Contents

List of Figures

List of Tables

CHAPTER 1

Mining User Behaviors in Large Social Networks

Meng Jiang

University of Illinois at Urbana-Champaign

Peng Cui

Tsinghua University

CONTENTS

GIVEN large social networks that have recorded millions of users and items (*e.g.*, messages on Facebook/Twitter, articles on Shashdot/Digg), billions of item adoption behaviors and millions of user connection behaviors, what is the underlying behavioral mechanism of the social network users? Can we accurately predict the most probable

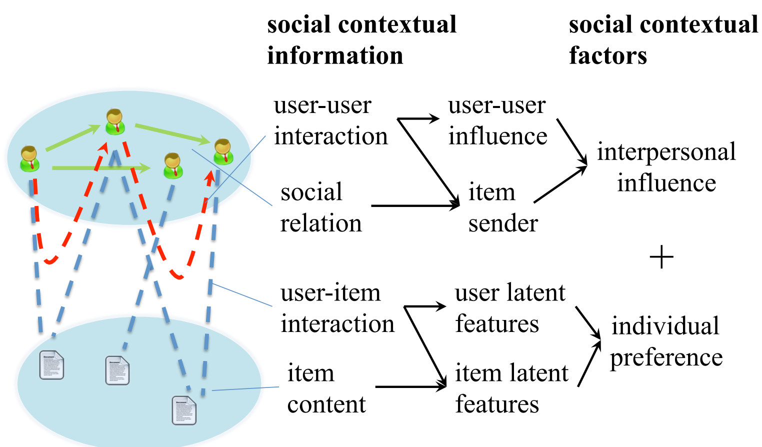


Figure 1.1 Contextual information in social networks.

items that the users will adopt? Can we recommend them what they really want? How can we alleviate the challenging problem of high sparsity of the behavioral data? The general philosophy underlying these applications is a deep understanding of the multi-aspect nature of users' behavioral intention.

This chapter will answer the above questions by introducing a series of work on mining user behaviors in three different thrusts: (1) uncovering the behavioral mechanism of social network users; (2) modeling context-aware behavior for recommendation and prediction; (3) mining cross-domain behavior in social networks to address the sparsity and cold-start issues.

1.1 MINING BEHAVIORAL MECHANISM FOR SOCIAL RECOMMENDATION

Social network users generate large volumes of information such as blogs, articles and tweets. Exponential growth of information generated by online social networks demands effective and scalable recommender systems to give useful results. A widely-used definition of social recommendation is any recommendation with online social relations as an additional input, *i.e.*, augmenting an existing recommendation engine with additional social signals [28]. Social relations can be trust relations, friendships, memberships or following relations [12, 9]. Besides the social relations, the social networks usually contain four types of contextual information: (1) user-user interaction, (2) social relation, (3) user-item interaction, and (4) item content, as show in Figure 1.1.

When receiving a new item (*e.g.*, message, article), users typically examine item content and information on senders [1, 25]. For example, in Twitter, when a user receives a tweet that is posted by one of his friends (the sender), she usually reads its content to see whether the item is interesting. Her preference considers both item content and user-item interaction information [24]. At the same time, the user cares about who the sender is and whether the sender is a close friend or authoritative [21, 7, 20]. If more than one friend sends her the same tweet, she may read it more attentively. Influence from her friends considers both social relation and user-user interaction information [19, 4, 3, 2, 5]. Both of these aspects are important for the user whether to adopt the item (*e.g.*, sharing, retweeting). The contextual information can be summarized as two contextual factors: (1) *individual preference* and (2) *interpersonal influence*. This section will study the social contexual factors derived from the rich information, and integrate them into a unified social recommendation framework [10, 13].

1.1.1 Social contextual factor anlaysis

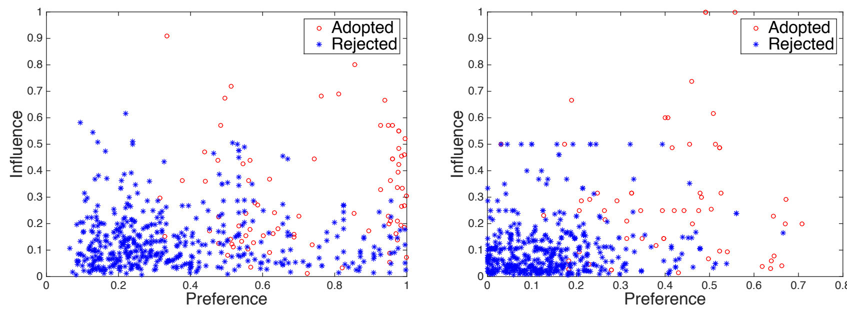
Statistical analysis results demonstrate the effect of social contextual factors (*i.e.*, individual preference and interpersonal influence) on users' decisions in social networks (*i.e.*, adopting or rejecting the received item). Two real-world social network datasets used in this section enable quantification of the factors. They are *Renren*, a Facebook-style social website in China, and *Tencent Weibo* one of the largest Twitter-style microblogging platforms in China.

Given an item, individual preference describes how much a user likes it, in other words, how much the topics/semantics of this item match the user's preferred topic distribution. Existing probabilistic graphical models such as advanced variants of the LDA model can extract topic-level distributions of the items. A straightforward preference measurement for how much user u likes item a is defined as

$$P_u(a) = T_a \cdot (\frac{1}{|A(u,a)|} \sum_{a' \in A(u,a)} T_{a'}), \tag{1.1}$$

where $A(u,a)$ is the set of items adopted by user u excluding a, and T_a is the topic distribution of item a.

Interpersonal influence describes whether the user has close rela-



(a) Facebook-style Renren (b) Twitter-style Tencent Weibo

Figure 1.2 Contextual factors of users' decisions: individual preference and interpersonal influence.

tionships with the item senders (e.g., followees who post the tweet in Twitter) from the perspective of user-user interactions on social web. It can be defined as the percentage of recommended items adopted by u from u's friends or followees who send the item a:

$$I_u(a) = \frac{1}{|V(u,a)|} \sum_{v \in V(u,a)} \frac{|\mathcal{S}(u,v) \cap \mathcal{A}(u)|}{|\mathcal{S}(u,v)|}, \tag{1.2}$$

where $V(u,a)$ is the set of senders who send item a to user u, $\mathcal{S}(u,v)$ is the set of items sent from v to u, and $\mathcal{A}(u)$ is the set of items that u adopts.

A user-item pair (u, a) can be labeled as either *adopted* or *rejected*. Figure 1.2 plots the pairs as points (red for adopted items and blue for rejected items) by individual preference $P_u(a)$ vs interpersonal influence $I_u(a)$: Renren dataset on the left-hand and Tencent Weibo dataset on the right-hand. It is easy to observe that the red points are located at the top and right corner of both the figures, indicating that users intend to adopt items with higher preference scores and from higher influential friends or followees in social networks.

1.1.2 Social contextual modeling for recommendation

The goal is to address social recommendation problem by answering how to model social contextual information as contextual factors and integrate them into a framework.

Framework design. Figure 1.3 shows the social recommendation framework based on probabilistic matrix factorization to incorporate

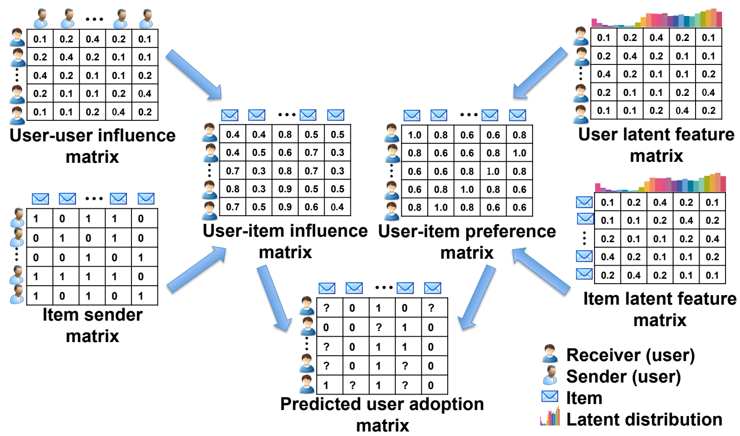


Figure 1.3 Social contextual recommendation framework.

individual preference and interpersonal influence [16, 18, 30]. Specifically, it factorizes the user-item interaction matrix into two intermediated latent matrices including the user-item influence matrix and user-item preference matrix, which are generated from three objective latent matrices: user latent feature matrix, item latent feature matrix, and user-user influence matrix. Moreover, it regularizes the three latent matrices with observed context-aware data including items' semantic similarity and users' interaction frequency.

The details of this framework are introduced as below. Suppose that there are M users with the i-th user denoted as u_i, and N items with the j-th item denoted as p_j. The information adoption matrix are denoted by $\mathbf{R} \in \{0,1\}^{M \times N}$, with its (i,j)-th entry as

$$R_{ij} = \begin{cases} 1 & \text{if } u_i \text{ adopted } p_j; \\ 0 & \text{otherwise.} \end{cases} \tag{1.3}$$

Then the social recommendation problem is converted to predict the unobserved entries in the information adoption matrix $\mathbf{R}$ based on the observed entries and other factors.

In social scenarios, whether a user adopts an item on social networks is determined by three aspects: (1) item content: what the item tells about, (2) user-item interaction: what items the user likes, and (3) social relation and user-user interaction: who the senders are. Let $\mathbf{U} \in \mathbb{R}^{k \times M}$ be the latent user feature matrix, and let $\mathbf{V} \in \mathbb{R}^{k \times N}$ be the latent item feature matrix. $\mathbf{S} \in \mathbb{R}^{M \times M}$ is the interpersonal influence matrix with each entry S_{ij} representing the value of influence user u_i has on user u_j. The value $S_{ij} > 0$ if and only if u_i is the friend of u_j in social networks such as Facebook and Renren, or is followed by u_j in microblogging services such as Twitter and Tencent Weibo. $\mathbf{G} \in \mathbb{R}^{N\times M}$ is the item sender matrix with entry $G_{ij} = 1$ meaning that u_j sends the item p_i and vice versa. Based on these denotations and the assumption that users can only receive items from their friends as social networks usually do ($G_{ii} = 0$), the social recommendation problem is to find out $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{S}$ so that $((\mathbf{S}\mathbf{G}^\top) \odot (\mathbf{U}^\top\mathbf{V}))$ can well approximate the observed entries in $\mathbf{R}$ without over-fitting, where $\odot$ is the Hadamard Product.

