^{(\mathcal{UT})^+})^{-1} \mathbf{W}^{(\mathcal{UT})^+}
\end{aligned} \tag{8}
$$

where elements $p_{ij}^{(\mathcal{UP})^+}$ and $p_{ij}^{(\mathcal{UP})^-}$ represent the transition probability that user u_i will/will not like post p_j, $p_{ij}^{(\mathcal{UT})^+}$ represents the transition probability that user u_i will own label t_j.

In social problems, we simultaneously learn relevance scores between each pair of users as $\mathbf{R}^{(\mathcal{U})} = \{r_{ij}^{(\mathcal{U})}\}$, which finally reflect the tie strength on realistic user graph. Element $r_{ij}^{(\mathcal{U})}$ represents the probability that a random walker jumps from user u_i to u_j. Now we consider the above transition paths and estimate the transition probabilities $p_{ij}^{(\mathcal{UP})^+}$, $p_{ij}^{(\mathcal{UP})^-}$, $p_{ij}^{(\mathcal{UT})^+}$ and $r_{ij}^{(\mathcal{U})}$ of one step random walk over $\mathcal{G}^{(\mathcal{UP})}$, $\mathcal{G}^{(\mathcal{UT})}$ and $\mathcal{G}^{(\mathcal{U})}$ as:

$$p_{ij}^{(\mathcal{UP})^+} = \delta \sum_{u_k \in \mathcal{U}} r_{ik}^{(\mathcal{U})} p_{kj}^{(\mathcal{UP})^+} + (1-\delta) \sum_{p_k \in \mathcal{P}} p_{ik}^{(\mathcal{UP})^+} r_{kj}^{(\mathcal{P})} \tag{9}$$

$$p_{ij}^{(\mathcal{UP})^-} = \delta \sum_{u_k \in \mathcal{U}} r_{ik}^{(\mathcal{U})} p_{kj}^{(\mathcal{UP})^-} + (1-\delta) \sum_{p_k \in \mathcal{P}} p_{ik}^{(\mathcal{UP})^-} r_{kj}^{(\mathcal{P})} \tag{10}$$

$$p_{ij}^{(\mathcal{UT})^+} = \eta \sum_{u_k \in \mathcal{U}} r_{ik}^{(\mathcal{U})} p_{kj}^{(\mathcal{UT})^+} + (1-\eta) \sum_{t_k \in \mathcal{T}} p_{ik}^{(\mathcal{UT})^+} r_{kj}^{(\mathcal{T})} \tag{11}$$

$$r_{ij}^{(\mathcal{U})} = \tau^{(\mathcal{P})}(\mu \sum_{p_k \in \mathcal{P}} p_{ik}^{(\mathcal{UP})^+} p_{jk}^{(\mathcal{UP})^+} + (1-\mu) \sum_{p_k \in \mathcal{P}} p_{ik}^{(\mathcal{UP})^-} p_{jk}^{(\mathcal{UP})^-})$$
$$+ \tau^{(\mathcal{T})} \sum_{t_k \in \mathcal{T}} p_{ik}^{(\mathcal{UT})^+} p_{jk}^{(\mathcal{UT})^+} + \tau^{(\mathcal{U})} \sum_{u_k \in \mathcal{U}} r_{ik}^{(\mathcal{U})} r_{kj}^{(\mathcal{U})} \tag{12}$$

where $0 \leq \delta, \eta, \mu, \tau^{(\mathcal{P})}, \tau^{(\mathcal{T})}, \tau^{(\mathcal{U})} \leq 1$ are the parameters for trading off the importance of different transition routes. Note that for the update of cross-domain transition probability matrices (Eq.(9) to Eq.(11)), we consider two types of routes shown in Figure 3. We also assume that the update of cross-domain transition probability matrices will affect the within-domain transition probability matrix of the user subgraph. The updating rule Eq.(12) considers three routes shown in Figure We can further give the matrix form of the
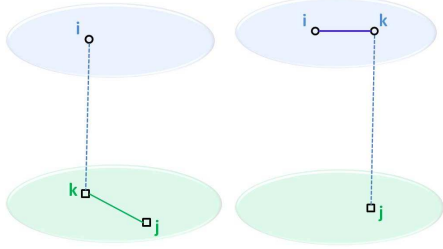
Figure 3: Transition routes we consider when updating cross-domain transition probability matrix.
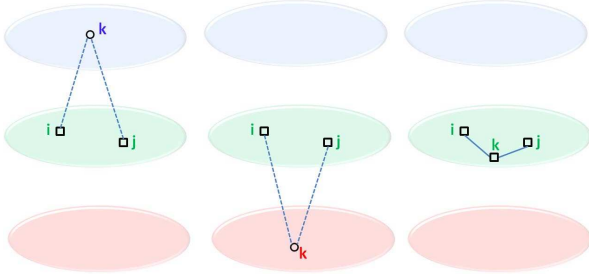
Figure 4: Transition routes we consider when updating within-domain transition probability matrix on the user subgraph.

above equation, showing the update of transition probability from time t to $t + 1$.

