Rating Knowledge Sharing in Cross-Domain Collaborative Filtering

Bin Li, Xingquan Zhu, *Senior Member, IEEE*, Ruijiang Li, and Chengqi Zhang, *Senior Member, IEEE*

***Abstract*—Cross-domain collaborative filtering (CF) aims to share common rating knowledge across multiple related CF domains to boost the CF performance. In this paper, we view CF domains as a 2-D site-time coordinate system, on which multiple related domains, such as similar recommender sites or successive time-slices, can share group-level rating patterns. We propose a unified framework for cross-domain CF over the site-time coordinate system by sharing group-level rating patterns and imposing user/item dependence across domains. A generative model, say ratings over site-time (ROST), which can generate and predict ratings for multiple related CF domains, is developed as the basic model for the framework. We further introduce cross-domain user/item dependence into ROST and extend it to two real-world cross-domain CF scenarios: 1) ROST (sites) for alleviating rating sparsity in the target domain, where multiple similar sites are viewed as related CF domains and some items in the target domain depend on their correspondences in the related ones; and 2) ROST (time) for modeling user-interest drift over time, where a series of time-slices are viewed as related CF domains and a user at current time-slice depends on herself in the previous time-slice. All these ROST models are instances of the proposed unified framework. The experimental results show that ROST (sites) can effectively alleviate the sparsity problem to improve rating prediction performance and ROST (time) can clearly track and visualize user-interest drift over time.**

***Index Terms*—Collaborative filtering (CF), cross-domain, knowledge transfer, rating sparsity, user-interest drift.**

I. Introduction

COLLABORATIVE filtering (CF) has become the most popular technique in real-world recommender systems since it can efficiently handle a large-scale database in a "content-free" manner. The basic idea of CF is memory-based that finds k-nearest neighboring users [1] or items [2] based on historical rating data to predict ratings for an active user. Model-based methods were proposed later and most of them also follow the same basic idea, i.e., finding similar users or/and items, but resort to more complicated clustering techniques in various ways, such as latent variable models [3], [4] and low-rank approximations [5], [6]. Thus, the essential problem of CF methods is how to find similar users/items and how to measure similarities between them.

Thus far, most existing CF methods are single-domain based, which make predictions based on one rating matrix. In other words, these methods can only find similar users/items in a single domain. However, in many recommendation scenarios, multiple related CF domains may be presented at the same time and finding similar users/items across domains becomes possible, such that common rating knowledge can be shared among related domains. We take two real-world cross-domain CF problems for example, which will be addressed later in this paper.

The first scenario is "cross-domain CF over sites" (i.e., CF across recommender systems), in which one site is viewed as one CF domain. This scenario can emerge when users wish to borrow rating knowledge from some related auxiliary domains (e.g., popular movie websites), whose rating matrices are relatively dense, to alleviate the rating sparsity problem in the sparse target domain (e.g., a new movie website), in which k-NN or clustering can hardly obtain good results due to sparsity. Although user/item sets of auxiliary and target domains are different, they may have some implicit correspondences. For example, movies of two domains have similar categories while users of two domains have similar interest distributions over movie categories. If we can find some rating-pattern matchings among related domains, the clustering knowledge can be naturally transferred from one domain to another.

The second scenario is "cross-domain CF over time" (i.e., CF across temporal domains), in which the ratings collected in each time-slice is viewed as one CF domain. Because users' interests keep drifting over time and a user is likely to be interested in different item categories at different time, we cannot simply view the multiple counterparts[1] of the same user in different temporal domains are identical. Some recent work has considered such interest-drift problem and proposed an approach by adding time-dependent components to a single-domain CF model [7]. An alternative way is to build on a cross-domain CF framework by viewing the counterparts of the same user in successive temporal domains are different but related users. If we can find the unchanged rating patterns (static components) shared across temporal domains,

Manuscript received April 1, 2013; revised April 29, 2014 and July 22, 2014; accepted July 25, 2014. Date of publication August 12, 2014; date of current version April 13, 2015. The work of B. Li was supported by UTS Early Career Research Grant. This paper was recommended by Associate Editor A. Nuernberger.

B. Li and C. Zhang are with the Center for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Ultimo, NSW 2007, Australia (e-mail: bin.li-1@uts.edu.au; chengqi.zhang@uts.edu.au).

X. Zhu is with the Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: xzhu3@fau.edu).

R. Li is with the School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: rjli@fudan.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2343982

[1] A "user-counterpart" refers to a user in a certain temporal domain. We view a user has multiple counterparts, which can have different interests in different temporal domains.

the drifting factors of users (changing components) in each temporal domain can be easily captured.

In this paper, we view CF domains as a 2-D site-time coordinate system, through which various cross-domain CF problems (such as the two introduced above) can be solved in the same framework. We first show that multiple related CF domains can share group-level rating patterns through matching user/item groups of different CF domains. Then we propose a unified framework for cross-domain CF over the site-time coordinate system by sharing a group-level rating matrix and imposing user/item dependence across domains (Section III). A Bayesian generative model, say ratings over site-time (ROST), which can generate and predict ratings for multiple related CF domains on the site-time coordinate system, is also developed as the basic model for the proposed cross-domain CF framework (Section IV). We further introduce cross-domain user/item dependence into ROST and extend it to the above two real-world cross-domain CF scenarios: 1) ROST (sites) for alleviating rating sparsity in the target domain, where multiple similar sites are viewed as related CF domains and some items in the target domain depend on their correspondences in the related domains (Section V) and 2) ROST (time) for modeling user-interest drifting over time, where a series of time-slices are viewed as related CF domains and a user at current time-slice depends on herself in the previous time-slice (Section VI). All these ROST models are instances of the proposed unified framework and they are defined with the same generative process, where the cross-domain user/item dependence is enabled by using Bayesian priors. We use Gibbs sampling for latent variable inference in these models. The main contribution of this paper is threefold.

1) Formalize cross-domain CF on a 2-D site-time coordinate system, on which various cross-domain CF problems can be formulated in the same way.
2) Propose a unified framework for cross-domain CF problems, which shares group-level rating patterns by matching user/item groups from different domains and imposing user/item dependence across domains.
3) Instantiate the proposed cross-domain CF framework by developing a generative model and its two extensions, which effectively address two real-world CF issues, rating sparsity and user-interest drifting.

This paper essentially advances our previous work on cross-domain CF [8]–[10] by: 1) providing a unified learning framework for rating knowledge sharing across site domains (asymmetric setting [8] and symmetric setting [10]) and temporal domains [9][2]; 2) developing Bayesian generative models to instantiate the framework and solve real-world cross-domain CF problems with efficient inference algorithms; and 3) conducting comprehensive cross-domain CF experiments and case study on real-world recommendation data sets.

Our comprehensive experiments on three real-world recommendation data sets show that ROST (sites) can effectively alleviate the sparsity problem to improve rating prediction performance in the sparse target domain, and ROST (time) can explicitly track and visualize user-interest drift over time (Section VII). We also conduct a case study of sharing ratings over both sites and time simultaneously to clearly demonstrate the advantage of our framework. The notations used through the paper are listed in Table I.

[2]Reference [9] is a special case of ROST applied to temporal domains.