From the observed social network data, regularizations can be derived to improve learning of $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{S}$. The following formulas compute user-user preference similarity matrix $\mathbf{W} \in \mathbb{R}^{M\times M}$, item-item content similarity matrix $\mathbf{C} \in \mathbb{R}^{N\times N}$, and user-user interaction matrix $\mathbf{F} \in \mathbb{R}^{M\times M}$:

$$\begin{aligned} W_{i,j} &= \frac{\sum_{a\in\mathcal{A}(u_i)} P_{u_i}(a)}{|\mathcal{A}(u_i)|} \cdot \frac{\sum_{a'\in\mathcal{A}(u_j)} P_{u_j}(a')}{|\mathcal{A}(u_j)|}, && (1.4)\\ C_{i,j} &= T_{a_i} \cdot T_{a_j}, && (1.5)\\ F_{i,j} &= \frac{|\mathcal{S}(u_i,u_j) \cap \mathcal{A}(u_i)|}{|\mathcal{S}(u_i,u_j)|}. && (1.6) \end{aligned}$$

The heuristic for choosing regularizations on the three latent factors with the observed matrices are listed as follows: (1) the users that are similar in user latent space $\mathbf{U}$ have similar preferences (derived from preference similarity matrix $\mathbf{W}$); (2) the items that are similar in item latent space $\mathbf{V}$ have similar descriptive contents (derived from content similarity matrix $\mathbf{C}$); (3) high interpersonal influence in the influence latent space $\mathbf{S}$ generates frequent interpersonal interactions $\mathbf{F}$; (4) the product of user latent space $\mathbf{U}$ and item latent space $\mathbf{V}$ corresponds to the users' individual preference on the items; (5) the Hadamard product of interpersonal influence and individual preference is proportional to the probability of item adoptions.

By adopting a probabilistic model with Gaussian observation noise, the conditional distribution over the observed entries in $\mathbf{R}$ can be defined as

$$P(\mathbf{R}|\mathbf{S},\mathbf{U},\mathbf{V},\sigma_R^2) = \prod_{i=1}^{M}\prod_{j=1}^{N} \mathcal{N}(\mathbf{R}_{ij}|\mathbf{S}_i\mathbf{G}_j^\top \odot \mathbf{U}_i^\top\mathbf{V}_j, \sigma_R^2). \qquad (1.7)$$

Incorporating the social contextual factors, we define the posterior distribution as

$$
\begin{aligned}
& P(\mathbf{S},\mathbf{U},\mathbf{V}|\mathbf{R},\mathbf{G},\mathbf{W},\mathbf{C},\mathbf{F}) = \tfrac{P(\mathbf{R},\mathbf{G},\mathbf{W},\mathbf{C},\mathbf{F}|\mathbf{S},\mathbf{U},\mathbf{V})P(\mathbf{S},\mathbf{U},\mathbf{V})}{P(\mathbf{R},\mathbf{G},\mathbf{W},\mathbf{C},\mathbf{F})} \\
\propto\ & P(\mathbf{R}|\mathbf{S},\mathbf{U},\mathbf{V})P(\mathbf{W}|\mathbf{U})P(\mathbf{C}|\mathbf{V})P(\mathbf{F}|\mathbf{S})P(\mathbf{S})P(\mathbf{U})P(\mathbf{V}) \\
=\ & \prod_{i,j}\mathcal{N}(R_{ij}|\mathbf{S}_i\mathbf{G}_j^\top \odot \mathbf{U}_i^\top\mathbf{V}_j, \sigma_R^2) \\
& \prod_{p,q}\mathcal{N}(W_{pq}|\mathbf{U}_p^\top\mathbf{U}_q, \sigma_W^2)\prod_{m,n}\mathcal{N}(C_{mn}|\mathbf{V}_m^\top\mathbf{V}_n, \sigma_C^2)\prod_{s,t}\mathcal{N}(F_{st}|S_{st}, \sigma_F^2) \\
& \prod_{x}\mathcal{N}(\mathbf{S}_x|0, \sigma_S^2)\prod_{y}\mathcal{N}(\mathbf{U}_y|0, \sigma_U^2)\prod_{z}\mathcal{N}(\mathbf{V}_z|0, \sigma_V^2),
\end{aligned}
\tag{1.8}
$$

where $\sigma_{(.)}$ denotes that zero-mean spherical Gaussian priors are placed on latent feature vectors and observed matrices. Then the logarithm of the above probability is

$$
\begin{aligned}
& \ln P(\mathbf{S},\mathbf{U},\mathbf{V}|\mathbf{R},\mathbf{G},\mathbf{M},\mathbf{C},\mathbf{F}) \\
\propto\ & -\tfrac{1}{2\sigma_R^2}\sum_{i,j}(R_{ij}-\mathbf{S}_i\mathbf{G}_j^\top \odot \mathbf{U}_i^\top\mathbf{V}_j)^2 - \tfrac{1}{2\sigma_W^2}\sum_{p,q}(W_{pq}-\mathbf{U}_p^\top\mathbf{U}_q)^2 \\
& -\tfrac{1}{2\sigma_C^2}\sum_{m,n}(C_{mn}-\mathbf{V}_m^\top\mathbf{V}_n)^2 - \tfrac{1}{2\sigma_F^2}\sum_{s,t}(F_{st}-S_{st})^2 - \tfrac{1}{2\sigma_S^2}\sum_{x}\mathbf{S}_x^\top\mathbf{S}_x \\
& -\tfrac{1}{2\sigma_U^2}\sum_{y}\mathbf{U}_y^\top\mathbf{U}_y - \tfrac{1}{2\sigma_V^2}\sum_{z}\mathbf{V}_z^\top\mathbf{V}_z.
\end{aligned}
\tag{1.9}
$$

Maximizing the posterior distribution is equivalent to minimizing the sum-of-squared errors function with hybrid quadratic regularization terms:

$$
\begin{aligned}
\mathcal{J}(\mathbf{S},\mathbf{U},\mathbf{V}) \ =\ & ||\mathbf{R}-\mathbf{S}\mathbf{G}^\top \odot \mathbf{U}^\top\mathbf{V}||_F^2 + \alpha||\mathbf{W}-\mathbf{U}^\top\mathbf{U}||_F^2 + \beta||\mathbf{C}-\mathbf{V}^\top\mathbf{V}||_F^2 \\
& +\gamma||\mathbf{F}-\mathbf{S}||_F^2 + \delta||\mathbf{S}||_F^2 + \eta||\mathbf{U}||_F^2 + \lambda||\mathbf{V}||_F^2,
\end{aligned}
\tag{1.10}
$$

where $\alpha = \frac{\sigma_R^2}{\sigma_W^2}, \beta = \frac{\sigma_R^2}{\sigma_C^2}, \gamma = \frac{\sigma_R^2}{\sigma_F^2}, \delta = \frac{\sigma_R^2}{\sigma_S^2}, \eta = \frac{\sigma_R^2}{\sigma_U^2}, \lambda = \frac{\sigma_R^2}{\sigma_V^2}$, and $||.||_F$ is the Frobenius norm.

To minimize the above objective function, the framework adopts a block coordinate descent scheme to solve the problem. That is, starting from random initialization on $\mathbf{S}, \mathbf{U}, \mathbf{V}$, it optimizes each of them alternatively with the other two matrices fixed and proceed step by step until convergence. As the objective is obviously lower bounded by 0 and the alternating gradient search procedure will reduce it monotonically, the algorithm is guaranteed to be convergent. Using a gradient search method, the gradients of the objective with respect to the variables are

$$\frac{\partial \mathcal{J}}{\partial \mathbf{S}} = -2(\mathbf{R} - \mathbf{S}\mathbf{G}^\top \odot \mathbf{U}^\top \mathbf{V})\mathbf{G} - 2\gamma(\mathbf{F} - \mathbf{S}) + 2\delta\mathbf{S}, \tag{1.11}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{U}} = -2\mathbf{V}(\mathbf{R} - \mathbf{S}\mathbf{G}^\top \odot \mathbf{U}^\top \mathbf{V})^\top - 4\alpha\mathbf{U}(\mathbf{W} - \mathbf{U}^\top \mathbf{U}) + 2\eta\mathbf{U}, \tag{1.12}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{V}} = -2\mathbf{U}(\mathbf{R} - \mathbf{S}\mathbf{G}^\top \odot \mathbf{U}^\top \mathbf{V}) - 4\beta\mathbf{V}(\mathbf{C} - \mathbf{V}^\top \mathbf{V}) + 2\lambda\mathbf{V}. \tag{1.13}$$

$\mathcal{J}$ decreases the fastest in the direction of gradients during each iteration and the sequence $\mathcal{J}^{(t)}$ converges to the desired minimum.

1.1.3 Conclusions

This section introduced a social recommendation model utilizing social contextual factors, *i.e.*, individual preference and interpersonal influence. The extensive experiments showed that social contextual information can greatly boost the performance of recommendation on social network datasets.