$$\mathbf{P}^{(\mathcal{UP})^+}(t+1) = \delta\mathbf{R}^{(\mathcal{U})}(t)\mathbf{P}^{(\mathcal{UP})^+}(t) + (1-\delta)\mathbf{P}^{(\mathcal{UP})^+}(t)\mathbf{R}^{(\mathcal{P})}$$

$$\mathbf{P}^{(\mathcal{UP})^-}(t+1) = \delta\mathbf{R}^{(\mathcal{U})}(t)\mathbf{P}^{(\mathcal{UP})^-}(t) + (1-\delta)\mathbf{P}^{(\mathcal{UP})^-}(t)\mathbf{R}^{(\mathcal{P})}$$

$$\mathbf{P}^{(\mathcal{UT})^+}(t+1) = \eta\mathbf{R}^{(\mathcal{U})}(t)\mathbf{P}^{(\mathcal{UT})^+}(t) + (1-\eta)\mathbf{P}^{(\mathcal{UT})^+}(t)\mathbf{R}^{(\mathcal{T})}$$

$$\mathbf{R}^{(\mathcal{U})}(t+1) =$$

$$\tau^{(\mathcal{P})}(\mu\mathbf{P}^{(\mathcal{UP})^+}(t)\mathbf{P}^{(\mathcal{UP})^+}(t)^T + (1-\mu)\mathbf{P}^{(\mathcal{UP})^-}(t)\mathbf{P}^{(\mathcal{UP})^-}(t)^T)$$

$$+\tau^{(\mathcal{T})}\mathbf{P}^{(\mathcal{UT})^+}(t)\mathbf{P}^{(\mathcal{UT})^+}(t)^T + \tau^{(\mathcal{U})}\mathbf{R}^{(\mathcal{U})}(t)\mathbf{R}^{(\mathcal{U})}(t)^T \quad (13)$$

With graphs $\mathcal{G}^{(\mathcal{U})}$, $\mathcal{G}^{(\mathcal{UP})}$ and $\mathcal{G}^{(\mathcal{UT})}$, the corresponding transition matrices $\mathbf{R}^{(\mathcal{U})}$, $\mathbf{P}^{(\mathcal{UP})^+}$, $\mathbf{P}^{(\mathcal{UP})^-}$, $\mathbf{P}^{(\mathcal{UT})^+}$ are computed for the next random walk step. Algorithm 1 summarizes the whole procedure of the second-order star-structured graph-based iteratively random walk method for predicting post and label adoptions. The space complexity of this algorithm is $O(m^2 + n^2 + l^2 + 2m(n+l))$. And the time complexity is $O((m^2 + 4m(n+l) + 2(n^2 + l^2))mT)$, where T is the number of iterations.

3.2 Random Walk on Hybrid High-Order Star-Structured Graph

Algorithm 1 Iterative Adoption Prediction through Random Walk over a Second-Order Star-Structured Graph

---

Require: $0 \leq \alpha, \beta, \gamma, \delta, \eta, \mu, \tau^{(\mathcal{P})}, \tau^{(\mathcal{T})}, \tau^{(\mathcal{U})} \leq 1$
1: Construct graphs $\mathcal{G}^{(\mathcal{U})}, \mathcal{G}^{(\mathcal{P})}, \mathcal{G}^{(\mathcal{T})}, \mathcal{G}^{(\mathcal{UP})}, \mathcal{G}^{(\mathcal{UT})}$
2: Compute transition probabilities $\mathbf{P}^{(\mathcal{U})}$, $\mathbf{P}^{(\mathcal{P})}$ and $\mathbf{P}^{(\mathcal{T})}$ using Eq.(5)
3: Derive steady-state distributions $\mathbf{R}^{(\mathcal{U})}$, $\mathbf{R}^{(\mathcal{P})}$ and $\mathbf{R}^{(\mathcal{T})}$ using Eq.(6)
4: Initialize the transition probability matrices $\mathbf{P}^{(\mathcal{UP})^+}(0)$, $\mathbf{P}^{(\mathcal{UP})^-}(0)$ and $\mathbf{P}^{(\mathcal{UT})^+}(0)$ using Eq.(8).
5: **for** t=1:T **do**
6:   Compute the user state distributions $\mathbf{R}^{(\mathcal{U})}(t)$ and transition probability matrices $\mathbf{P}^{(\mathcal{UP})^+}(t)$, $\mathbf{P}^{(\mathcal{UP})^-}(t)$ and $\mathbf{P}^{(\mathcal{UT})^+}(t)$ using Eq.(13)
7: **end for**
8: **Output:** The final transition probability matrices $\mathbf{R}^{(\mathcal{U})}$, $\mathbf{P}^{(\mathcal{UP})^+}$, $\mathbf{P}^{(\mathcal{UP})^-}$, and $\mathbf{P}^{(\mathcal{UT})^+}$.

In Section 3.1, we assume that there are two types of item domains, web posts and user labels, associated with each user. However, online social network is such an unprecedented comprehensive platform where there are a number of different types of UGC, e.g., posts, labels, musics and movies. In this case, the second-order graph is not enough to describe all the contents. Figure 1 shows a typical example of the hybrid high-order star-structured graph when there are four different types of UGC. We need a method to predict user behaviors on various item domains. Actually the random walk strategy we have introduced in the previous section can be easily extended to high-order cases. For notational convenience, we first introduce some notations in Table 2. We further use the following symbols to represent the different subgraphs contained in the high-order hybrid graph.