TABLE I
NOTATIONS

Notation	Description
T	number of domains (index t)
$N^{(t)}$	number of users in domain t (index n)
$M^{(t)}$	number of items in domain t (index m)
$\mathbf{X}_{nm}^{(t)}$	rating provided by user n on item m in domain t
$\mathcal{S}^{(t)}$	user-item set of the observed ratings in $\mathbf{X}^{(t)}$
K	number of the shared user groups (index k)
L	number of the shared item groups (index l)
R	rating scales (scores) $\{1, \ldots, R\}$ (index r)
$\theta_n^{\mathcal{U}(t)}$	mixing proportion of K user groups for user n in domain t (user-group membership)
$\theta_m^{\mathcal{I}(t)}$	mixing proportion of L item groups for item m in domain t (item-group membership)
$\phi_{k,l}$	mixing proportion of R rating scales for user-item joint group (k, l)
$\alpha^{\mathcal{U}}$	hyper-parameter for $\theta_n^{\mathcal{U}(t)}$
$\alpha^{\mathcal{I}}$	hyper-parameter for $\theta_m^{\mathcal{I}(t)}$
β	hyper-parameter for $\phi_{k,l}$
$z_{nm}^{\mathcal{U}(t)}$	latent user-group variable for a rating $\mathbf{X}_{nm}^{(t)}$
$z_{nm}^{\mathcal{I}(t)}$	latent item-group variable for a rating $\mathbf{X}_{nm}^{(t)}$

II. Related Work

In the following, we first introduce latent variable model based CF. Then we introduce some state-of-the-art temporal CF methods because ROST (time) introduced in Section VI is a temporal CF method. After that, we briefly review existing works on cross-domain CF.

A. Latent Variable Models

Our ROST models are latent variable models (LVMs). Probabilistic latent semantic analysis (pLSA) [3] is an early work that applies LVM to CF. Later, LVMs with a pair of latent variables, associated to users and items respectively, were also proposed for CF, such as two-sided clustering [11] and flexible mixture models [4]. Recently, many Bayesian extensions of pLSA-style models [12], [13] were extensively exploited for CF, including Bi-LDA [14], which underlies the fundamental algorithm of ROST. Since the ROST models take into account relatedness and dynamics between related CF domains, our work is also related to dynamic probabilistic models [15]–[17]. To achieve more flexibilities in different cross-domain CF problem settings, we adopt the approximate inference strategy used in [16] and [17] to establish dependence between related domains using Bayesian priors.

B. Temporal Collaborative Filtering

The state-of-the-art in this area is TimeSVD++ [7], in which a latent feature of user/item (factors of SVD decomposition of the user-item rating matrix) is a combination of one static component and one time-dependent component. In [18], apart from "user" and "item," "time" is viewed as the third dimension and tensor factorization is used to factorize temporal components. Besides temporal dynamics, a spatio-temporal CF method further considers user similarities [19]. Unfortunately, the extracted temporal components in these methods cannot be interpreted to describe user-interest drifting. On the contrary, ROST (time) is able to explicitly visualize users' interest variations using user-group memberships.

C. Cross-Domain Collaborative Filtering

In the following, we will give a brief introduction to the existing works on cross-domain CF in terms of knowledge transfer styles (please also refer to our work [20] for a brief survey on cross-domain CF).

1) Rating-Pattern Sharing: Rating-pattern sharing was first proposed in [8], where it is also called CodeBook transfer (CBT), for solving domain adaptation problems in CF. The idea was latter incorporated into a probabilistic model, rating-matrix generative model (RMGM) [10], for solving multitask learning problems in CF. More recently, this knowledge transfer style was adapted to temporal domains in [9] and multisource domains in [21]. In these works, multiple rating matrices from related CF domains can share group-level rating patterns (or partially share [22]). Such knowledge transfer style can be viewed as a type of feature-representation transfer [23] (e.g., self-taught learning [24]); differently, rating patterns are a two-sided feature representation for both rows and columns. In this paper, we propose a unified framework to accommodate a variety of cross-domain CF problems, based on this rating-pattern transfer style.

2) Latent-Feature Sharing: This knowledge transfer style was applied to CF to incorporate side-information by simultaneously factorizing the rating matrix and some other related matrices, for example, user-movie, movie-genre, and actor-movie matrices for movie recommendation [25] and document-citation, document-author, and document-venue matrices for document recommendation [26]. Recently, a two-sided latent-feature sharing method named coordinate system transfer (CST) is introduced in [27], which incorporates both user and item side-information into rating matrix factorization. Later, implicit user feedback was taken into account for knowledge transfer in [28]. Tri-factorization is also applied to capture user-item-domain interactions in [29]. The existing works in this knowledge transfer style require to share the same user and/or item sets across CF domains, which are unavailable in our cross-domain CF problem settings.

3) Domain Correlating: The idea of exploiting correlations among item (or user) domains was first mentioned in [30] without giving a solution. Some methods based on this idea were proposed later, including collective link prediction (CLP) [31] and multidomain collaborative filtering (MCF) [32]. Both CLP and MCF explore user/item domain correlations via latent feature learning. More recently, a method resorts to estimating domain correlations based on explicit cues (tag similarities) [33]. However, these works are based on a strong assumption that the users (or items) of different CF domains are the same set. The problem setting of this knowledge transfer style is different from ours where no same user/item set is required for different CF domains.

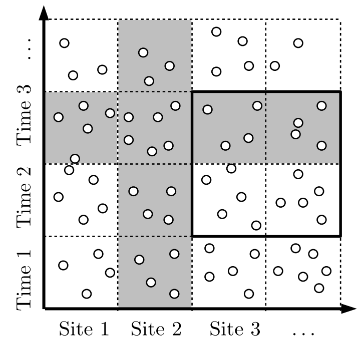


Fig. 1. Site-time coordinate system for cross-domain CF. The horizontal shaded band illustrates an example of cross-domain CF over sites (four site domains at time 3). The vertical shaded band illustrates an example of cross-domain CF over time (four temporal domains on site 2). The small circles represent ratings from certain sites at certain time.

4) Neighborhood Augmentation: The last knowledge transfer style is augmenting user/item neighborhoods from related CF domains. In [34], the user affinity graphs from two CF domains are embedded and aligned in the same low-dimensional space such that a user profile can be reconstructed from the user profiles in the other domain. However, this paper aims to predict a user's profile in domain *B* given the same user's profile in domain *A*, which is different from our problem setting. Shared collaborative filtering [35] augments the neighborhood set of a user in domain *A* from the item-affinity information in domain *B*; but it is required that the two CF domains have the same item set. Our recent work [36] also makes use of the item-affinity information of a related domain to help noisy user detection in the target domain. Finally, [37] proposes a transitive closure method to augment paths between the item-affinity graphs of two CF domains and the ratings in the combined rating matrices can be predicted using a single-domain CF method.

III. UNIFIED FRAMEWORK FOR CROSS-DOMAIN CF

To clearly describe our cross-domain CF problem settings, we first give some important definitions. Then, we introduce our rating knowledge sharing scheme. Finally, we propose a unified framework for cross-domain CF.

A. Definitions

Definition 1 (Site-Time Coordinate System): We view the space of $Site \times Time$ as a 2-D coordinate system and the ratings collected from different sites at different time are pooled on it. Each rating is located on the site-time coordinate system based on its site-identity and time-stamp (see Fig. 1).

It is worth noting that the elements on the site-coordinate are the union of user-item pairs from multiple sites. Each site has a user set and an item set, and the user/item sets of different sites can have overlaps.

Definition 2 (Collaborative Filtering Domains): Given a site-time coordinate system, CF domains can be 1) a set of sites on the site-coordinate or 2) a series of time-slices on the time-coordinate. In our problem, "site" and "time" are equivalent concepts; either site-coordinate or time-coordinate can be partitioned into multiple CF domains.

Definition 3 (Cross-Domain Collaborative Filtering): Cross-domain CF is performed on multiple related CF domains by sharing ratings in those domains, where "related CF domains" refer to a set of 1) rating matrices from multiple similar sites within the same time-span[3] (say cross-domain CF over sites) or 2) snapshot rating matrices of a series of time-slices from one site (say cross-domain CF over time).