1.2 MINING CONTEXTUAL BEHAVIOR FOR PREDICTION

Scientists study human behavior from a variety of cultural, political, and psychological perspectives, looking for consistent patterns of individual and social behavior and for scientific explanations on those patterns. The discovered patterns can be used in many real world applications such as web search, recommender system and advertisement targeting. It is well accepted that human behavior is the product of a multitude of interrelated factors. The factors such as physical environment, social interaction, and social identity, affect how the behavior takes place with our personalities and interests. As an example, if a researcher changes his affiliation, he will start to collaborate with new friends, join in new projects and eventually study new topics. Given the complexity of multi-faceted factors influencing human behaviors, it is difficult to concisely summarize what they are and how they interact. Moreover, psychological studies demonstrate that human behaviors naturally evolve with the changing of both endogenous factors (e.g., personality) and exogenous factors (e.g, environment), resulting in different dynamic (temporal) behavioral patterns over time [17, 23, 22]. For example, in the early 1990s, many researchers focused on database systems and query processing. In the late 1990s, with various data collective methods emerging and scales of unlabeled data increasing,

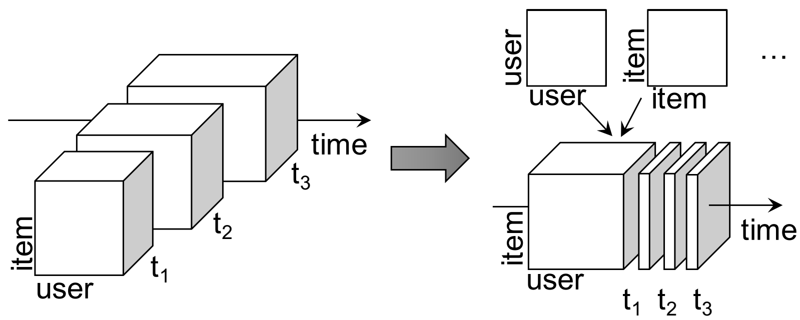


Figure 1.4 Tensor sequences with regularizations for modeling multi-faceted dynamic behaviors.

they turned to work on clustering and pattern mining problems. In the 2000s, people started to focus on social networks and communities since Facebook and Twitter become popular. Consequently, the patterns of human behaviors differ from place to place, era to era and across environments, depending on *spatio-temporal contexts*. These characteristics cause severe data sparsity and computational complexity problems: they pose great challenges to understanding and predicting human behaviors. Behavioral modeling with spatio-temporal contexts requires multi-faceted and temporal information [11]. This section will introduce how to model the multi-faceted dynamic behaviors, and how to develop fast algorithms for behavior prediction with advanced flexible evolutionary multi-faceted analysis (FEMA).

1.2.1 Modeling multi-faceted dynamic behaviors

Problem definition. Let the Facebook dataset be an example of this problem: we focus on finding temporal patterns of web posting behavior. The dataset can be denoted by a list of tuples (u, g, w, t): a Facebook user u at a geo-location g posts a message about a word w at time t $(t = 1, \ldots, T)$. The data is thus represented with a 3-order tensor sequence [26, 27, 15, 14] $\mathcal{X}_\mathbf{t} \in \mathbb{R}^{n^{(u)} \times n^{(g)} \times n^{(w)}}$, where $n^{(u)}$ is the number of users, $n^{(g)}$ is the number of geo-locations, and $n^{(w)}$ is the number of words. $\mathcal{X}_\mathbf{t}(u, g, w)$ has a value of the number of existing tuples (u, g, w, t') $(t' \leq t)$. The goal is to factorize the tensor sequence

$$\mathcal{X}_\mathbf{t} \approx \mathcal{Y}_\mathbf{t} \times_{(u)} \mathbf{U_t} \times_{(g)} \mathbf{G_t} \times_{(w)} \mathbf{W_t} \tag{1.14}$$

where

1. $\mathcal{Y}_{\mathbf{t}} \in \mathbb{R}^{r^{(u)} \times r^{(g)} \times r^{(w)}}$ is the core tensor sequence, which encodes the temporal behavioral patterns, i.e., the relationship among user, geo-location and word groups. $\mathcal{Y}_{\mathbf{t}}(j^{(u)}, j^{(g)}, j^{(w)})$ indicates the probability of the behavior before time t if the $j^{(u)}$-th user group in the $j^{(g)}$-th geo-location group posts about the $j^{(w)}$-th word group.
2. $\mathbf{U_t} \in \mathbb{R}^{n^{(u)} \times r^{(u)}}$ is the users' projection matrix before time t. $\mathbf{U}(i^{(u)}, j^{(u)})$ represents the probability that the $i^{(u)}$-th user belongs to the $j^{(u)}$-th group before time t.
3. $\mathbf{G_t} \in \mathbb{R}^{n^{(g)} \times r^{(g)}}$ is the geo-locations' projection matrix before time t. $\mathbf{G_t}(i^{(g)}, j^{(g)})$ represents the probability that the $i^{(g)}$-th geo-location belongs to the $j^{(g)}$-th group before time t.
4. $\mathbf{W_t} \in \mathbb{R}^{n^{(w)} \times r^{(w)}}$ is the words' projection matrix before time t. $\mathbf{W_t}(i^{(w)}, j^{(w)})$ represents the probability that the $i^{(w)}$-th word belongs to the $j^{(w)}$-th group before time t.

Figure 1.4 describes how to model a time-evolving dataset - using tensor sequence. The key to solving the sparsity problem in *tensor sequence* decompositions is to learn the *flexible regularizers* such as the users' social relations, the geo-locations' geographical distance and the words' semantic information. The regularizers can be encoded as Laplacian matrices $\mathbf{L^{(u)}}$, $\mathbf{L^{(g)}}$, $\mathbf{L^{(w)}}$, where the (i, j)-th element represents the similarity between the i-th and j-th entities (users, geo-locations, words). Examples of possible similarity functions are the frequency of user communication and the proximity of geographic locations.

The problem is now how to compute the factorizations for the core tensor sequence and projection matrices, given the tensor sequence and constraints. Note that the scale of the tensors are large but the changes are very small. We denote by $\Delta\mathcal{X}_{\mathbf{t}}$ the increment at time t, which is very sparse: for any $1 \leq t < T$, $\Delta\mathcal{X}_{\mathbf{t}} = \mathcal{X}_{\mathbf{t+1}} - \mathcal{X}_{\mathbf{t}}$. The problem can be summarized into two steps:

1. **Given** the first tensor $\mathcal{X}_{\mathbf{1}}$ and the constraints $\mathbf{L^{(u)}}$, $\mathbf{L^{(g)}}$, $\mathbf{L^{(w)}}$, **find** the projection $\mathbf{U_1}$, $\mathbf{G_1}$, $\mathbf{W_1}$, and the first core tensor $\mathcal{Y}_{\mathbf{1}}$.
2. At time t $(1 \leq t < T)$, **given** the tensor $\mathcal{X}_{\mathbf{t}}$, the increment $\Delta\mathcal{X}_{\mathbf{t}}$, the old projection matrices $\mathbf{U_t}$, $\mathbf{G_t}$, $\mathbf{W_t}$, and the constraints $\mathbf{L^{(u)}}$, $\mathbf{L^{(g)}}$, $\mathbf{L^{(w)}}$, **find** the new projection matrices $\mathbf{U_{t+1}}$, $\mathbf{G_{t+1}}$, $\mathbf{W_{t+1}}$, and the new core tensor $\mathcal{Y}_{\mathbf{t+1}}$.

A general problem definition. It is easy to extend the formulation from 3 to M dimensions and give a general definition.

Definition 1.1 (Flexible Evolutionary Multi-faceted Analysis)

Initialization: *Given the first* M*-way tensor* $\mathcal{X}_\mathbf{1} \in \mathbb{R}^{n^{(1)}\times\ldots\times n^{(M)}}$ *and the constraints* $\mathbf{L^{(m)}}|_{m=1}^{M} \in \mathbb{R}^{n^{(m)}\times n^{(m)}}$*, find the first projection matrices* $\mathbf{A_1^{(m)}}|_{m=1}^{M} \in \mathbb{R}^{n^{(m)}\times r^{(m)}}$ *and the first core tensor* $\mathcal{Y}_\mathbf{1} \in \mathbb{R}^{r^{(1)}\times\ldots\times r^{(M)}}$. **Evolutionary analysis:** *At time* t $(1 \le t < T)$*, given the tensor* $\mathcal{X}_\mathbf{t} \in \mathbb{R}^{n^{(1)}\times\ldots\times n^{(M)}}$*, the increment* $\Delta\mathcal{X}_\mathbf{t}$*, the old projection matrices* $\mathbf{A_t^{(m)}}|_{m=1}^{M}$*, and the constraints* $\mathbf{L^{(m)}}|_{m=1}^{M}$*, find the new projection matrices* $\mathbf{A_{t+1}^{(m)}}|_{m=1}^{M}$ *and the new core tensor* $\mathcal{Y}_\mathbf{t+1}$.

Challenges. The first challenge is **high sparsity**. Multi-faceted data in real applications is often very sparse. In user-location-word case, for example, users cannot appear in many geo-locations or all the words in the dataset. The problem is even disastrous when adding temporal dimension to the multi-faceted behavioral information. The second one is **high complexity**. Considering the dynamic characteristic, new multi-faceted human behaviors continuously appear over time. The continuously generated data of high volume, high dimension and high sparsity pose great challenges for modeling and analysis due to high computational complexity. The issue of fast processing increments is still critical for modeling and predicting human behavior.

1.2.2 Flexible evolutionary multi-faceted analysis

Framework design. Flexible Evolutionary Multi-faceted Analysis (FEMA) is designed based on a dynamic scheme of tensor factorization for temporal multi-faceted behavior pattern mining and prediction (see Figure 1.5). First, flexible regularizers are imposed to alleviate the problems brought by high sparsity. Second, in order to efficiently decompose high-order tensor sequences, instead of re-decomposing the updated a matricized tensor, it uses approximation algorithms to factorize the new tensor with sparse increments, where the bound of approximation loss is theoretically proved.

Initialization. We denote by $\mu^{(m)}$ the weight of the mode-m Laplacian matrix $\mathbf{L^{(m)}}$. The covariance matrix of the m-th mode at time $t = 1$ is

$$\mathbf{C_1^{(m)}} = \mathbf{X_1^{(m)}}\mathbf{X_1^{(m)T}} + \mu^{(m)}\mathbf{L^{(m)}} \tag{1.15}$$

where $\mathbf{X_1^{(m)}} \in \mathbb{R}^{n^{(m)}\times\prod_{i\neq m} n^{(i)}}$ is the mode-m matricizing of the tensor $\mathcal{X}_\mathbf{1}$. The projection matrices $\mathbf{A_1^{(m)}}|_{m=1}^{M}$ can be computed by diagonalization: they are the top $r^{(m)}$ eigenvectors of the covariance matrix $\mathbf{C_1^{(m)}}|_{m=1}^{M}$.