Table 2: Notations of Our Method

u_i	The i-th user
$\mathcal{U} = \{u_1, u_2, \cdots, u_m\}$	The set of users
d_{ij}	The j-th item in i-th domain
$\mathcal{D}_i = \{d_{i1}, d_{i2}, \cdots, d_{i|\mathcal{D}_i|}\}$	The set of items in i-th domain
$\mathcal{D} = \{D_1, D_2, \cdots, D_N\}$	The set of item domains

- $\mathcal{G}^{(\mathcal{U})} = \{\mathcal{U}, \mathcal{E}^{(\mathcal{U})}\}$, where $\mathcal{E}^{(\mathcal{U})}$ represents the edge set linking the nodes in $\mathcal{U}$

- $\mathcal{G}^{(\mathcal{D}_i)} = \{\mathcal{D}_i, \mathcal{E}^{(\mathcal{D}_i)}\}$, where $\mathcal{E}^{(\mathcal{D}_i)}$ represents the edge set linking the nodes in $\mathcal{D}_i$, $i = 1, \cdots, N$

- $\mathcal{G}^{(\mathcal{UD}_i)} = \{\mathcal{U} \bigcup \mathcal{D}_i, \mathcal{E}^{(\mathcal{UD}_i)}\}$, where $\mathcal{E}^{(\mathcal{UD}_i)}$ represents the edges linking the nodes in $\mathcal{U}$ and $\mathcal{D}_i$, $i = 1, \cdots, N$

With the individual domain knowledge on $\mathcal{G}^{(\mathcal{U})}$ and $\{\mathcal{G}^{(\mathcal{D}_i)}\}_{i=1}^N$, we construct their corresponding edge weight matrices $\mathbf{W}^{(\mathcal{U})}$ and $\{\mathbf{W}^{(\mathcal{D}_i)}\}_{i=1}^N$. Thus the within-domain transition probability matrices can be obtained by ($i = 1, \cdots, N$)

$$\mathbf{P}^{(\mathcal{U})} = (\mathbf{D}^{(\mathcal{U})})^{-1}\mathbf{W}^{(\mathcal{U})} \quad (14)$$

$$\mathbf{P}^{(\mathcal{D}_i)} = (\mathbf{D}^{(\mathcal{D}_i)})^{-1}\mathbf{W}^{(\mathcal{D}_i)} \quad (15)$$

where $\mathbf{D}^{(\mathcal{U})}$ and $\{\mathbf{D}^{(\mathcal{D}_i)}\}_{i=1}^N$ are the degree matrices induced by $\mathbf{W}^{(\mathcal{U})}$ and $\{\mathbf{W}^{(\mathcal{D}_i)}\}_{i=1}^N$. The final steady-state probability matrices can be iteratively calculated by

$$\mathbf{R}^{(\mathcal{U})}(t+1) = \alpha\mathbf{P}^{(\mathcal{U})}\mathbf{R}^{(\mathcal{U})}(t) + (1-\alpha)\mathbf{I} \quad (16)$$

$$\mathbf{R}^{(\mathcal{D}_i)}(t+1) = \beta_i\mathbf{P}^{(\mathcal{D}_i)}\mathbf{R}^{(\mathcal{D}_i)}(t) + (1-\beta_i)\mathbf{I} \quad (17)$$

where $i = 1, 2, \cdots, N, \ 0 \le \alpha, \beta_1, \cdots, \beta_N \le 1$.

For cross-domain subgraphs $\{\mathcal{G}^{(\mathcal{U}\mathcal{D}_i)}\}_{i=1}^N$, we also calculate the edge weight matrices $\{\mathbf{W}^{(\mathcal{U}\mathcal{D}_i)}\}_{i=1}^N$ based on the user interactions with other item domains $\{\mathcal{D}_i\}_{i=1}^N$. Thus the cross-domain transition probability matrices can be computed as (with $i = 1, \cdots, N$)

$$\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+} = (\mathbf{D}^{(\mathcal{U}\mathcal{D}_i)^+})^{-1}\mathbf{W}^{(\mathcal{U}\mathcal{D}_i)^+} \quad (18)$$

$$\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-} = (\mathbf{D}^{(\mathcal{U}\mathcal{D}_i)^-})^{-1}\mathbf{W}^{(\mathcal{U}\mathcal{D}_i)^-} \quad (19)$$

When updating the cross-domain transition probability matrices, we also consider the transition routes shown in Figure 3 and Figure 4, so that they can be updated using the following rules ($i = 1, 2, \cdots, N$).

$$\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(t+1) = \delta_i\mathbf{R}^{(\mathcal{U})}(t)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(t) + (1-\delta_i)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(t)\mathbf{R}^{(\mathcal{D}_i)}$$

$$\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(t+1) = \delta_i\mathbf{R}^{(\mathcal{U})}(t)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(t) + (1-\delta_i)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(t)\mathbf{R}^{(\mathcal{D}_i)}$$

$$\mathbf{R}^{(\mathcal{U})}(t+1) = \sum_{\mathcal{D}_i \in \mathcal{D}} \tau_i\mu_i\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(t)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(t)^T$$

$$+ \sum_{\mathcal{D}_i \in \mathcal{D}} \tau_i(1-\mu_i)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(t)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(t)^T$$

$$+ \tau^{(\mathcal{U})}\mathbf{R}^{(\mathcal{U})}(t)\mathbf{R}^{(\mathcal{U})}(t)^T \quad (20)$$

where $0 \le \delta_i, \mu_i, \tau_i \le 1$ ($i = 1, 2, \cdots, N$) are the tradeoff parameters. For a domain $\mathcal{D}_i$ without negative user-item links, the updating process of user tie strength $\mathbf{R}^{(\mathcal{U})}$ sets $\mu_i = 1$.