As illustrated in Fig. 1, cross-domain CF over sites is performed on the ratings in a horizontal band on the site-time coordinate system while cross-domain CF over time is performed on the ratings in a vertical band. The ratings in each cell on the site-time coordinate system can be used to construct a user-item rating matrix for a certain site at certain time. The successive cells in a horizontal/vertical band form a set of related rating matrices for cross-domain CF. The user/item sets in these related CF domains have different distributions on user/item-groups since their interests/atributes may change over sites and time. In this paper, we aim to propose a unified framework for accommodating cross-domain CF problems over both sites and time.

B. Rating Knowledge Sharing

We start our study on cross-domain CF over sites and time by introducing how to transfer rating knowledge across related rating matrices. Since user-item ratings are dyadic data with two finite sets of objects (user set and item set) [38], we can co-cluster users and items simultaneously and find implicit correspondences among different rating matrices by matching group-level rating patterns. We take movie rating matrices from two different sites (domains) for example. On one hand, movies from two domains should have similar categories in terms of genres, actors, and other attributes. On the other hand, users from two domains are the subsets of the real-world population and should reflect similar interest distributions over movie categories. We can thus simultaneously group users based on their ratings on items and group items based on their associated ratings provided by users in both domains to find shared group-level rating patterns.

In Fig. 2, we illustrate this rating-matrix matching process, which is obtained using the basic ROST model introduced in Section IV on two randomly generated synthetic rating matrices (left). By simultaneously grouping users and items in the two rating matrices, we can uncover the latent block structure shared between them (middle). If we estimate the expected ratings for each block (user-item group dyad), we can obtain a 3×3 group-level rating matrix shared between two domains (right). This is an ideal case that two rating matrices can be perfectly matched. In real-world scenarios, there may be substantial amount of noise and each rating matrix may only have a subset of the rating patterns (a submatrix) in the shared group-level rating matrix. Nevertheless, this illustration intuitively shows that, if user-interests and item-categories in multiple rating matrices indeed have similar distributions, we can uncover the underlying shared rating patterns by matching user/item groups among different domains.

[3]This is because, if rating matrices from different sites belong to different ages (e.g., 2000s and 2010s), they are unlikely to be related since products and user-tastes keep evolving over time.

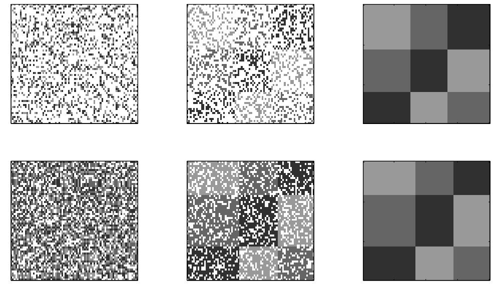


Fig. 2. Illustration of group-level rating pattern matching between two synthetic rating matrices (left). Three gray scales denote different rating scores and white entries denote missing values. By simultaneously grouping users and items in the two rating matrices (middle), a shared 3×3 block structure, say group-level rating matrix, can be discovered (right).

C. Cross-Domain CF Framework

The above observation suggests that we can match T rating matrices on the site-time coordinate system as long as they are related in user/item groups (e.g., rating matrices from similar sites or successive time-slices). We use t to index related CF domains and decompose an $N^{(t)} \times M^{(t)}$ rating matrix $\mathbf{X}^{(t)}$ from one of related domains into three parts: a $K \times L$ group-level rating matrix $\mathbf{B}$, an $N^{(t)} \times K$ user-group membership matrix $\mathbf{P}^{(t)}$ and an $M^{(t)} \times L$ item-group membership matrix $\mathbf{Q}^{(t)}$. Each row in $\mathbf{P}^{(t)}$ and $\mathbf{Q}^{(t)}$ sums to 1 ($\mathbf{P}^{(t)}\mathbf{1} = \mathbf{1}$ and $\mathbf{Q}^{(t)}\mathbf{1} = \mathbf{1}$), which means a mixing proportion of user/item groups. We assume that $\mathbf{B}$ can be shared across domains so it is not assigned a domain index. An entry in $\mathbf{B}$, say $\mathbf{B}_{kl}$, denotes the expected rating provided by user group k on item group l. The rating matrix $\mathbf{X}^{(t)}$ can be reconstructed by

$$\hat{\mathbf{X}}^{(t)} = \mathbf{P}^{(t)}\mathbf{B}\left[\mathbf{Q}^{(t)}\right]^{\top} \tag{1}$$

each entry in $\hat{\mathbf{X}}^{(t)}$ is indeed a weighted sum of the expected group-level ratings in $\mathbf{B}$ and the weights in terms of user/item-group memberships are different from one another. The tri-factorization representation of a rating matrix in (1) suggests a cross-domain CF framework—sharing a group-level rating matrix ($\mathbf{B}$), which encodes the common rating knowledge of multiple related domains, while assuming that users and items ($\mathbf{P}^{(t)}$ and $\mathbf{Q}^{(t)}$) from different domains are independent. It is worth noting that, within this framework, the rating matrices of different sizes or have different user/item sets can also be matched as long as they have similar user and item distributions.

In many real-world scenarios, we can easily find relatedness between users/items from different CF domains. For example, we can identify a set of same movies existing in multiple sites for cross-domain CF over sites. We can also view the user-counterparts in successive temporal domains

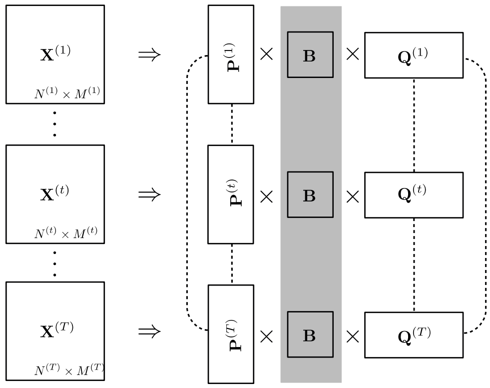


Fig. 3. Unified framework for cross-domain CF over site-time coordinates. $\{\mathbf{X}^{(t)}\}_{t=1}^{T}$ can be rating matrices from T sites or snapshot rating matrices of T time-slices. A group-level rating matrix $\mathbf{B}$ is shared across domains (in the shaded area) and additional relatedness between users/items from different domains can also be considered (represented in dashed lines).

are different but related users for cross-domain CF over time. Thus, besides sharing group-level rating patterns, we can also take into account additional relatedness between users/items from different CF domains. We will show later that we can enable such relatedness by imposing dependence on related users/items using Bayesian priors. Now we can construct a unified framework for cross-domain CF over the site-time coordinate system, by: 1) sharing a group-level rating matrix ($\mathbf{B}$) across multiple related CF domains and 2) enabling additional relatedness between users/items ($\mathbf{P}^{(t)}$/$\mathbf{Q}^{(t)}$) from different CF domains. The proposed framework is shown in Fig. 3.

IV. Basic ROST Model

In this section, we introduce the basic ROST model for the proposed cross-domain CF framework. With ROST, we can infer latent group-level rating patterns based on the observed ratings collected from multiple related CF domains. But for the moment, ROST has not yet taken into account cross-domain user/item dependence (i.e., no relatedness between $\{\mathbf{P}^{(t)}\}$ or $\{\mathbf{Q}^{(t)}\}$). We will introduce user/item dependence into ROST in the next two sections.

ROST is a Bayesian generative model which has the following advantages: 1) its generative process is intuitive and interpretable; 2) cross-domain user/item dependence can be enabled using prior knowledge (prior distributions); and 3) a variety of off-the-shelf approximate inference algorithms can be used for inferring latent variables in the proposed models (we will use Gibbs sampling).