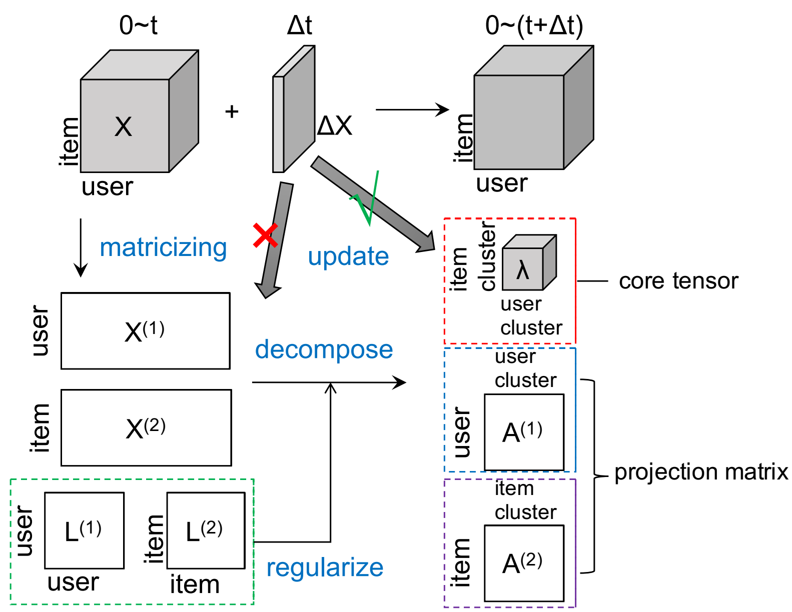


Figure 1.5 Flexible evolutionary multi-faceted analysis.

Evolutionary analysis. Next we introduce an efficient technique based on tensor perturbation to adjust the projection matrices according to changes of the tensor. We denote by $\mathbf{X_t^{(m)}} \in \mathbb{R}^{n^{(m)} \times \prod_{i \neq m} n^{(i)}}$ the mode-m matricizing of the tensor $\mathcal{X}_\mathbf{t}$. We define the covariance matrix $\mathbf{C_t^{(m)}} = \mathbf{X_t^{(m)}} \mathbf{X_t^{(m)}}^\top + \mu^{(m)} \mathbf{L^{(m)}}$ and define $(\lambda_{t,i}^{(m)}, \mathbf{a_{t,i}^{(m)}})$ as one eigenvalue-eigenvector pair of the matrix $\mathbf{C_t^{(m)}}$. The vector $\mathbf{a_{t,i}^{(m)}}$ is exactly the i-th column of the projection matrix $\mathbf{A_t^{(m)}}$. Then we can rewrite $(\lambda_{t+1,i}^{(m)}, \mathbf{a_{t+1,i}^{(m)}})$ as

$$
\begin{aligned}
\lambda_{t+1,i}^{(m)} &= \lambda_{t,i}^{(m)} + \Delta\lambda_{t,i}^{(m)} && (1.16)\\
\mathbf{a_{t+1,i}^{(m)}} &= \mathbf{a_{t,i}^{(m)}} + \Delta\mathbf{a_{t,i}^{(m)}} && (1.17)
\end{aligned}
$$

To simplify the denotions, we omit "t" in the terms and equations when it is unnecessary. Thus we can obtain

$$
\begin{aligned}
& [(\mathbf{X^{(m)}} + \Delta\mathbf{X^{(m)}})(\mathbf{X^{(m)}} + \Delta\mathbf{X^{(m)}})^\top + \mu^{(m)}\mathbf{L^{(m)}}] \cdot (\mathbf{a_i^{(m)}} + \Delta\mathbf{a_i^{(m)}}) \\
= \; & (\lambda_i^{(m)} + \Delta\lambda_i^{(m)})(\mathbf{a_i^{(m)}} + \Delta\mathbf{a_i^{(m)}}) \qquad (1.18)
\end{aligned}
$$

Now the key questions are how to compute changes to the eigenvalue $\Delta\lambda_i^{(m)}$ and eigenvector $\Delta\mathbf{a_i^{(m)}}$, respectively.

If we concentrate on first-order approximation when expanding Eq.(1.18), we assume all high order perturbation terms are negligible. By further using the fact that $(\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top + \mu^{(m)}\mathbf{L^{(m)}})\mathbf{a_i^{(m)}} = \lambda_i^{(m)}\mathbf{a_i^{(m)}}$, we can obtain

$$\begin{aligned} & \mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top \Delta\mathbf{a_i^{(m)}} + (\mathbf{X^{(m)}}\Delta\mathbf{X^{(m)}}^\top + \Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top)\mathbf{a_i^{(m)}} + \mu^{(m)}\mathbf{L^{(m)}}\Delta\mathbf{a_i^{(m)}} \\ = \ & \lambda_i^{(m)}\Delta\mathbf{a_i^{(m)}} + \Delta\lambda_i^{(m)}\mathbf{a_i^{(m)}} \end{aligned} \tag{1.19}$$

Now multiplying both sides of Eq.(1.19) with $\mathbf{a_i^{(m)}}^\top$ and because of the symmetry of $\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top$ and $\mathbf{L^{(m)}}$, we get

$$\Delta\lambda_i^{(m)} = \mathbf{a_i^{(m)}}^\top(\mathbf{X^{(m)}}\Delta\mathbf{X^{(m)}}^\top + \Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top)\mathbf{a_i^{(m)}} \tag{1.20}$$

Since the eigenvectors are orthogonal to each other, we assume that the change of the eigenvector $\Delta\mathbf{a_i^{(m)}}$ is in the subspace spanned by those original eigenvectors, i.e.,

$$\Delta\mathbf{a_i^{(m)}} \approx \sum_{j=1}^{r^{(m)}} \alpha_{ij}\mathbf{a_j^{(m)}} \tag{1.21}$$

where $-\alpha_{ij}''$ are small constants to be determined. Bringing Eq.(1.21) into Eq.(1.19), we obtain

$$\begin{aligned} & (\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top + \mu^{(m)}\mathbf{L^{(m)}})\textstyle\sum_{j=1}^{r^{(m)}} \alpha_{ij}\mathbf{a_j^{(m)}} + (\mathbf{X^{(m)}}\Delta\mathbf{X^{(m)}}^\top + \Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top)\mathbf{a_i^{(m)}} \\ = \ & \lambda_i^{(m)}\textstyle\sum_{j=1}^{r^{(m)}} \alpha_{ij}\mathbf{a_j^{(m)}} + \Delta\lambda_i^{(m)}\mathbf{a_i^{(m)}} \end{aligned} \tag{1.22}$$

which is equivalent to

$$\begin{aligned} & \sum_{j=1}^{r^{(m)}} \lambda_j^{(m)}\alpha_{ij}\mathbf{a_j^{(m)}} + \mathbf{X^{(m)}}\Delta\mathbf{X^{(m)}}^\top\mathbf{a_i^{(m)}} + \Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top\mathbf{a_i^{(m)}} \\ = \ & \lambda_i^{(m)}\sum_{j=1}^{r^{(m)}} \alpha_{ij}\mathbf{a_j^{(m)}} + \Delta\lambda_i^{(m)}\mathbf{a_i^{(m)}} \end{aligned} \tag{1.23}$$

Multiplying $\mathbf{a_k^{(m)}}^\top$ $(k \neq i)$ on both sides of the above equation, we get

$$\lambda_k^{(m)}\alpha_{ik} + \mathbf{a_k^{(m)}}^\top\mathbf{X^{(m)}}\Delta\mathbf{X^{(m)}}^\top\mathbf{a_i^{(m)}} + \mathbf{a_k^{(m)}}^\top\Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top\mathbf{a_i^{(m)}} = \lambda_i^{(m)}\alpha_{ik} \tag{1.24}$$

Therefore,

$$\alpha_{ik} = \frac{\mathbf{a_k^{(m)}}^\top (\mathbf{X^{(m)}} \Delta \mathbf{X^{(m)}}^\top + \Delta \mathbf{X^{(m)}} \mathbf{X^{(m)}}^\top) \mathbf{a_i^{(m)}}}{\lambda_i^{(m)} - \lambda_k^{(m)}} \tag{1.25}$$

To get α_{ii}, we use the fact that

$$(\mathbf{a_i^{(m)}} + \Delta\mathbf{a_i^{(m)}})^\top (\mathbf{a_i^{(m)}} + \Delta\mathbf{a_i^{(m)}}) = 1 \Longleftrightarrow 1 + 2\mathbf{a_i^{(m)}}^\top \Delta\mathbf{a_i^{(m)}} + O(\|\Delta\mathbf{a_i^{(m)}}\|^2) = 1 \tag{1.26}$$

Discarding the high order term, and bringing in Eq.(1.21), we get $\alpha_{ii} = 0$. Therefore,

$$\Delta\mathbf{a_i^{(m)}} = \sum_{j \neq i} \frac{\mathbf{a_j^{(m)}}^\top (\mathbf{X^{(m)}} \Delta \mathbf{X^{(m)}}^\top + \Delta \mathbf{X^{(m)}} \mathbf{X^{(m)}}^\top) \mathbf{a_i^{(m)}}}{\lambda_i^{(m)} - \lambda_j^{(m)}} \mathbf{a_j^{(m)}} \tag{1.27}$$

Note that the constraints $\mathbf{L^{(m)}}$ do not appear in the eigenvalue and eigenvector updating functions Eq.(1.20) and Eq.(1.27). Note that the constraints have to be learnt only *once*.