Algorithm 2 summarizes the whole procedure of random walk on high-order hybrid star-structured graph for predicting user adoptions on different item domains. The space complexity of Algorithm 2 is $O(m^2 + 2m\sum|\mathcal{D}_i| + \sum|\mathcal{D}_i|^2)$. And the time complexity is $O((m^2 + 4m\sum|\mathcal{D}_i| + 2\sum|\mathcal{D}_i|^2)mT)$, where T is the number of iterations. In practice, we only need to check users that are in a small community or circle and it takes only a few iterations to converge.

4. EXPERIMENTS

In this section, we introduce experimental results of applying our hybrid random walk method to a real world dataset, which is a second-order star-structured graph with web posts and user labels as two item domains. We evaluate the social recommendation performance on user-post adoption and rejection links.

4.1 Dataset

The dataset we use is crawled during January 2011 from Tencent Weibo website (`t.qq.com`), which is a Twitter-style microblogging service in China. We crawled data from users who own at least one user label, 17.2% of total. While the website allows users to edit at most 9 labels, the average number of labels per user we have is 6.2. We did not filter any social relationships. The average number of friends per user is 20.6.

Algorithm 2 Iterative Adoption Prediction through Random Walk over a High-Order Hybrid Star-Structured Graph

Require: $0 \le \alpha, \{\beta_i\}_{i=1}^N, \{\delta_i\}_{i=1}^N, \{\mu_i\}_{i=1}^N, \{\tau_i\}_{i=1}^N \le 1$
1: Construct $\mathcal{G}^{(\mathcal{U})}, \{\mathcal{G}^{(\mathcal{D}_i)}\}_{i=1}^N, \{\mathcal{G}^{(U\mathcal{D}_i)}\}_{i=1}^N$
2: Compute transition probabilities $\mathbf{P}^{(\mathcal{U})}$ and $\{\mathbf{P}^{(\mathcal{D}_i)}\}_{i=1}^N$ using Eq.(14) and Eq.(15)
3: Derive steady-state distributions $\mathbf{R}^{(\mathcal{U})}$ and $\{\mathbf{R}^{(\mathcal{D}_i)}\}_{i=1}^N$ using Eq.(16) and Eq.(17)
4: Initialize the transition probability matrices $\{\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(0)\}_{i=1}^N$ and $\{\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(0)\}_{i=1}^N$ using Eq.(18) and Eq.(19).
5: **for** $t = 1 : T$ **do**
6:    Compute the user state distributions $\mathbf{R}^{(\mathcal{U})}(t)$ and transition probability matrices $\{\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(t)\}_{i=1}^N$ and $\{\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(t)\}_{i=1}^N$ using Eq.(20)
7: **end for**
8: **Output:** The final transition probability matrices $\mathbf{R}^{(\mathcal{U})}$, $\{\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}\}_{i=1}^N$ and $\{\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}\}_{i=1}^N$

Table 3 summarizes the basic information contained in the dataset, where we have 53,438 users, 141,879 web posts and 111 user labels. Both web-post and user-label domains are sparse but in different levels, i.e., the positive density, which represents the percentage of positive observed links between every user and item, of web-post domain is 0.02% and the negative density is 0.04%, while the density of user-label domain is 5.57%, which is nearly 100 times higher.

Table 3: Dataset Information

Domain	Object	Cross-Domain Link	
		Accept (+)	Refuse (−)
user	53,438	–	–
web post	141,879	1,474,389 0.02%	3,399,980 0.04%
user label	111	330,157 5.57%	–

4.2 Experimental Settings

We present our experimental settings in detail in this section, including how we collect negative examples, i.e., user-item rejection links, and parameter settings.

In social network analysis, negative examples reveal facts on information adoption and help improve the recommendation performance [13]. However, we cannot observe user-item rejection behaviors from crawled data. Here we estimate the negative links in the following steps: (1) We collect the items that each user receives from his/her friends or followers. The social networking system organizes these items in chronological order. (2) We observe user adoptions to set up a time window (the time period before adopting items) within which users are considered online and having read all items that were sent during the period. (3) We identify rejected items (negative examples) as those received by a user within a 5-minute time window but never got adopted. Later, we will show the benefits of using the estimated negative samples.

With regard to parameter settings, we tune δ and η as the relative weights of user tie strength over item similarity

on user-post and user-label link prediction, μ as the relative weight of positive samples over negative samples on user-post link prediction, $\tau^{(\mathcal{P})}$ and $\tau^{(\mathcal{T})}$ as the relative weights of cross-domain links from web-post domain over user-label domain on influencing user tie strength. All parameters range from 0 to 1 and we tune them by gradient search to explain more on our method. We set the step length as 0.001 and in each iteration find the directions of these 5 parameters (δ, η, μ, $\tau^{(\mathcal{P})}$ and $\tau^{(\mathcal{T})}$) to reduce the error metrics.

4.3 Evaluation Metrics

We adopt three types of evaluation measures: reconstruction error, prediction accuracy and ranking-based metrics.