We will extend a variant of latent Dirichlet allocation (LDA) [12], say Bi-LDA [14], to solve the tri-factorization problem (1). Some well-known co-clustering methods (see [39]) are also applicable to CF [40]. However, [39] is an orthogonal factorization, $\mathbf{X} = \mathbf{USV}^{\top}$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal bases, which cannot be interpreted as mixing proportions. In contrast, Dirichlet distributions adopted in the proposed models are straightforward for modeling mixing proportions (user/item-group memberships).

A. Generative Process

Given T related CF domains, each domain is represented with an $N^{(t)} \times M^{(t)}$ rating matrix $\mathbf{X}^{(t)}$, where $\mathbf{X}_{nm}^{(t)}$ is the rating provided by user n on item m in domain t, $(n, m) \in \mathcal{S}^{(t)}$ and $\mathcal{S}^{(t)}$ denotes the user-item set of the observed ratings in $\mathbf{X}^{(t)}$. Each rating is associated with a pair of latent variables $(z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)})$, which are the user-group and item-group indices of the rating $\mathbf{X}_{nm}^{(t)}$, respectively. The generative process of the basic ROST model is as follows.

1) For user-item joint group (k, l), choose $\phi_{k,l} \sim \mathit{Dirichlet}(\beta)$.
2) For user n in domain t, choose $\theta_n^{\mathcal{U}(t)} \sim \mathit{Dirichlet}(\alpha^{\mathcal{U}})$.
3) For item m in domain t, choose $\theta_m^{\mathcal{I}(t)} \sim \mathit{Dirichlet}(\alpha^{\mathcal{I}})$.
4) For the rating $\mathbf{X}_{nm}^{(t)}$:
   a) choose a user group $z_{nm}^{\mathcal{U}(t)} \sim \mathit{Multinomial}(\theta_n^{\mathcal{U}(t)})$;
   b) choose an item group $z_{nm}^{\mathcal{I}(t)} \sim \mathit{Multinomial}(\theta_m^{\mathcal{I}(t)})$;
   c) choose a rating $\mathbf{X}_{nm}^{(t)} \sim \mathit{Multinomial}(\phi_{z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)}})$.

where $\phi_{k,l}$ is the mixing proportion of R rating scales for the ratings provided by user group k on item group l, and $\sum_{r=1}^{R} \phi_{k,l,r} = 1$; $\theta_n^{\mathcal{U}(t)}$ is the mixing proportion of K user groups (user-group membership) for user n, and $\sum_{k=1}^{K} \theta_{n,k}^{\mathcal{U}(t)} = 1$; $\theta_m^{\mathcal{I}(t)}$ is the mixing proportion of L item groups (item-group membership) for item m, and $\sum_{l=1}^{L} \theta_{m,l}^{\mathcal{I}(t)} = 1$; $\alpha^{\mathcal{U}}$, $\alpha^{\mathcal{I}}$, and β are hyper-parameters of the Dirichlet priors. The graphical model representation of the basic ROST model is illustrated in Fig. 4.

Now, the shared group-level rating matrix $\mathbf{B}$ across T CF domains and the user/item-group membership matrices $\{\mathbf{P}^{(t)}\}_{t=1}^{T}$ and $\{\mathbf{Q}^{(t)}\}_{t=1}^{T}$ can be represented using the above latent variables: $\mathbf{B}_{kl} = \sum_{r=1}^{R} r\phi_{k,l,r}$, $\mathbf{P}_{nk}^{(t)} = \theta_{n,k}^{\mathcal{U}(t)}$ and $\mathbf{Q}_{ml}^{(t)} = \theta_{m,l}^{\mathcal{I}(t)}$. This suggests that we can solve the cross-domain CF problem formulated in (1) if we infer the latent variables in ROST.

B. Inference and Prediction

The above generative process can be viewed as an extension of Bi-LDA [14] in multidomain scenarios. We can thus adopt the similar Gibbs sampling algorithm used in [14]. Let $\mathbf{X} = \{\mathbf{X}^{(t)}\}_{t=1}^{T}$, $z^{\mathcal{U}} = \{z^{\mathcal{U}(t)}\}_{t=1}^{T}$, $z^{\mathcal{I}} = \{z^{\mathcal{I}(t)}\}_{t=1}^{T}$, $\theta^{\mathcal{U}} = \{\theta^{\mathcal{U}(t)}\}_{t=1}^{T}$ and $\theta^{\mathcal{I}} = \{\theta^{\mathcal{I}(t)}\}_{t=1}^{T}$. The joint distribution of all the random variables in ROST gives

$$P\left(\mathbf{X}, z^{\mathcal{U}}, z^{\mathcal{I}}, \phi, \theta^{\mathcal{U}}, \theta^{\mathcal{I}} | \beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}}\right) = P\left(\mathbf{X} | z^{\mathcal{U}}, z^{\mathcal{I}}, \phi\right) P(\phi|\beta) P\left(z^{\mathcal{U}} | \theta^{\mathcal{U}}\right) P\left(\theta^{\mathcal{U}} | \alpha^{\mathcal{U}}\right) P\left(z^{\mathcal{I}} | \theta^{\mathcal{I}}\right) P\left(\theta^{\mathcal{I}} | \alpha^{\mathcal{I}}\right). \quad (2)$$

By analytically marginalizing out all the parameters of the Dirichlet-Multinomial conjugate distributions in (2), i.e., $\{\phi, \theta^{\mathcal{U}}, \theta^{\mathcal{I}}\}$, we can obtain an expression for the joint probability $P(\mathbf{X}, z^{\mathcal{U}}, z^{\mathcal{I}} | \beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}})$. Then we use the collapsed Gibbs sampling algorithm for inferring the latent variables $z^{\mathcal{U}}$ and $z^{\mathcal{I}}$. For each latent variable pair $(z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)})$, the

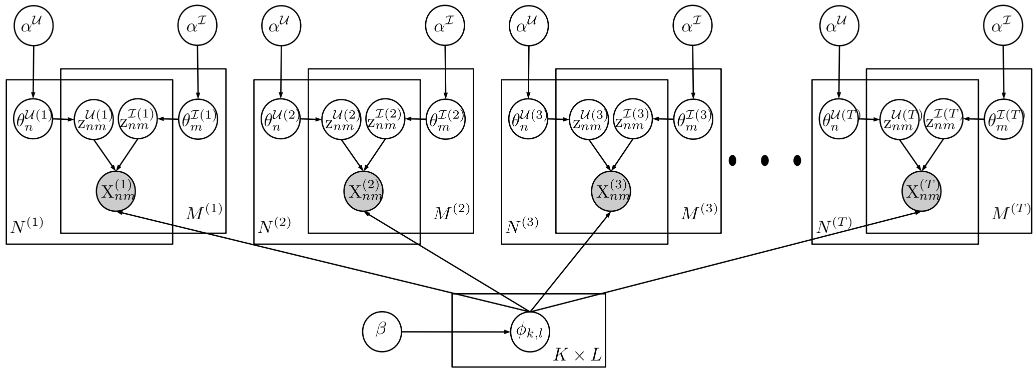


Fig. 4. Graphical model representation of the basic ROST model, in which the group-level rating patterns (ϕ) are shared by T CF domains.

S1: Input $\{\mathbf{X}^{(t)}\}_{t=1}^{T}$, K, L, $\alpha^{\mathcal{U}}$, $\alpha^{\mathcal{I}}$, β, and $iter$.
S2: Sample $(z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)})$ based on (3) for $iter$ epoches. One epoch means scanning $\{\mathcal{S}^{(t)}\}_{t=1}^{T}$ once.
S3: Estimate $\mathbf{B}$, $\{\mathbf{P}^{(t)}\}_{t=1}^{T}$ and $\{\mathbf{Q}^{(t)}\}_{t=1}^{T}$ using (4), (5) and (6), respectively.
S4: Predict a rating $\mathbf{X}_{nm}^{(t)}$ using (1).