Computational complexity. For the m-th mode, we define $D^{(m)}$ as the number of features of each point on the m-th dimension. Since the tensors are usually extremely sparse, we know $D^{(m)} \leq E \ll \prod_{m' \neq m} n^{(m')}$, where E is the number of non-zero entries in the tensors. In order to compute the increment on the eigenvalue and eigenvector using Eq.(1.20) and Eq.(1.27) for the m-th mode, we need to compute $\mathbf{v_i^{(m)}}$, which requires $O(n^{(m)} D^{(m)})$ time. As $\Delta\mathbf{X^{(m)}}$ is very sparse, $\Delta\mathbf{X^{(m)}}\mathbf{v_i^{(m)}}$ only requires constant time $O(D^{(m)})$. Therefore, for computing $\Delta\lambda_i^{(m)}$ and $\Delta\mathbf{a_i^{(m)}}$, we need $O(r^{(m)} n^{(m)} D^{(m)} + r^{(m)} D^{(m)})$ time, and updating eigenvalues and eigenvectors for T times requires $O(T \sum_{m=1}^{M} r^{(m)} (n^{(m)} + 1) D^{(m)})$ time. In comparison, if we redo the eigenvalue decomposition on $\mathcal{X}_{\mathbf{t+1}}$, it costs $O(T \sum_{m=1}^{M} (D^{(m)} (n^{(m)})^2 + (n^{(m)})^3))$ time, which is much higher.

Approximation quality. We now present two theorems that bound the magnitude of $\Delta\lambda_i^{(m)}$ and $\Delta\mathbf{a_i^{(m)}}$. Both theorems confirm our intuition that the magnitude of $\Delta\lambda_i^{(m)}$ and $\Delta\mathbf{a_i^{(m)}}$ is directly related to the norm of $\Delta\mathbf{X^{(m)}}$. Also since the higher order terms are ignored in the approximation, FEMA only works when those terms are relatively small.

Theorem 1.1 *The magnitude of the variation on the eigenvalue, i.e.,* $|\Delta\lambda_i^{(m)}|$, $(\forall i = 1, \ldots, r^{(m)})$, *satisfies the following inequality*

$$|\Delta\lambda_i^{(m)}| \leq 2(\lambda^{\max}_{\mathbf{X^{(m)}}^\top \mathbf{X^{(m)}}})^{\frac{1}{2}} \|\Delta\mathbf{X^{(m)}}\|_2 \tag{1.28}$$

where $\lambda^{\max}_{\mathbf{X^{(m)}}^\top \mathbf{X^{(m)}}}$ *is the maximum eigenvalue of the data inner product matrix* $\mathbf{X^{(m)}}^\top \mathbf{X^{(m)}}$, $\|\Delta\mathbf{X^{(m)}}\|_2$ *is the 2-norm of* $\Delta\mathbf{X^{(m)}}$.

Proof: see the solution to exercise 1.2.

Theorem 1.2 *The magnitude of the variation on the eigenvector, i.e.,* $|\Delta\mathbf{a_i^{(m)}}|$, $(\forall i = 1, \ldots, r^{(m)})$, *satisfies the following inequality*

$$|\Delta\mathbf{a_i^{(m)}}| \leq 2\|\Delta\mathbf{X^{(m)}}\|_2 \sum_{j \neq i} \frac{(\lambda^{\max}_{\mathbf{X^{(m)}}^\top \mathbf{X^{(m)}}})^{\frac{1}{2}}}{|\lambda_i^{(m)} - \lambda_j^{(m)}|} \tag{1.29}$$

where $\lambda^{\max}_{\mathbf{X^{(m)}}^\top \mathbf{X^{(m)}}}$ *is the maximum eigenvalue of the data inner product matrix* $\mathbf{X^{(m)}}^\top \mathbf{X^{(m)}}$, $\|\Delta\mathbf{X^{(m)}}\|_2$ *is the 2-norm of* $\Delta\mathbf{X^{(m)}}$.

Proof: see the solution to exercise 1.2.

1.2.3 Conclusions

This section introduced a tensor factorization based framework for temporal multi-faceted behavior prediction and behavioral pattern mining. The model used flexible regularizers to alleviate the sparsity problem and gave approximation algorithms to efficiently process the increments with a theoretical guarantee. Extensive experiments performed on real world datasets demonstrate that the framework was effective and efficient in behavior prediction tasks.

1.3 MINING CROSS-DOMAIN BEHAVIOR FOR KNOWLEDGE TRANSFER

Social networks allow users to create and adopt different types of items - not only web posts but also user labels, images and videos. Traditional web post recommendation approaches suffer from data sparsity (i.e., limited interaction between users and web posts) and the issue of cold start (i.e., giving recommendations to new users who have not yet

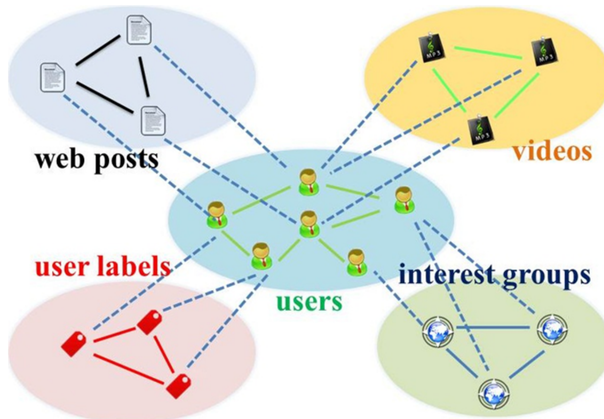


Figure 1.6 Star-structured graph for cross-domain behavior modeling.

created any web posts). The social connections and multiple item domains provide an unprecedented opportunity to alleviate these issues in real applications. This section will introduce a general framework for cross-domain behavior modeling [12, 9].

1.3.1 Cross-domain behavior modeling

Users' characteristics relate both to social connections and to different user-item interactions. For example, users read web posts created by their community and may adopt similar user labels to their friends. Therefore, an effective social recommendation approach should acknowledge (1) social tie strength (henceforth, tie strength) between users and (2) different user-item interactions. How to incorporate a social domain and auxiliary item domains (*e.g.*, user labels and images) into a unified framework?

Note that multiple item domains reflect users' intrinsic preferences and tend to be tightly connected among a massive number of users. Figure 1.6 shows a new representation of social networks and propose a star-structured graph, where the social domain is at the center and is connected to the surrounding item domains.

The value of the *cross-domain link*[1] weight represents how often a given user adopts a given item, while the value of the *within-domain*

[1] Cross-domain links are user-item links (item adoptions), i.e., links between the social domain and the item domains.

link[2] weight in the social domain represents the tie strength between users. Users are more likely to have stronger ties if they share similar characteristics. Cross-domain links reflect users' characteristics in different ways. For example, a cross-domain link from a user to a web post about iPhones shows his/her short-term interest in iPhones, and a cross-domain link from him/her to a label "iPhone Fan" implies his/her long-term interest in iPhones. A basic assumption is that the more auxiliary knowledge we have, the more we know about the users, thereby enabling more accurate estimates of tie strength. When a user and his/her friend have many common user labels, we assume a greater tie strength and expect them to be more similar in terms of their web post adoption behaviors. Even if the web post domain is extremely sparse, we may still produce effective recommendations by transferring auxiliary knowledge from other item domains through the social domain.

General idea. The Hybrid Random Walk (HRW) method can transfer knowledge from auxiliary item domains according to a star-structured configuration to improve social recommendations in a target domain. HRW estimates weights for (1) links between user nodes within the social domain, and (2) links between user nodes in the social domain and item nodes in the item domain. The weights respectively represent (1) tie strength between users and (2) the probability of a user adopting or rejecting an item. HRW integrates knowledge from multiple relational domains and alleviates sparsity and cold-start issues.

1.3.2 Hybrid random walk algorithm

A real-world example of a second-order hybrid star-structured graph is presented in Figure 1.7. It differs from traditional star-structured graphs that do not include entity relationships within each domain. This hybrid graph considers both within-domain and cross-domain entity relationships. Table 1.1 summarizes the symbols used in this section to denote the five subgraphs in Figure 1.7:

1. $\mathcal{G}^{(\mathcal{U})} = \{\mathcal{U}, \mathcal{E}^{(\mathcal{U})}\}$, where $\mathcal{E}^{(\mathcal{U})}$ represents the edge set linking the nodes in $\mathcal{U}$;
2. $\mathcal{G}^{(\mathcal{P})} = \{\mathcal{P}, \mathcal{E}^{(\mathcal{P})}\}$, where $\mathcal{E}^{(\mathcal{P})}$ represents the edge set linking the nodes in $\mathcal{P}$;

[2] Within-domain links are user-user links in the social domain (social connections) and the item-item links in each item domain.

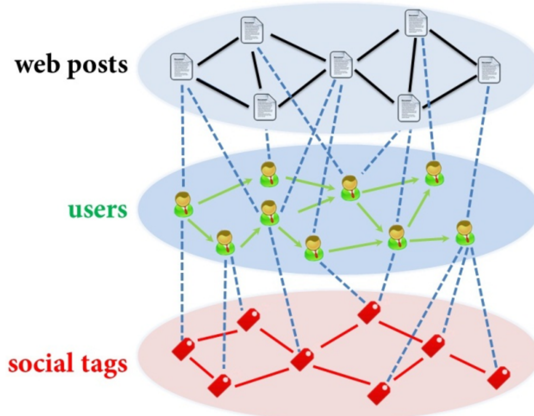


Figure 1.7 Hybrid random walk on a second-order star-structured graph.

Table 1.1 Symbols for hybrid random walk algorithm

Symbol	Description
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
d_{ij}	The j-th item in the i-th domain
$\mathcal{D}_i = \{d_{i1}, d_{i2}, \cdots, d_{i\|\mathcal{D}_i\|}\}$	The set of items in the i-th domain
$\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N\}$	The set of item domains

3. $\mathcal{G}^{(\mathcal{T})} = \{\mathcal{T}, \mathcal{E}^{(\mathcal{T})}\}$, where $\mathcal{E}^{(\mathcal{T})}$ represents the edge set linking the nodes in $\mathcal{T}$;
4. $\mathcal{G}^{(\mathcal{UP})} = \{\mathcal{U} \bigcup \mathcal{P}, \mathcal{E}^{(\mathcal{UP})}\}$, where $\mathcal{E}^{(\mathcal{UP})}$ represents the edges linking the nodes in $\mathcal{U}$ and $\mathcal{P}$;
5. $\mathcal{G}^{(\mathcal{UT})} = \{\mathcal{U} \bigcup \mathcal{T}, \mathcal{E}^{(\mathcal{UT})}\}$, where $\mathcal{E}^{(\mathcal{UT})}$ represents the edges linking the nodes in $\mathcal{U}$ and $\mathcal{T}$.