- Mean Absolute Error (MAE), which is calculated as

$$MAE = \frac{1}{N} \sum_{u_i,p_j} (|p_{ij}^{(\mathcal{UP})^+} - \hat{p}_{ij}^{(\mathcal{UP})^+}| + |p_{ij}^{(\mathcal{UP})^-} - \hat{p}_{ij}^{(\mathcal{UP})^-}|)$$
(21)

where $p_{ij}^{(\mathcal{UP})^+}$ and $p_{ij}^{(\mathcal{UP})^-}$ are the ground truth adoption and rejection of user u_i on item p_j in the testing set, $\hat{p}_{ij}^{(\mathcal{UP})^+}$ and $\hat{p}_{ij}^{(\mathcal{UP})^-}$ denote the prediction result, and N denotes the size of the testing set.

- F1 measure, which is another measure to evaluate the prediction performance. It is the harmonic mean of *precision* and *recall*, which are defined as follows:

$$precision = \frac{|\{(u_i, p_j)|\hat{p}_{ij}^{(\mathcal{UP})^+} > \hat{p}_{ij}^{(\mathcal{UP})^-}, p_{ij}^{(\mathcal{UP})^+} = 1\}|}{|\{(u_i, p_j)|\hat{p}_{ij}^{(\mathcal{UP})^+} > \hat{p}_{ij}^{(\mathcal{UP})^-}\}|}$$

$$recall = \frac{|\{(u_i, p_j)|\hat{p}_{ij}^{(\mathcal{UP})^+} > \hat{p}_{ij}^{(\mathcal{UP})^-}, p_{ij}^{(\mathcal{UP})^+} = 1\}|}{|\{(u_i, p_j)|p_{ij}^{(\mathcal{UP})^+} = 1\}|}$$
(22)

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

- Kendall's ranking coefficient $\hat{\tau}$ [3]. We rank the testing pairs $r(k)$ according to $|\hat{p}_{ij}^{(\mathcal{UP})^+} - \hat{p}_{ij}^{(\mathcal{UP})^-}|$. Then we compute Kendall's ranking coefficient $\hat{\tau}$ as follows, which computes how many pairs in the testing set are ordered correctly by our method.

$$\hat{\tau} = 1 - \frac{2 \sum_{k_1 < k_2} 1 \cdot (r(k_1) > r(k_2))}{N(N-1)}$$
(23)

where 1 corresponding to perfect ranking and 0 representing the reverse ranking.

4.4 Baseline Methods

Table 4: Comparison of Our Method (HRW) and Different Configurations.

Algorithm	$\mathbf{R}^{(\mathcal{U})}$	$\mathbf{W}^{(\mathcal{U})}$	$\mathbf{W}^{(\mathcal{P})}$	$\mathbf{W}^{(\mathcal{T})}$
HRW	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
BRW-R_U-P (TrustWalker)	$\checkmark$	$\checkmark$	$\checkmark$	$\times$
BRW-R_U	$\checkmark$	$\checkmark$	$\times$	$\times$
BRW-W_U-P	$\times$	$\checkmark$	$\checkmark$	$\times$
BRW-W_U (ItemRank)	$\times$	$\checkmark$	$\times$	$\times$
BRW-P	$\times$	$\times$	$\checkmark$	$\times$

We first prove the rationality and necessity of the matrices in our *Hybrid Random Walk* method, and then we compare it with several state-of-the-art algorithms. Table 4 shows the comparison of our method and its different configurations. These algorithms learn within-domain links in social domain, web post domain and user-label domain from $\mathbf{W}^{(\mathcal{U})}$, $\mathbf{W}^{(\mathcal{P})}$ and $\mathbf{W}^{(\mathcal{T})}$, and learn user tie strength of user graph by updating $\mathbf{R}^{(\mathcal{U})}$, i.e., weights on social links. The methods predict user-post links with *Bipartite Random Walk* (BRW), if they take no use of rich knowledge from user-label domain. Besides the RWR models, we take the newest matrix factorization algorithm which incorporates rich user and item information, and a standard collaborative filtering method as baselines.

- BRW-R_U-P, *TrustWalker* [10], which combines item-based recommendation and the weighted user graph as the trust network, with post similarity utilized and user tie strength updated.

- BRW-R_U, which predicts user-post links, with user tie strength updated on a bipartite graph.

- BRW-W_U-P, which predicts user-post links, with social relation and post similarity.

- BRW-W_U, *ItemRank* [8], which uses social relational graph to compute a biased value of PageRank as the probabilistic result of user-post link prediction.

- BRW-P, which learns from post similarities to predict user-post links with a random walk model by item-based recommendation.

- MCF, *Matrix Co-factorization* [5], which incorporates rich user and item information into recommendation by an approximation equation:

$$J(U, V, X, Y) = ||W \odot (\mathbf{P}^{(\mathcal{UP})^+} - UV)||_F^2$$
$$+ \lambda_1(||\mathbf{P}^{(\mathcal{UT})^+} - UX||_F^2 + ||B - YV||_F^2)$$
$$+ \lambda_2(||U||_F^2 + ||V||_F^2 + ||X||_F^2 + ||Y||_F^2)$$
(24)

where W is the weight matrix, λ_1 and λ_2 are regularization parameters. U and X represent user's preferences over latent factors, while V and Y represents item's word distributions over latent factors. All these 4 low-rank matrices can be solved by weighted Alternative Least Square.