Fig. 5. Basic ROST model: training and prediction.

conditional distribution is

$$P\left(z_{nm}^{\mathcal{U}(t)} = k, z_{nm}^{\mathcal{I}(t)} = l | z_{\neg(nmt)}^{\mathcal{U}}, z_{\neg(nmt)}^{\mathcal{I}}, \mathbf{X}; \beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}}\right) \propto \left(\frac{h_{kl\mathbf{X}_{nm}^{(t)}}^{\neg(nmt)} + \beta}{\sum_r h_{klr}^{\neg(nmt)} + R\beta}\right)\left(h_{nkt}^{\neg(nmt)} + \alpha^{\mathcal{U}}\right)\left(h_{mlt}^{\neg(nmt)} + \alpha^{\mathcal{I}}\right) \quad (3)$$

where $z_{\neg(nmt)}^{\mathcal{U}}$ denotes the set $z^{\mathcal{U}}$ excluding $z_{nm}^{\mathcal{U}(t)}$; $h_{klr}^{\neg(nmt)}$ denotes the number of ratings that fall in cell (k, l, r), excluding $\mathbf{X}_{nm}^{(t)}$. Similar definitions are applicable for $h_{nkt}^{\neg(nmt)}$ and $h_{mlt}^{\neg(nmt)}$. The full expression of the joint distribution (2) and the derivation of the conditional distribution (3) for Gibbs sampling are detailed in Appendix.

After inferring $\{z^{\mathcal{U}}, z^{\mathcal{I}}\}$, we can estimate the shared group-level rating matrix $\mathbf{B}$ and the user/item-group membership matrices, $\{\mathbf{P}^{(t)}, \mathbf{Q}^{(t)}\}_{t=1}^{T}$, for each domain as follows:

$$\mathbf{B}_{kl} = \sum_{r=1}^{R} r\phi_{k,l,r} = \frac{\sum_r r(h_{klr} + \beta)}{\sum_r h_{klr} + R\beta} \quad (4)$$

$$\mathbf{P}_{nk}^{(t)} = \theta_{n,k}^{\mathcal{U}(t)} = \frac{h_{nkt} + \alpha^{\mathcal{U}}}{\sum_k h_{nkt} + K\alpha^{\mathcal{U}}} \quad (5)$$

$$\mathbf{Q}_{ml}^{(t)} = \theta_{m,l}^{\mathcal{I}(t)} = \frac{h_{mlt} + \alpha^{\mathcal{I}}}{\sum_l h_{mlt} + L\alpha^{\mathcal{I}}}. \quad (6)$$

When $\mathbf{B}$ and $\{\mathbf{P}^{(t)}, \mathbf{Q}^{(t)}\}_{t=1}^{T}$ are obtained, we can predict any rating in the given T domains by using (1). We sum up the modeling steps for the basic ROST model in Fig. 5.

V. Cross-Domain CF Over Sites

In real-world scenarios, in addition to implicit group-level rating pattern matchings, two CF domains may also have some explicit relatedness, which can be easily established by identifying a subset of similar or even same users/items from related CF domains (e.g., same movies from two movie recommendation websites). Based on this observation, a common cross-domain CF problem can be raised: Given a target CF domain whose rating matrix is sparse, can we borrow useful rating knowledge from related auxiliary domains and further take into account some explicit relatedness to make a more reliable user/item-group matching estimation? In this section, we extend the basic ROST model by taking into account some explicit relatedness between users/items from different CF domains and introduce ROST (sites). The goal of ROST (sites) is to alleviate the rating sparsity problem and improve recommendation performance in the target domain.

A. Problem Setting

Given T related CF domains, we represent each domain as an $N^{(t)} \times M^{(t)}$ rating matrix $\mathbf{X}^{(t)}$. Note that the user/item sets in different domains are different from one another and can be of different sizes. A target domain, whose rating matrix is sparse, is one of the given domains, and the others are auxiliary domains whose rating matrices are relatively dense and are related to the target domain. Our task is to make use of the rating knowledge from auxiliary domains to alleviate the rating sparsity in the target domain for better rating predictions. Without loss of generality, we let the last domain be the target domain, i.e., $\mathbf{X}^{(T)}$, and the other domains $\{\mathbf{X}^{(t)}\}_{t=1}^{T-1}$ be the auxiliary domains. In the case of $T = 2$, there is only one auxiliary domain (more common in practice).

We use a mapping function ℓ to link an item m in the target domain to its correspondence $\ell(m)$ in the auxiliary domains, where $m \in \mathcal{C}^{\mathcal{I}}$ and $\mathcal{C}^{\mathcal{I}}$ denotes the item set in the target domain which have correspondences in the auxiliary domains. Like the basic ROST model, a group-level rating matrix $\mathbf{B}$ is shared by $\{\mathbf{X}^{(t)}\}_{t=1}^{T}$ from both auxiliary and target domains. In addition, we impose dependence for an item $m \in \mathcal{C}^{\mathcal{I}}$ by letting its correspondence's item-group membership be the prior of its item-group membership: $\theta_m^{\mathcal{I}(T)} \sim Dirichlet(\lambda\theta_{\ell(m)}^{\mathcal{I}(t_m)})$, where t_m is the domain index of $\ell(m)$ and λ is a weighting parameter for tuning the concentration of the Dirichlet prior. The intuition of $\lambda\theta_{\ell(m)}^{\mathcal{I}(t_m)}$ can be interpreted as the pseudo count of ratings for item m in L item groups before any rating from the target domain is observed [13]. Similar dependence is also applicable for users between auxiliary and target domains. The

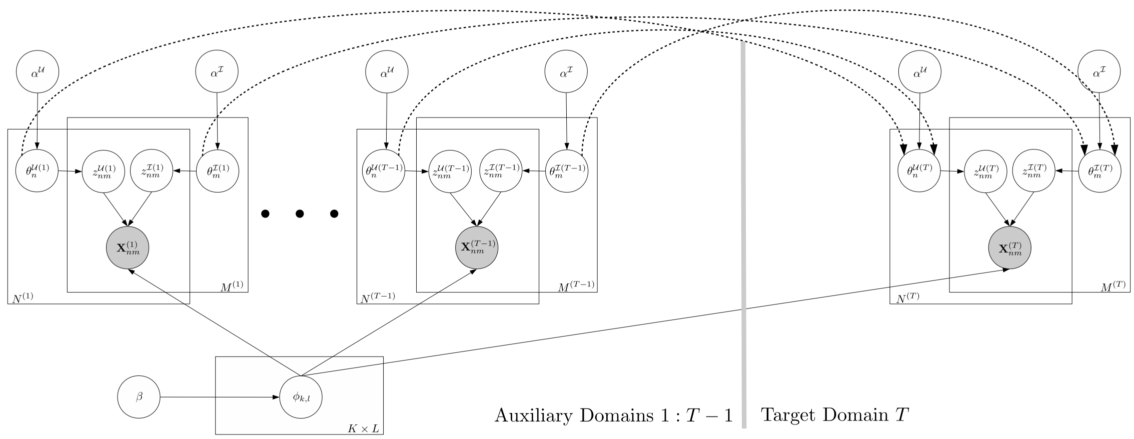


Fig. 6. Graphical model representation of ROST (sites). The group-level rating patterns (ϕ) are first learned from the $T-1$ auxiliary domains and then are reused by the target domain, while the user/item-group memberships ($\theta^{\mathcal{U}(T)}/\theta^{\mathcal{I}(T)}$) of the target domain partially depend on their corresponding users/items in the $T-1$ auxiliary domains (dashed arrowed lines denote partially dependence).

user set having correspondences is denoted by $\mathcal{C}^{\mathcal{U}}$ and the corresponding users can also be mapped by using ℓ.