To conceptualize user relationships in $\mathcal{G}^{(\mathcal{U})}$, consider the relevance from user u_i to u_j as

$$w_{ij}^{(\mathcal{U})} = \begin{cases} 1 & \text{if user } u_i \text{ is a friend of } u_j, \\ 0 & \text{otherwise.} \end{cases} \tag{1.30}$$

To compute web-post relationships in $\mathcal{P}$, HRW uses a Term Frequency-

Inverse Document Frequency (TF-IDF) representation vector for each post $\mathrm{b}_i = [b_{i1}, \cdots, b_{ik}, \cdots, b_{iK}]^\top$ in matrix $\mathbf{B}$ (where K is the size of vocabulary), and then measure the semantic similarity between post b_i and b_j as

$$w_{ij}^{(\mathcal{P})} = \frac{\sum_k b_{ik} b_{jk}}{\sqrt{\sum_k {b_{ik}}^2}\sqrt{\sum_k {b_{jk}}^2}}. \tag{1.31}$$

For user labels, HRW measures their similarity using the Jaccard similarity. 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 relationship is computed as

$$w_{ij}^{(\mathcal{T})} = \frac{c_{ij}}{c_i + c_j - c_{ij}}. \tag{1.32}$$

Thus, three similarity matrices $\mathbf{W}^{(\mathcal{U})} = \{w_{ij}^{(\mathcal{U})}\}$, $\mathbf{W}^{(\mathcal{P})} = \{w_{ij}^{(\mathcal{P})}\}$ and $\mathbf{W}^{(\mathcal{T})} = \{w_{ij}^{(\mathcal{T})}\}$ have been constructed to encode edge weights for three within-domain subgraphs.

Further, we have 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, 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 $e_{ij}^{(\mathcal{UT})}$. Their weights are determined as follows.

$$w_{ij}^{(\mathcal{UP})^+} = \begin{cases} 1 & \text{if user } u_i \text{ adopts web post } \rho_j, \\ 0 & \text{otherwise;} \end{cases} \tag{1.33}$$

$$w_{ij}^{(\mathcal{UP})^-} = \begin{cases} 1 & \text{if user } u_i \text{ rejects web post } \rho_j, \\ 0 & \text{otherwise;} \end{cases} \tag{1.34}$$

$$w_{ij}^{(\mathcal{UT})^+} = \begin{cases} 1 & \text{if user } u_i \text{ adopts web post } t_j, \\ 0 & \text{otherwise.} \end{cases} \tag{1.35}$$

Thus, we obtain the three weight matrices $\mathbf{W}^{(\mathcal{UP})^+}=\{w_{ij}^{(\mathcal{UP})^+}\}$, $\mathbf{W}^{(\mathcal{UP})^-}=\{w_{ij}^{(\mathcal{UP})^-}\}$ and $\mathbf{W}^{(\mathcal{UT})^+}=\{w_{ij}^{(\mathcal{UT})^+}\}$.

Now we can derive a random walk algorithm [29, 6, 8] to predict missing links on $\mathcal{G}^{(\mathcal{UP})}$ and $\mathcal{G}^{(\mathcal{UT})}$, which includes both within-domain and cross-domain random walks. For $\mathcal{G}^{(\mathcal{U})}$, $\mathcal{G}^{(\mathcal{P})}$ and $\mathcal{G}^{(\mathcal{T})}$, we derive steady-state distributions, indicating the intrinsic relevance among users, posts and labels. For a standard random walk model, a walker starts from the i-th vertex and iteratively jumps to other vertices with transition probabilities $\mathbf{p}_i = \{p_{i1}, \cdots, p_{in}\}$. After reaching the steady state, the probability of the 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:

$$\mathbf{P}^{(\mathcal{U})} = (\mathbf{D}^{(\mathcal{U})})^{-1}\mathbf{W}^{(\mathcal{U})}, \tag{1.36}$$
$$\mathbf{P}^{(\mathcal{P})} = (\mathbf{D}^{(\mathcal{P})})^{-1}\mathbf{W}^{(\mathcal{P})}, \tag{1.37}$$
$$\mathbf{P}^{(\mathcal{T})} = (\mathbf{D}^{(\mathcal{T})})^{-1}\mathbf{W}^{(\mathcal{T})}, \tag{1.38}$$

where we denote the degree matrices of cross-domain links by $\mathbf{D}^{(\mathcal{UP})^+}$, $\mathbf{D}^{(\mathcal{UP})^-}$, and $\mathbf{D}^{(\mathcal{UT})^+}$. The final steady-state probability matrices can be obtained by iterating the following updates:

$$\mathbf{R}^{(\mathcal{U})}(t+1) = \alpha\mathbf{P}^{(\mathcal{U})}\mathbf{R}^{(\mathcal{U})}(t) + (1-\alpha)\mathbf{I}, \tag{1.39}$$
$$\mathbf{R}^{(\mathcal{P})}(t+1) = \beta\mathbf{P}^{(\mathcal{P})}\mathbf{R}^{(\mathcal{P})}(t) + (1-\beta)\mathbf{I}, \tag{1.40}$$
$$\mathbf{R}^{(\mathcal{T})}(t+1) = \gamma\mathbf{P}^{(\mathcal{T})}\mathbf{R}^{(\mathcal{T})}(t) + (1-\gamma)\mathbf{I}, \tag{1.41}$$

where $\mathbf{R}^{(\mathcal{U})}(t)$, $\mathbf{R}^{(\mathcal{P})}(t)$, $\mathbf{R}^{(\mathcal{T})}(t)$, $\mathbf{R}^{(\mathcal{U})}(t+1)$, $\mathbf{R}^{(\mathcal{P})}(t+1)$ and $\mathbf{R}^{(\mathcal{T})}(t+1)$ are the state probability matrices at time t and $t+1$; and $0 \leq \alpha, \beta, \gamma \leq 1$ 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 steady state matrices when $t \to \infty$:

$$\mathbf{R}^{(\mathcal{U})} = (1-\alpha)\left(\mathbf{I} - \alpha\mathbf{P}^{(\mathcal{U})}\right)^{-1}, \tag{1.42}$$
$$\mathbf{R}^{(\mathcal{P})} = (1-\beta)\left(\mathbf{I} - \beta\mathbf{P}^{(\mathcal{P})}\right)^{-1}, \tag{1.43}$$
$$\mathbf{R}^{(\mathcal{T})} = (1-\gamma)\left(\mathbf{I} - \gamma\mathbf{P}^{(\mathcal{T})}\right)^{-1}. \tag{1.44}$$

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

$$\mathbf{P}^{(\mathcal{UP})^+} = \left(\mathbf{D}^{(\mathcal{UP})^+}\right)^{-1}\mathbf{W}^{(\mathcal{UP})^+}, \tag{1.45}$$
$$\mathbf{P}^{(\mathcal{UP})^-} = \left(\mathbf{D}^{(\mathcal{UP})^-}\right)^{-1}\mathbf{W}^{(\mathcal{UP})^-}, \tag{1.46}$$
$$\mathbf{P}^{(\mathcal{UT})^+} = \left(\mathbf{D}^{(\mathcal{UT})^+}\right)^{-1}\mathbf{W}^{(\mathcal{UT})^+}, \tag{1.47}$$

where elements $p_{ij}^{(\mathcal{UP})^+}$ and $p_{ij}^{(\mathcal{UP})^-}$ represent the transition probability that user u_i will adopt/ignore post p_j; and $p_{ij}^{(\mathcal{UT})^+}$ represents the transition probability that user u_i will adopt label t_j. Now, we simultaneously determine relevance scores $\mathbf{R}^{(\mathcal{U})} = \{r_{ij}^{(\mathcal{U})}\}$ between each pair

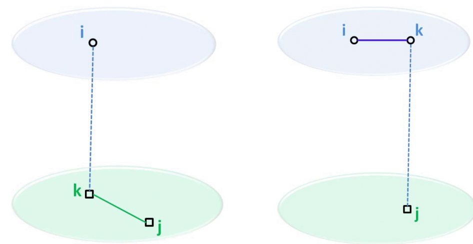


Figure 1.8 Transition routes we consider when updating cross-domain transition probability matrix.

of users, which finally reflects the tie strength on the real 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

$$p_{ij}^{(\mathcal{UP})^+} = \delta \sum_{u_k \in \mathcal{U}} r_{ik}^{(\mathcal{U})} p_{kj}^{(\mathcal{UP})^+} + (1-\delta) \sum_{p_k \in \mathcal{P}} p_{ik}^{(\mathcal{UP})^+} r_{kj}^{(\mathcal{P})}, \tag{1.48}$$

$$p_{ij}^{(\mathcal{UP})^-} = \delta \sum_{u_k \in \mathcal{U}} r_{ik}^{(\mathcal{U})} p_{kj}^{(\mathcal{UP})^-} + (1-\delta) \sum_{p_k \in \mathcal{P}} p_{ik}^{(\mathcal{UP})^-} r_{kj}^{(\mathcal{P})}, \tag{1.49}$$

$$p_{ij}^{(\mathcal{UT})^+} = \eta \sum_{u_k \in \mathcal{U}} r_{ik}^{(\mathcal{U})} p_{kj}^{(\mathcal{UT})^+} + (1-\eta) \sum_{t_k \in \mathcal{T}} p_{ik}^{(\mathcal{UT})^+} r_{kj}^{(\mathcal{T})}, \tag{1.50}$$

$$\begin{aligned} r_{ij}^{(\mathcal{U})} = {} & \tau^{(\mathcal{P})}(\mu \sum_{p_k \in \mathcal{P}} p_{ik}^{(\mathcal{UP})^+} p_{jk}^{(\mathcal{UP})^+} + (1-\mu) \sum_{p_k \in \mathcal{P}} p_{ik}^{(\mathcal{UP})^-} p_{jk}^{(\mathcal{UP})^-}) \\ & + \tau^{(\mathcal{T})} \sum_{t_k \in \mathcal{T}} p_{ik}^{(\mathcal{UT})^+} p_{jk}^{(\mathcal{UT})^+} + \tau^{(\mathcal{U})} \sum_{u_k \in \mathcal{U}} r_{ik}^{(\mathcal{U})} r_{kj}^{(\mathcal{U})}, \end{aligned} \tag{1.51}$$

where $0 \leq \delta, \eta, \mu, \tau^{(\mathcal{P})}, \tau^{(\mathcal{T})}, \tau^{(\mathcal{U})} \leq 1$ are the parameters for trading off different transition routes. Note that for the update of cross-domain transition probability matrices (Eq.(1.48) to Eq.(1.50)), we consider two types of routes shown in Figure 1.8. HRW also assumes that the update of the cross-domain probability matrices will affect the within-domain probability matrix of the user subgraph.