- CF, *Item-based Collaborative Filtering* [23], which is based on the idea that users like posts recommended by others who have similar adoption behaviors, with users' social relations.

4.5 Experimental Results

In this section, we demonstrate the prediction performance of the proposed method. We randomly select 80% of user-post links for training and the remaining for testing, while user-label links are completely utilized. This random selection is carried out 20 times independently. Table 5 and 6 compare the performances of our method with its different configurations and with the above baselines, respectively, on the average results and standard deviations of evaluations including the MAE, precision, recall, F1 measure and Kendall's ranking-based method. Our method achieves the

Table 5: Experimental Results of the Different Configurations of Our Method

Algorithm	MAE	Precision	Recall	F1	Kendall's $\hat{\tau}$
HRW	**0.227±1.5e-3**	**0.711±1.3e-3**	0.921±1.4e-3	**0.802±1.1e-3**	**0.792±2.5e-3**
BRW-R_U-P (TrustWalker)	0.276±1.1e-3	0.657±7.6e-4	**0.935±9.8e-4**	0.772±7.6e-4	0.774±1.6e-3
BRW-R_U	0.282±5.3e-3	0.655±4.0e-3	0.921±1.2e-2	0.765±7.7e-3	0.725±2.8e-3
BRW-W_U-P	0.292±1.1e-3	0.666±7.0e-4	0.900±5.2e-4	0.765±6.6e-4	0.725±8.5e-4
BRW-W_U (ItemRank)	0.318±1.4e-3	0.671±1.5e-3	0.713±2.4e-3	0.691±1.2e-3	0.661±2.2e-3
BRW-P	0.438±2.6e-4	0.571±3.4e-4	0.499±4.2e-4	0.532±3.2e-4	0.606±2.3e-4

Table 6: Experimental Results of Our Method and Baselines

Algorithm	MAE	Precision	Recall	F1	Kendall's $\hat{\tau}$
HRW	**0.227±1.5e-3**	**0.711±1.3e-3**	0.921±1.4e-3	**0.802±1.1e-3**	**0.792±2.5e-3**
BRW-R_U-P (TrustWalker) [10]	0.276±1.1e-3	0.657±7.6e-4	0.935±9.8e-4	0.772±7.6e-4	0.774±1.6e-3
BRW-W_U (ItemRank) [8]	0.318±1.4e-3	0.671±1.5e-3	0.713±2.4e-3	0.691±1.2e-3	0.661±2.2e-3
MCF [5]	0.352±2.3e-4	0.592±1.8e-3	**0.951±6.0e-4**	0.730±1.3e-3	0.582±4.3e-4
CF [23]	0.506±3.4e-4	0.552±1.5e-3	0.589±7.2e-4	0.570±1.0e-3	0.540±5.2e-4

best performance in experimental trials and it is insensitive to initialization.

From Table 5, we can observe that:

- BRW-W_U (ItemRank) reduces MAE by 27.4% over BRW-P, an item-based recommendation implemented by the random walk algorithm. BRW-W_U exploits user dependent preferences from friendships and performs better than the collaborative approach on large social datasets. BRW-R_U reduces MAE by 11.3% over BRW-W_U, which updates user tie strength on the user graph with user-post links. BRW-R_U-P reduces MAE by 13.2% over BRW-W_U, which learns both within-domain links (post similarities) and cross-domain links (user-post links) to update user tie strength. The reason is that the motivations of user behavior on social networks are: (1) Users like to adopt web posts which highly correlate with those adopted before. (2) Users like to adopt posts recommended by their friends or followers with high tie strength. BRW-R_U-P combines these two aspects to solve the social recommendation problem.

- HRW reduces MAE by 17.8% over BRW-R_U-P (TrustWalker). This is consistent with the assumption that in social networks, user tie strength on user graph is shaped by multiple relational domains such as web-post and user-label domains. Our method effectively utilizes auxiliary information to formulate the weighted user graph, and performs the best in solving the sparsity problem of user-post link prediction.

From Table 6, we can observe that:

- HRW outperforms the matrix co-factorization approach MCF with side information of posts and users (word distributions of posts and user-label links) incorporated, reducing MAE by 35.5%. Though MCF takes rich information about numerous resources and users, when accommodated to social recommendation, it needs careful consideration. User behaviors on social networks stem from the interrelationships among users, tight or loose, that have been naturally shaped. However, MCF does not involve the user tie strength.

- BRW-W_U implements the collaborative filtering technique via a random walk model. MAE of BRW-W_U is

37.2% lower than that of the traditional item-based CF method. It reveals the advantage of random walk models which predict missing user-post links and then reversely influence other within-domain and cross-domain link weights. It is adequate to design a random walk model on this social problem.

4.6 Insights

In this section we discuss the insights we have gained from experimental results. Specifically, we want to discuss three questions.

(1) Are item similarity and tie strength important in predicting user-post links?

On web-post domain, δ is the weight of tie strength over web post similarity in calculating links. If δ increases, users are more likely to accept the recommended posts for their social relationships than preferences. η is the corresponding weight on user-label domain. Figure 5(a) shows the performances varying δ and η from 0 to 1. For δ, $\delta = 0$ means prediction without considering item similarity, while $\delta = 1$ means prediction without considering social tie strength. From the figure we can clearly observe a valley when δ is around 0.4, which means that incorporating both tie strength and post similarity can significantly improve the performance. Similar trends can be observed from the changing MAE curves when we vary η.