B. ROST (Sites)

For ROST (sites), we adopt an asymmetric knowledge transfer strategy used in [8], i.e., to learn group-level rating patterns from the auxiliary domains first and transfer them to the target domain later. In ROST (sites), besides transferring group-level rating patterns, user/item-group memberships of related users/items between auxiliary and target domains should also be transferred. To this end, we first use the basic ROST model introduced in Section IV to learn ϕ and $\{\theta^{\mathcal{U}(t)}, \theta^{\mathcal{I}(t)}\}_{t=1}^{T-1}$ on the auxiliary domains $\{\mathbf{X}^{(t)}\}_{t=1}^{T-1}$, where the group-level rating patterns are encoded in ϕ and some elements in $\{\theta^{\mathcal{U}(t)}, \theta^{\mathcal{I}(t)}\}_{t=1}^{T-1}$ will be used as the Dirichlet priors for their correspondences in the target domain T.

Given ϕ and $\{\theta^{\mathcal{U}(t)}, \theta^{\mathcal{I}(t)}\}_{t=1}^{T-1}$ learned from the auxiliary domains, the generative process of ROST (sites) is the same as the basic ROST model except for the user/item-group membership priors for users in $\mathcal{C}^{\mathcal{U}}$ and items in $\mathcal{C}^{\mathcal{I}}$.

1) For user n in the target domain T:
   a) choose $\theta_n^{\mathcal{U}(T)} \sim Dirichlet(\alpha^{\mathcal{U}})$, if $n \notin \mathcal{C}^{\mathcal{U}}$;
   b) choose $\theta_n^{\mathcal{U}(T)} \sim Dirichlet(\lambda\theta_{\ell(n)}^{\mathcal{U}(t_n)})$, if $n \in \mathcal{C}^{\mathcal{U}}$.
2) For item m in the target domain T:
   a) choose $\theta_m^{\mathcal{I}(T)} \sim Dirichlet(\alpha^{\mathcal{I}})$, if $m \notin \mathcal{C}^{\mathcal{I}}$;
   b) choose $\theta_m^{\mathcal{I}(T)} \sim Dirichlet(\lambda\theta_{\ell(m)}^{\mathcal{I}(t_m)})$, if $m \in \mathcal{C}^{\mathcal{I}}$.

The graphical model representation of ROST (sites) is illustrated in Fig. 6.

Since we consider the asymmetric knowledge transfer setting in which the group-level rating patterns learned from the auxiliary domains are directly reused in the target domain, the mixing proportions of rating scales for user-item joint groups (i.e., ϕ) are fixed in the generative process of ROST (sites). The conditional distribution becomes

$$P\left(z_{nm}^{\mathcal{U}(T)} = k, z_{nm}^{\mathcal{I}(T)} = l | z_{\neg(nm)}^{\mathcal{U}(T)}, z_{\neg(nm)}^{\mathcal{I}(T)}, \mathbf{X}^{(T)}; \beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}}\right) \propto \phi_{k,l,\mathbf{X}_{nm}^{(T)}} \left(h_{nkT}^{\neg(nmT)} + \gamma_{nk}^{\mathcal{U}}\right) \left(h_{mlT}^{\neg(nmT)} + \gamma_{ml}^{\mathcal{I}}\right) \quad (7)$$

S1:	Input $\mathbf{X}^{(T)}$, K, L, $\alpha^{\mathcal{U}}$, $\alpha^{\mathcal{I}}$, β, λ, $\mathcal{C}^{\mathcal{U}}$, $\mathcal{C}^{\mathcal{I}}$, $iter$, and ϕ and $\{\theta^{\mathcal{U}(t)}, \theta^{\mathcal{I}(t)}\}_{t=1}^{T-1}$ learned from $\{\mathbf{X}^{(t)}\}_{t=1}^{T-1}$.
S2:	Sample $(z_{nm}^{\mathcal{U}(T)}, z_{nm}^{\mathcal{I}(T)})$ based on (7) for $iter$ epoches. One epoch means scanning $\mathcal{S}^{(T)}$ once.
S3:	Estimate $\mathbf{B}$, $\mathbf{P}^{(T)}$ and $\mathbf{Q}^{(T)}$ using (4), (10) and (11), respectively.
S4:	Predict a rating $\mathbf{X}_{nm}^{(T)}$ using (1).

Fig. 7. ROST (sites): training and prediction.

where

$$\gamma_{nk}^{\mathcal{U}} = \begin{cases} \alpha^{\mathcal{U}} & \text{if } n \notin \mathcal{C}^{\mathcal{U}} \\ \lambda\theta_{\ell(n),k}^{\mathcal{U}(t_n)} & \text{if } n \in \mathcal{C}^{\mathcal{U}} \end{cases} \quad (8)$$

$$\gamma_{ml}^{\mathcal{I}} = \begin{cases} \alpha^{\mathcal{I}} & \text{if } m \notin \mathcal{C}^{\mathcal{I}} \\ \lambda\theta_{\ell(m),l}^{\mathcal{I}(t_m)} & \text{if } m \in \mathcal{C}^{\mathcal{I}}. \end{cases} \quad (9)$$

Note in (7), we only sample the ratings in the target domain. The derivation of the conditional distribution (7) is similar to (3) except for fixing ϕ and using different prior parameters.

Then we can estimate the user/item-group membership matrices, $\mathbf{P}^{(T)}$ and $\mathbf{Q}^{(T)}$, as follows:

$$\mathbf{P}_{nk}^{(T)} = \theta_{n,k}^{\mathcal{U}(T)} = \frac{h_{nkT} + \gamma_{nk}^{\mathcal{U}}}{\sum_k \left(h_{nkT} + \gamma_{nk}^{\mathcal{U}}\right)} \quad (10)$$

$$\mathbf{Q}_{ml}^{(T)} = \theta_{m,l}^{\mathcal{I}(T)} = \frac{h_{mlT} + \gamma_{ml}^{\mathcal{I}}}{\sum_l \left(h_{mlT} + \gamma_{ml}^{\mathcal{I}}\right)} \quad (11)$$

where $\gamma_{nk}^{\mathcal{U}}$ and $\gamma_{ml}^{\mathcal{I}}$ are defined in (8) and (9), respectively. We do not estimate the group-level rating matrix $\mathbf{B}$ since we have obtained it from the auxiliary domains and can reuse it in the target domain. We sum up the modeling steps for ROST (sites) in Fig. 7.

We further compare the conditional distributions of ROST (sites) (7) and the basic ROST model (3). ROST (sites) directly reuses group-level rating patterns ϕ learned from auxiliary domains so the first term $\phi_{k,l,\mathbf{X}_{nm}^{(T)}}$ in (7) is fixed over Gibbs sampling epoches. The second term is proportional to the probability of selecting user-group k while the third term is proportional to the probability of selecting item-group l.