On the other hand, the update of cross-domain transition probability affects only the within-domain transition probabilities of the user graph, because the user tie strength is influenced by (1) common posts, (2) common labels, and (3) social relationships. The cross-domain links

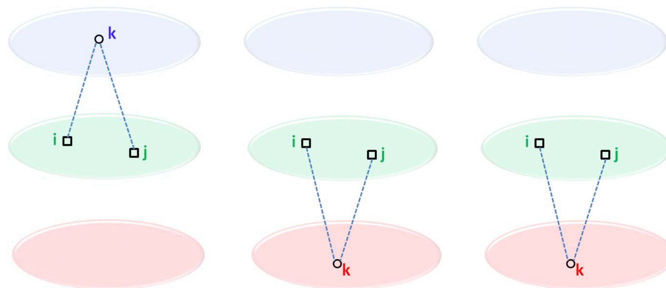


Figure 1.9 Transition routes when updating within-domain transition probability matrix on the user subgraph.

(adoption behaviors) do not affect the within-domain transition probability of other item domains. The rationale is that the transition probability of items (posts, labels, videos, etc.) should be derived by their semantic similarity, which would not be changed by the users who adopt them. Therefore, the HRW method updates cross-domain links between the user and all types of items as well as within-domain links between users (not items). The updating rule Eq.(1.51) considers three routes shown in Figure 1.9.

We can further give matrix formulations for the update of transition probability from time t to $t+1$.

$$\begin{aligned}
\mathbf{P}^{(\mathcal{UP})^+}(t+1) &= \delta\mathbf{R}^{(\mathcal{U})}(t)\mathbf{P}^{(\mathcal{UP})^+}(t)+(1-\delta)\mathbf{P}^{(\mathcal{UP})^+}(t)\mathbf{R}^{(\mathcal{P})}, &(1.52)\\
\mathbf{P}^{(\mathcal{UP})^-}(t+1) &= \delta\mathbf{R}^{(\mathcal{U})}(t)\mathbf{P}^{(\mathcal{UP})^-}(t)+(1-\delta)\mathbf{P}^{(\mathcal{UP})^-}(t)\mathbf{R}^{(\mathcal{P})}, &(1.53)\\
\mathbf{P}^{(\mathcal{UT})^+}(t+1) &= \eta\mathbf{R}^{(\mathcal{U})}(t)\mathbf{P}^{(\mathcal{UT})^+}(t)+(1-\eta)\mathbf{P}^{(\mathcal{UT})^+}(t)\mathbf{R}^{(\mathcal{T})}, &(1.54)\\
\mathbf{R}^{(\mathcal{U})}(t+1) &= \tau^{(\mathcal{P})}(\mu\mathbf{P}^{(\mathcal{UP})^+}(t)\mathbf{P}^{(\mathcal{UP})^+}(t)^T \\
&\quad +(1-\mu)\mathbf{P}^{(\mathcal{UP})^-}(t)\mathbf{P}^{(\mathcal{UP})^-}(t)^T) \\
&\quad +\tau^{(\mathcal{T})}\mathbf{P}^{(\mathcal{UT})^+}(t)\mathbf{P}^{(\mathcal{UT})^+}(t)^T \\
&\quad +\tau^{(\mathcal{U})}\mathbf{R}^{(\mathcal{U})}(t)\mathbf{R}^{(\mathcal{U})}(t)^T. &(1.55)
\end{aligned}$$

With graphs $\mathcal{G}^{(\mathcal{U})}$, $\mathcal{G}^{(\mathcal{UP})}$, and $\mathcal{G}^{(\mathcal{UT})}$, the corresponding transition matrices $\mathbf{R}^{(\mathcal{U})}$, $\mathbf{P}^{(\mathcal{UP})^+}$, $\mathbf{P}^{(\mathcal{UP})^-}$, and $\mathbf{P}^{(\mathcal{UT})^+}$ are computed for the next random walk step.

Complexity. The space complexity of the HRW 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 m, n, and l are the number of users, posts, and labels, respectively, and T is the number of iterations. Since the matrices are usually sparse and $m, n \gg l$, the space complexity is $O((m+n)^2)$ and the time complexity is $O((m+n)ET)$, where E is the number of links between users and posts.

1.3.3 Conclusions

This section addressed the problems of data sparsity and cold start in social recommendation. It reconsidered the problem from the transfer learning perspective and alleviated the data sparsity problem in a target domain by transferring knowledge from other auxiliary social relational domains. By considering the special structures of multiple relational domains in social networks, an innovative random walk method has been derived on a star-structured graph, which is a general method to incorporate complex and heterogeneous link structures. Extensive experiments on a large real-world social network dataset showed that this method greatly boosts the social recommendation performance. In particular, it gained improvement in web-post recommendation by transferring knowledge from the user-label domain for the user tie strength updating process, compared with the recommendation methods, which only use information from the web-post domain. In addition, by using only 27.6% of the available information in the target domain, the method achieved comparable performance with methods that use all available information in the target domain without transfer learning. The proposed method and insightful experiments indicate a promising and general way to solve the data sparsity problem.

1.4 SUMMARY

This chapter introduces novel techniques of mining user behaviors for three social applications: (1) uncovering the behavioral mechanism in social networks and modeling the social contexts for social recommendation, (2) modeling spatio-temporal contexts for behavior prediction, and (3) mining cross-domain behavior to address the sparsity challenge. Section 1.1 demonstrates the important roles of preference and influence in users' behavioral intention of information adoption and rejection. The next section proposes a framework of flexible evolutionary multi-faceted analysis to model the multi-aspect and dynamic behavioral patterns. The third section proposes a hybrid random walk method that uses the social relationship as bridge to connect multiple domains in social networks. The procedure of knowledge transfer can alleviate the data sparsity issue in a single domain. We sincerely hope these three parts of work would shed insights on mining and modeling large behavioral data in social networks.

EXERCISES

1.1 In Section 1.1, use statistical measures to prove there is NOT a close relationship between scores on the two contextual factors: *individual preference* and *interpersonal influence*.

Solution :

We can calculate the correlation between the two factors in recommendation cases. If we use P and I for each user to denote preference and influence of her adopted items. The Pearson correlation is defined as

$$\rho_{P,I} = \frac{cov(P,I)}{\sigma_P \sigma_I} = \frac{E[(P-\mu_P)(I-\mu_I)]}{\sigma_P \sigma_I}, \qquad \text{(P1.1)}$$

where $\mu_.$ is the mean value, $\sigma_.$ denotes the standard deviation and $cov(.,.)$ denotes the covariance. The correlation is 1 or -1 in the case of perfect positive or negative linear relationship, and zero if preference and influence are uncorrelated. We can demonstrate the existence and significance of the two factors when the absolute correlation value is small.

1.2 In Section 1.2, prove Theorem 1.1 and Theorem 1.2 for the guarantee of FEMA's effectiveness.

Solution :

(1) Proof for Theorem 1.1: According to Eq.(1.20), we have

$$|\Delta\lambda_i^{(m)}| = |\mathbf{a_i^{(m)}}^\top (\mathbf{X^{(m)}}\Delta\mathbf{X^{(m)}}^\top + \Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top)\mathbf{a_i^{(m)}}|. \qquad \text{(P1.2)}$$

By Cauchy-Schwarz inequality,

$$\begin{aligned} & |\mathbf{a_i^{(m)}}^\top (\mathbf{X^{(m)}}\Delta\mathbf{X^{(m)}}^\top + \Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top)\mathbf{a_i^{(m)}}| \\ \leq \;& 2\|\Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top \mathbf{a_i^{(m)}}\|_2 \|\mathbf{a_i^{(m)}}\|_2 \\ = \;& 2\|\Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top \mathbf{a_i^{(m)}}\|_2, \qquad \text{(P1.3)} \end{aligned}$$

where in the first step we use the symmetry of $\mathbf{X^{(m)}}\Delta\mathbf{X^{(m)}}^\top +$

$\Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top$ and in the second step we use the fact that $\|\mathbf{a_i^{(m)}}\| = 1$. By the definition of matrix 2-norm, we have that

$$\|\Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top\|_2 = \sup_{\|\mathbf{w}\|_2=1} \|\Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top\mathbf{w}\|_2. \qquad \text{(P1.4)}$$

Therefore

$$\begin{aligned} & |\mathbf{a_i^{(m)}}^\top(\mathbf{X^{(m)}}\Delta\mathbf{X^{(m)}}^\top + \Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top)\mathbf{a_i^{(m)}}| \\ \leq\ & 2\|\Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top\|_2 \leq 2\|\mathbf{X^{(m)}}\|_2\|\Delta\mathbf{X^{(m)}}\|_2 \\ =\ & 2(\lambda^{\max}_{\mathbf{X^{(m)}}^\top\mathbf{X^{(m)}}})^{\frac{1}{2}}\|\Delta\mathbf{X^{(m)}}\|_2 \qquad \text{(P1.5)} \end{aligned}$$