(2) Are user-post and user-label links important in influencing user tie strength?

$\tau^{(\mathcal{P})}$ and $\tau^{(\mathcal{T})}$ correspond to the weights of knowledge learned from user-post and user-label links on calculating the tie strength, with $\tau^{(\mathcal{P})} + \tau^{(\mathcal{T})} = 1$. Figure 5(b) shows the MAE varying $\tau^{(\mathcal{P})}$ from 0 to 1. When $\tau^{(\mathcal{P})} = 1$, i.e., we discard the knowledge transferred from user-label links on tie strength, the resultant MAE is around 0.28. When $\tau^{(\mathcal{P})} = 0$, i.e, $\tau^{(\mathcal{T})} = 1$, MAE is around 0.26, which can be explained that user-label domain is more helpful in predicting tie strength since it is easier for congenial users to share the same user labels than the same posts. The minimum MAE ($\tau^{(\mathcal{P})} = 0.6$) suggests that knowledge from user-label links provides potential clues on user-post link prediction through user tie strength and recommender systems should incorporate these two kinds of links.

(3) Are negative samples of web-post domain helpful?

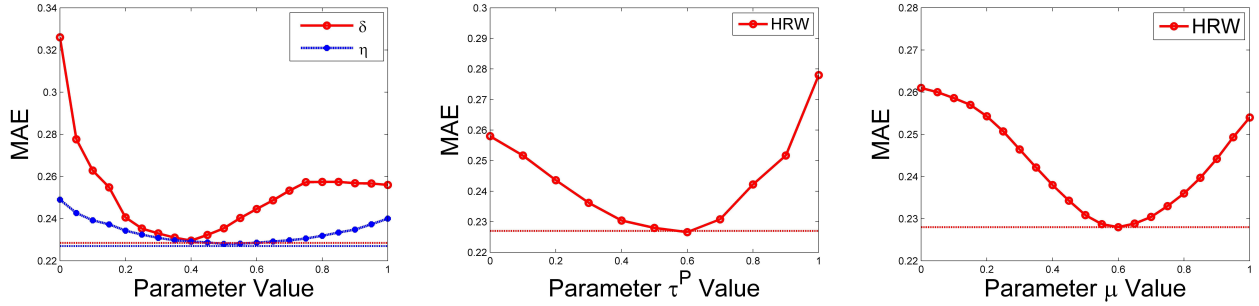μ is the relative weight of positive samples from web-post

Figure 5: (a) MAE v.s. δ and η. Performances on tuning weights of tie strength and item similarity on cross-domain link prediction. (b) MAE v.s. $\tau^{(\mathcal{P})}$. Performances on tuning weights of knowledge from web post and user-label domains transferred to user tie strength. (c) MAE v.s. μ. Performances on tuning weights of positive and negative user-post samples on user tie strength.

domain influencing user tie strength, while $1 - \mu$ is that of negative samples. Figure 5(c) shows the changing curve of MAE when varying μ from 0 to 1. When $\mu = 1$ (only train positive samples) and $\mu = 0$ (only train negative samples), the MAE is higher than taking both samples for training ($\mu = 0.6$). This means that considering both positive and negative samples in the recommendation procedure would be very helpful for the final algorithm performance. MAE is lower with only negative samples than with only positive ones.

The above discussion proves that our method is reasonable and effective, which considers comprehensive factors on user behavior and user tie strength while taking both positive and negative samples seriously.

4.7 Performance on Cold Start Problem

In this section, we show the performance of HRW when we control the density of training entries of testing users (the percentage of training user-post links per testing user, testing item), and compare its performance with the version without using knowledge transferred from user-label domain (BRW-R_U-P) as the baseline, and the results are shown in Figure 6. If no training entry is hidden, where the density of training entries of testing users is 0.06%, our model reduces MAE by 11.3% over the baseline (0.227 over 0.256). If we hide all training entries, i.e., the density of training entries of testing users is zero, which means the testing users are new in the application without previous behaviors in the historical dataset, our model reduces MAE by 34.4% (0.288 over 0.387). From Figure 6, we observe that the performance gain increases as the cold start problem becomes more serious. Moreover, our transfer learning model needs only 29.5% of the training entries (0.018% dense) to reach the same level performance of BRW-R_U-P with the whole training set (0.061% dense), which saves 70.5% of training information. With user label of a new user, our method needs only 3-day historical data to reach the same recommendation performance of 10-day data without labels. Therefore, if we motivate new users to add several user labels, the transferred knowledge from user-label domain would greatly improve user experience on personal recommendation services.