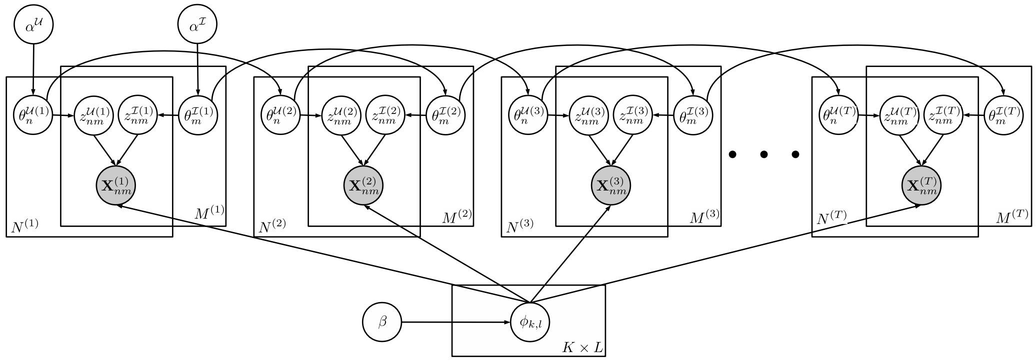


Fig. 8. Graphical model representation of ROST (time). The group-level rating patterns (ϕ) are shared by T temporal domains, while the user/item-group memberships of the current domain ($\theta^{\mathcal{U}(t)}/\theta^{\mathcal{I}(t)}$) depend on their corresponding users/items in the previous domain ($\theta^{\mathcal{U}(t-1)}/\theta^{\mathcal{I}(t-1)}$).

By making weighted user/item-group memberships of auxiliary domains as pseudo counts of ratings (prior knowledge) for corresponding users/items in the target domain, the users/items in $\mathcal{C}^{\mathcal{U}}/\mathcal{C}^{\mathcal{I}}$ are more likely to obtain similar group memberships as their corresponding users/items in auxiliary domains.

VI. Cross-Domain CF Over Time

Another cross-domain CF problem commonly found in real-world scenarios is CF over time, in which a time-slice is viewed as a temporal domain. This problem is motivated by the fact that users' interests may drift over time since they are continuously affected by moods, contexts, and pop culture trends. Thus, the ratings provided by the same user at different time may reflect different interests as those ratings are provided by different users. Based on this observation, we can assume that a user has multiple counterparts across temporal domains and the counterparts in successive temporal domains are different but closely related. In this section, we will extend the basic ROST model by taking into account the correspondences between user-counterparts in successive temporal domains and introduce ROST (time). The goal of ROST (time) is to model and track user-interest over time for better understanding users' preferences.

A. Problem Setting

Given an $N \times M$ rating matrix $\mathbf{X}$ whose element $\mathbf{X}_{nm}$ is associated with a time-stamp, indicating the time user n provided a rating on item m. We split the entire time span of the ratings in $\mathbf{X}$ into T equal time-slices and each time-slice is viewed as a temporal domain. Then we let $\mathbf{X}^{(t)}$ denote the snapshot rating matrix in temporal domain t and $\mathbf{X} = \{\mathbf{X}^{(t)}\}_{t=1}^{T}$, where $\mathbf{X}_{nm}^{(t)}$ denotes a rating provided by user n on item m at time-slice t, $(n, m) \in \mathcal{S}^{(t)}$ and $\mathcal{S}^{(t)}$ denotes the user-item set of the observed ratings in $\mathbf{X}^{(t)}$.

Although individual users' interests keep drifting, the interest distribution of a large population should be consistent over time. In other words, if we group user-counterparts based on their interests in each temporal domain, we will have same user groups in all temporal domains. Thus, a group-level rating matrix can also be shared across temporal domains. For user-group memberships, as they keep drifting, we let them change over time and, at the same time, impose dependence on them in successive temporal domains. We adopt a simple strategy to enable the underlying relatedness between them, that is to let the user-group memberships in the previous temporal domain be the Dirichlet priors in the current one: $\theta_n^{\mathcal{U}(t)} \sim Dirichlet(\lambda\theta_n^{\mathcal{U}(t-1)})$. This strategy can also be applied to imposing dependence on item-group memberships in successive temporal domains.[4]

B. ROST (Time)

For ROST (time), we aim to learn a group-level rating matrix $\mathbf{B}$ shared across T temporal domains and learn user/item-group memberships, $\{\mathbf{P}^{(t)}, \mathbf{Q}^{(t)}\}_{t=1}^{T}$, for the user/item-counterparts in T temporal domains. It is feasible to directly apply the basic ROST model to this problem by viewing T temporal domains be independent and user/item-counterparts in successive domains have no relatedness. However, since users usually have few ratings in each temporal domain, neglecting relatedness between user/item-counterparts in successive temporal domains may lead to overfitting. Thus, in ROST (time), in addition to sharing group-level rating patterns across all temporal domains, we also take into account the dependence between user/item-counterparts in successive temporal domains.

The generative process of ROST (time) is the same as the basic ROST model except for the user/item-group membership priors for all the user/item-counterparts:

1) for user-counterpart n in temporal domain t, choose $\theta_n^{\mathcal{U}(t)} \sim Dirichlet(\lambda\theta_n^{\mathcal{U}(t-1)})$;
2) for item-counterpart m in temporal domain t, choose $\theta_m^{\mathcal{I}(t)} \sim Dirichlet(\lambda\theta_m^{\mathcal{I}(t-1)})$.

λ is a tradeoff parameter: a larger λ will introduce more knowledge from the previous temporal domain to the current one (i.e., regularization) while a smaller λ will make the model concentrate more on the rating knowledge in the current domain (but may lead to overfitting). The graphical model representation of ROST (time) is illustrated in Fig. 8.

Since the hyper-parameters in the Dirichlet priors are themselves the latent variables to be inferred, we cannot marginalize out $\theta^{\mathcal{U}}$ and $\theta^{\mathcal{I}}$ as done for the basic ROST model. We resort to another approximation used in [16] and [17] by viewing

[4] To consider a general setting, we also introduce item-counterparts into ROST (time). In practice, item-character can be viewed static over time.

S1:	Input $\{\mathbf{X}^{(t)}\}_{t=1}^{T}$, K, L, $\alpha^{\mathcal{U}}$, $\alpha^{\mathcal{I}}$, β, λ, and $iter$.
S2:	Sample $(z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)})$ based on (12) for $iter$ epoches. One epoch means scanning $\{\mathcal{S}^{(t)}\}_{t=1}^{T}$ once.
S3:	Estimate $\mathbf{B}$, $\{\mathbf{P}^{(t)}\}_{t=1}^{T}$ and $\{\mathbf{Q}^{(t)}\}_{t=1}^{T}$ using (4), (15) and (16), respectively.
S4:	Predict a rating $\mathbf{X}_{nm}^{(t)}$ using (1).

Fig. 9. ROST (time): training and prediction.

$\theta^{\mathcal{U}}$ and $\theta^{\mathcal{I}}$ in the Dirichlet priors as another two sets of independent parameters, $\hat{\theta}^{\mathcal{U}}$ and $\hat{\theta}^{\mathcal{I}}$, which have the same values of $\theta^{\mathcal{U}}$ and $\theta^{\mathcal{I}}$, respectively. Then, we can marginalize out $\theta^{\mathcal{U}}$ and $\theta^{\mathcal{I}}$ and use the same Gibbs sampling algorithm of the basic ROST model for inference. The only difference between ROST (time) and the basic model is that, after each sampling epoch, $\hat{\theta}^{\mathcal{U}}$ and $\hat{\theta}^{\mathcal{I}}$ are updated with the new values of $\theta^{\mathcal{U}}$ and $\theta^{\mathcal{I}}$.