(2) Proof for Theorem 1.2: From Eq.(1.27), we have that

$$\begin{aligned} |\Delta\mathbf{a_i^{(m)}}| &= 2\left|\sum_{j\neq i}\frac{\mathbf{a_j^{(m)}}^\top\Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top\mathbf{a_i^{(m)}}}{\lambda_i^{(m)} - \lambda_j^{(m)}}\mathbf{a_j^{(m)}}\right| \\ &\leq 2\sum_{j\neq i}\left\|\frac{\mathbf{a_j^{(m)}}^\top\Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top\mathbf{a_i^{(m)}}}{\lambda_i^{(m)} - \lambda_j^{(m)}}\mathbf{a_j^{(m)}}\right\| \\ &\leq 2\sum_{j\neq i}\frac{\|\mathbf{a_j^{(m)}}\|}{|\lambda_i^{(m)} - \lambda_j^{(m)}|}\|\mathbf{a_j^{(m)}}^\top\Delta\mathbf{X^{(m)}}\mathbf{X^{(m)}}^\top\mathbf{a_i^{(m)}}\| \\ &\leq 2\|\Delta\mathbf{X^{(m)}}\|_2\sum_{j\neq i}\frac{(\lambda^{\max}_{\mathbf{X^{(m)}}^\top\mathbf{X^{(m)}}})^{\frac{1}{2}}}{|\lambda_i^{(m)} - \lambda_j^{(m)}|}. \qquad \text{(P1.6)} \end{aligned}$$

1.3 In Section 1.3, we used two types of item domains, web post and user label. However, social networks have multiple types of User-Generated Content (UGC), e.g., posts, labels, music, and movies. Therefore, the second-order graph is insufficient. How can the random walk strategy be extended to higher-order cases?

Solution :

We represent the following subgraphs contained in the high-order hybrid graph:

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

With respect to $\mathcal{G}^{(\mathcal{U})}$ 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 $(i = 1, \cdots, N)$

$$\begin{aligned} \mathbf{P}^{(\mathcal{U})} &= (\mathbf{D}^{(\mathcal{U})})^{-1}\mathbf{W}^{(\mathcal{U})}, && \text{(P1.7)} \\ \mathbf{P}^{(\mathcal{D}_i)} &= (\mathbf{D}^{(\mathcal{D}_i)})^{-1}\mathbf{W}^{(\mathcal{D}_i)}, && \text{(P1.8)} \end{aligned}$$

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

$$\begin{aligned} \mathbf{R}^{(\mathcal{U})}(t+1) &= \alpha\mathbf{P}^{(\mathcal{U})}\mathbf{R}^{(\mathcal{U})}(t) + (1-\alpha)\mathbf{I}, && \text{(P1.9)} \\ \mathbf{R}^{(\mathcal{D}_i)}(t+1) &= \beta_i\mathbf{P}^{(\mathcal{D}_i)}\mathbf{R}^{(\mathcal{D}_i)}(t) + (1-\beta_i)\mathbf{I}, && \text{(P1.10)} \end{aligned}$$

where $i = 1, 2, \cdots, N$, $0 \leq \alpha, \beta_1, \cdots, \beta_N \leq 1$. For the cross-domain subgraphs $\{\mathcal{G}^{(\mathcal{U}\mathcal{D}_i)}\}_{i=1}^N$, we compute 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

$$\begin{aligned} \mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+} &= \left(\mathbf{D}^{(\mathcal{U}\mathcal{D}_i)^+}\right)^{-1}\mathbf{W}^{(\mathcal{U}\mathcal{D}_i)^+}, && \text{(P1.11)} \\ \mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-} &= \left(\mathbf{D}^{(\mathcal{U}\mathcal{D}_i)^-}\right)^{-1}\mathbf{W}^{(\mathcal{U}\mathcal{D}_i)^-}, && \text{(P1.12)} \end{aligned}$$

where $i = 1, \cdots, N$. The cross-domain transition probability matrices can be updated using the following rules:

$$\begin{aligned} \mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(t+1) &= \delta_i\mathbf{R}^{(\mathcal{U})}(t)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(t) \\ &\quad + (1-\delta_i)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(t)\mathbf{R}^{(\mathcal{D}_i)}, && \text{(P1.13)} \\ \mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(t+1) &= \delta_i\mathbf{R}^{(\mathcal{U})}(t)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(t) \\ &\quad + (1-\delta_i)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(t)\mathbf{R}^{(\mathcal{D}_i)}, && \text{(P1.14)} \\ \mathbf{R}^{(\mathcal{U})}(t+1) &= \sum_{\mathcal{D}_i \in \mathcal{D}} \tau_i\mu_i\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(t)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^+}(t)^T \\ &\quad + \sum_{\mathcal{D}_i \in \mathcal{D}} \tau_i(1-\mu_i)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(t)\mathbf{P}^{(\mathcal{U}\mathcal{D}_i)^-}(t)^T \\ &\quad + \tau^{(\mathcal{U})}\mathbf{R}^{(\mathcal{U})}(t)\mathbf{R}^{(\mathcal{U})}(t)^T, && \text{(P1.15)} \end{aligned}$$

where $0 \leq \delta_i, \mu_i, \tau_i \leq 1$ are the trade-off parameters and $i = 1, 2, \cdots, N$. For a domain $\mathcal{D}_i$ without negative user-item links, we set $\mu_i = 1$ to update $\mathbf{R}^{(\mathcal{U})}$. The space complexity of this algorithm 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

Bibliography

[1] Marko Balabanovic and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[2] Peng Cui, Shifei Jin, Linyun Yu, Fei Wang, Wenwu Zhu, and Shiqiang Yang. Cascading outbreak prediction in networks: a data-driven approach. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 901–909. ACM, 2013.

[3] Peng Cui, Fei Wang, Shaowei Liu, Mingdong Ou, Shiqiang Yang, and Lifeng Sun. Who should share what?: item-level social influence prediction for users and posts ranking. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 185–194. ACM, 2011.

[4] Peng Cui, Fei Wang, Shiqiang Yang, and Lifeng Sun. Item-level social influence prediction with probabilistic hybrid factor matrix factorization. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[5] Peng Cui, Tianyang Zhang, Fei Wang, and Peng He. Perceiving group themes from collective social and behavioral information. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[6] Marco Gori, Augusto Pucci, V Roma, and I Siena. Itemrank: A random-walk based scoring algorithm for recommender engines. In *IJCAI*, volume 7, pages 2766–2771, 2007.

[7] Junming Huang, Xueqi Cheng, Jiafeng Guo, Huawei Shen, and Kun Yang. Social recommendation with interpersonal influence. In *ECAI*, volume 10, pages 601–606, 2010.

[8] Mohsen Jamali and Martin Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 397–406. ACM, 2009.

[9] Meng Jiang, Peng Cui, Xumin Chen, Fei Wang, Wenwu Zhu, and Shiqiang Yang. Social recommendation with cross-domain transferable knowledge. *Knowledge and Data Engineering, IEEE Transactions on (TKDE)*, 2015.

[10] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. Social contextual recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM)*, pages 45–54. ACM, 2012.

[11] Meng Jiang, Peng Cui, Fei Wang, Xinran Xu, Wenwu Zhu, and Shiqiang Yang. Fema: flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD)*, pages 1186–1195. ACM, 2014.

[12] Meng Jiang, Peng Cui, Fei Wang, Qiang Yang, Wenwu Zhu, and Shiqiang Yang. Social recommendation across multiple relational domains. In *Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM)*, pages 1422–1431. ACM, 2012.

[13] Meng Jiang, Peng Cui, Fei Wang, Wenwu Zhu, and Shiqiang Yang. Scalable recommendation with social contextual information. *Knowledge and Data Engineering, IEEE Transactions on (TKDE)*, 26(11):2789–2802, 2014.

[14] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[15] Tamara G Kolda and Jimeng Sun. Scalable tensor decompositions for multi-aspect data mining. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 363–372. IEEE, 2008.

[16] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[17] Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[19] Jure Leskovec, Ajit Singh, and Jon Kleinberg. Patterns of influence in a recommendation network. In *Advances in Knowledge Discovery and Data Mining*, pages 380–389. Springer, 2006.

[20] Lu Liu, Jie Tang, Jiawei Han, Meng Jiang, and Shiqiang Yang. Mining topic-level influence in heterogeneous networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 199–208. ACM, 2010.

[21] Hao Ma, Irwin King, and Michael R Lyu. Learning to recommend with explicit and implicit social relations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):29, 2011.

[22] Kanika Narang, Seema Nagar, Sameep Mehta, L Venkata Subramaniam, and Kuntal Dey. Discovery and analysis of evolving topical social discussions on unstructured microblogs. In *Advances in Information Retrieval*, pages 545–556. Springer, 2013.

[23] Kira Radinsky, Krysta Svore, Susan Dumais, Jaime Teevan, Alex Bocharov, and Eric Horvitz. Modeling and predicting behavioral dynamics on the web. In *Proceedings of the 21st international conference on World Wide Web*, pages 599–608. ACM, 2012.

[24] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[25] Luo Si and Rong Jin. Unified filtering by combining collaborative filtering and content-based filtering via mixture model and

exponential model. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 156–157. ACM, 2004.

[26] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383. ACM, 2006.

[27] Jimeng Sun, Dacheng Tao, Spiros Papadimitriou, Philip S Yu, and Christos Faloutsos. Incremental tensor analysis: Theory and applications. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(3):11, 2008.

[28] Jiliang Tang, Xia Hu, and Huan Liu. Social recommendation: a review. *Social Network Analysis and Mining*, 3(4):1113–1133, 2013.

[29] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. 2006.

[30] Fei Wang, Hanghang Tong, and Ching-Yung Lin. Towards evolutionary nonnegative matrix factorization. In *AAAI*, volume 11, pages 501–506, 2011.