5. CONCLUSION

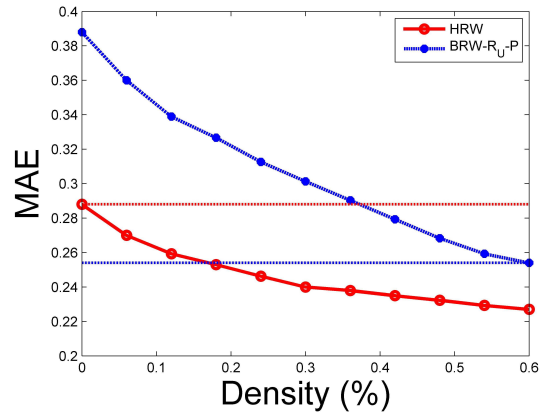In this paper, we address the problems of sparsity and

Figure 6: Performance of our transfer learning method (HRW) and one without transferring method (BRW-R_U-P) on the cold start problem.

cold start in social recommendation. Here we rethink the problem from the transfer learning perspective and alleviate the sparsity problem in the target domain by transferring knowledge from other auxiliary social relational domains. By considering the special structures of multiple relational domains in social networks, we propose a novel *Hybrid Random Walk* method on a star-structured graph, which is a general method to incorporate complex and heterogeneous link structures, including directed/undirected links, sign/unsigned links and within-domain/cross-domain links.

We have conducted extensive experiments on a large real-world social network dataset and showed that the proposed method can greatly boost the social recommendation performance. In particular, we have gained improvement in web post recommendation by transferring knowledge from user-label domain for user tie strength updating process, compared with the recommendation methods which solely use the information in web-post domain. Also, we have demonstrated that, by using only 29.5% of the available information in the target domain, our method can achieve equal performance with the methods that use all the available information in target domain without transfer learning. The proposed method and insightful experiments indicate a

promising and general way to solve the sparsity problem in different real-world recommendation scenarios.

7. REFERENCES

[1] J. Brown, A. J. Broderick, and N. Lee. Word of mouths communication within online communities: Conceptualizing the online social network. *Journal of Interactive Marketing*, 21:2–20, August 2007.

[2] B. Cao, N. N. Liu, and Q. Yang. Transfer learning for collective link prediction in multiple heterogenous domains. *In Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 159–166, 2010.

[3] P. Cui, F. Wang, S. Liu, M. Ou, S. Yang, and L. Sun. Who should share what? item-level social influence prediction for users and posts ranking. *In Proceedings of SIGIR*, pages 185–194, 2011.

[4] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TIS)*, 22:143–177, January 2004.

[5] Y. Fang and L. Si. Matrix co-factorization for recommendation with rich side information and implicit feedback. *International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pages 65–69, 2011.

[6] B. Gao, T. Liu, and W. Ma. Star-structured high-order heterogeneous data co-clustering based on consistent information theory. *In Proceedings of ICDM*, pages 880–884, 2006.

[7] B. Gao, T. Liu, X. Zheng, Q. Cheng, and W. Ma. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. *In Proceedings of KDD*, pages 41–50, 2005.

[8] M. Gori and A. Pucci. Itemrank: A random-walk based scoring algorithm for recommender engines. *In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2766–2771, 2007.

[9] M. Harvey, M. J. Carman, I. Ruthven, and F. Crestani. Bayesian latent variable models for collaborative item rating prediction. *In Proceedings of CIKM*, pages 699–708, 2011.

[10] M. Jamali and M. Ester. Trustwalker: A random walk model for combining trust-based and item-based recommendation. *In Proceedings of KDD*, pages 397–406, 2009.

[11] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. *In Proceedings of SIGIR*, pages 195–202, 2009.

[12] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42:30–37, August 2009.

[13] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: Mining a social network with negative edges. *In Proceedings of WWW*, pages 741–750, 2009.

[14] J. Leskovec, A. Singh, and J. Kleinberg. Patterns of influence in a recommendation network. *Lecture Notes in Computer Science*, 3918:380–389, 2006.

[15] Y. Li, J. Hu, C. Zhai, and Y. Chen. Improving one-class collaborative filtering by incorporating rich user information. *In Proceedings of CIKM*, pages 959–968, 2010.

[16] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. *In Proceedings of CIKM*, pages 199–208, 2010.

[17] N. N. Liu, M. Zhao, and Q. Yang. Probabilistic latent preference analysis for collaborative filtering. *In Proceedings of CIKM*, pages 759–766, 2009.

[18] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: Social recommendation using probabilistic matrix factorization. *In Proceedings of CIKM*, pages 931–940, 2008.

[19] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender system with social regularization. *In Proceedings of WSDM*, pages 287–296, 2011.

[20] P. Massa and P. Avesani. Trust-aware recommender systems. *In Proceedings of the First ACM Conference on Recommender Systems (RecSys)*, pages 17–24, 2007.

[21] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22:1345–1359, October 2010.

[22] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang. Transfer learning in collaborative filtering for sparsity reduction. *In Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, pages 230–235, 2010.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. *In Proceedings of WWW*, pages 285–295, 2001.

[24] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. *In Proceedings of KDD*, pages 650–658, 2008.

[25] L. Tang and H. Liu. Relational learning via latent social dimensions. *In Proceedings of KDD*, pages 817–826, 2009.

[26] H. Tong, C. Faloutsos, and J. yu Pan. Fast random walk with restart and its applications. *In Proceedings of ICDM*, pages 613–622, 2006.

[27] J. Wang, H. Zeng, Z. Chen, H. Lu, L. Tao, and W.-Y. Ma. Recom: Reinforcement clustering of multi-type interrelated data objects. *In Proceedings of SIGIR*, pages 274–281, 2003.

[28] M. Wang, K. Yang, X.-S. Hua, and H. Zhang. Towards a relevant and diverse search of social images. *IEEE Transactions on Multimedia*, 12(8):829–842, August 2010.

[29] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. *In Proceedings of the 20th International Conference on Machine Learning (ICML)*, 2003.