The conditional distribution of each latent variable pair $(z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)})$ in ROST (time) gives

$$P\Big(z_{nm}^{\mathcal{U}(t)} = k, z_{nm}^{\mathcal{I}(t)} = l | z_{\neg(nmt)}^{\mathcal{U}}, z_{\neg(nmt)}^{\mathcal{I}}, \mathbf{X}; \beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}}\Big) \propto \left(\frac{h_{kl\mathbf{X}_{nm}^{(t)}}^{\neg(nmt)} + \beta}{\sum_r h_{klr}^{\neg(nmt)} + R\beta}\right)\left(h_{nkt}^{\neg(nmt)} + \delta_{nkt}^{\mathcal{U}}\right)\left(h_{mlt}^{\neg(nmt)} + \delta_{mlt}^{\mathcal{I}}\right) \tag{12}$$

where

$$\delta_{nkt}^{\mathcal{U}} = \lambda \hat{\theta}_{n,k}^{\mathcal{U}(t-1)} \tag{13}$$

$$\delta_{mlt}^{\mathcal{I}} = \lambda \hat{\theta}_{m,l}^{\mathcal{I}(t-1)} \tag{14}$$

where $\theta_n^{\mathcal{U}(0)} = \alpha^{\mathcal{U}}$, $\theta_m^{\mathcal{I}(0)} = \alpha^{\mathcal{I}}$ for $t = 1$.

Then we can estimate the user/item-group membership matrices, $\{\mathbf{P}^{(t)}, \mathbf{Q}^{(t)}\}_{t=1}^{T}$, for the user/item-counterparts in T temporal domains as follows:

$$\mathbf{P}_{nk}^{(t)} = \theta_{n,k}^{\mathcal{U}(t)} = \frac{h_{nkt} + \delta_{nkt}^{\mathcal{U}}}{\sum_k \left(h_{nkt} + \delta_{nkt}^{\mathcal{U}}\right)} \tag{15}$$

$$\mathbf{Q}_{ml}^{(t)} = \theta_{m,l}^{\mathcal{I}(t)} = \frac{h_{mlt} + \delta_{mlt}^{\mathcal{I}}}{\sum_l \left(h_{mlt} + \delta_{mlt}^{\mathcal{I}}\right)} \tag{16}$$

where $\delta_{nkt}^{\mathcal{U}}$ and $\delta_{mlt}^{\mathcal{I}}$ are defined in (13) and (14), respectively. The group-level rating matrix $\mathbf{B}$ can be estimated using (4). We sum up the modeling steps for ROST (time) in Fig. 9.

We further compare the conditional distributions of ROST (time) (12) and the basic ROST model (3). ROST (time) has the same expression of the group-level rating patterns ϕ as that in (3). The second term is proportional to the probability of selecting user-group k while the third term is proportional to the probability of selecting item-group l. By making weighted user/item-group memberships of temporal domain $t-1$ as pseudo counts of ratings (prior knowledge) for corresponding user/item-counterparts in temporal domain t, user/item-group memberships are likely to change smoothly over all temporal domains.

Finally, we can track user n's interest-drift by investigating $\{[\mathbf{P}_n^{(1)}\mathbf{B}]^\top, \ldots, [\mathbf{P}_n^{(T)}\mathbf{B}]^\top\}$, where $\mathbf{P}_n^{(t)}$ denotes the nth row in $\mathbf{P}^{(t)}$. The column vector $[\mathbf{P}_n^{(t)}\mathbf{B}]^\top$ can be interpreted as the expected ratings provided by user n on L item groups (e.g., interests on L different movie themes) at time-slice t.

TABLE II
MODEL SELECTION (AVERAGE RMSE/MAE OVER 5 SPLITS)

(a) Numbers of user and item groups (K and L) selection for ROST

	$L=10$	$L=20$	$L=50$	$L=100$
$K=10$	0.921/0.729	0.921/0.730	0.921/0.729	0.922/0.730
$K=20$	0.921/0.730	**0.919/0.728**	0.922/0.730	0.923/0.731
$K=50$	0.922/0.730	0.920/0.728	0.923/0.731	0.927/0.735
$K=100$	0.921/0.730	0.920/0.729	0.925/0.733	0.931/0.738

(b) Weighting parameter λ selection for ROST (sites), $K=20, L=20$

$\lambda=0.1$	$\lambda=1$	$\lambda=10$	$\lambda=100$	$\lambda=1000$
0.924/0.732	0.924/0.730	0.922/0.729	**0.920/0.728**	0.920/0.730

(c) Weighting parameter λ selection for ROST (time), $K=20, L=20$

$\lambda=0.1$	$\lambda=1$	$\lambda=10$	$\lambda=100$	$\lambda=1000$
0.928/0.747	0.928/0.746	**0.926/0.743**	0.927/0.745	0.927/0.746

VII. EXPERIMENTS

We study the proposed ROST models empirically on three real-world recommendation data sets.

1) Validate that ROST (sites) can indeed transfer useful rating knowledge from auxiliary CF domains to alleviate the sparsity problem in the target domain and outperform the compared both single- and cross-domain CF methods.
2) Validate that ROST (time) can model user-interest drift over time and explicitly visualize the underlying drift to achieve state-of-the-art performance.
3) Demonstrate a case study to simultaneously share ratings over both sites and time-slices and illustrate some interesting behaviors of items revealed by the ROST models.

1) Evaluation Protocol: The evaluation metrics employed in our experiments are the root mean squared error (RMSE), $\sqrt{\sum_{i\in\mathcal{S}}(r_i - \hat{r}_i)^2/|\mathcal{S}|}$, and the mean absolute error (MAE), $(\sum_{i\in\mathcal{S}}|r_i - \hat{r}_i|)/|\mathcal{S}|$, where $\mathcal{S}$ denotes the set of test ratings, r_i the ground-truth rating and $\hat{r}_i$ the predicted rating. A smaller RMSE/MAE indicates better performance. Note that all the reported RMSE/MAE results in our experiments are the average performance over five random data splits.[5]

2) Model Selection: The parameters in the ROST models are selected as follows: for the hyper-parameters of the Dirichlet priors, we set $\alpha^{\mathcal{U}} = 1/K$, $\alpha^{\mathcal{I}} = 1/L$, and $\beta = 1/R$, according to [14]. The weighting parameter λ in the Dirichlet priors is selected in {0.1, 1, 10, 100, 1000} on a separate validation data set and we find that $\lambda = 100$ gives the best result for ROST (sites) and $\lambda = 10$ gives the best result for ROST (time). The numbers of users and item groups are selected on the separate validation data set and we find that $K = 20$ and $L = 20$ give the best result; the selected numbers are as same as those in [10]. The model selection results are shown

[5]We conduct t-test for each pair of methods based on five runs of results and claim that all the comparison results in Tables IV and V, and Fig. 10 are statistically significant at the 5% significance level.

TABLE III
RESULTS OF ROST (SITES) IN DIFFERENT AUXILIARY AND TARGET DOMAIN SETTINGS (AVERAGE RMSE / MAE OVER 5 SPLITS)

	target density 0.3%	target density 0.5%	target density 0.8%	target density 1.0%	target density 1.2%
1 auxiliary domain	*0.950 / 0.754*	0.932 / 0.738	0.919 / 0.727	0.916 / 0.724	0.911 / 0.720
2 auxiliary domains	*0.941 / 0.746*	0.927 / 0.734	0.918 / 0.726	0.913 / 0.721	0.910 / 0.719
3 auxiliary domains	***0.935* / *0.740***	**0.924** / **0.731**	**0.915** / **0.724**	**0.912** / **0.720**	**0.910** / **0.719**

TABLE IV
COMPARISON RESULTS FOR CROSS-DOMAIN CF OVER SITES (AVERAGE RMSE / MAE OVER 5 SPLITS)

	target density 0.3%	target density 0.5%	target density 0.8%	target density 1.0%	target density 1.2%
BPMF [6]	*0.952 / 0.753*	0.938 / 0.744	0.924 / 0.732	0.914 / 0.722	**0.910** / **0.718**
Bi-LDA [14]	*0.965 / 0.764*	0.947 / 0.751	0.934 / 0.742	0.922 / 0.731	0.914 / 0.722
TransClosure [37]	*0.970 / 0.769*	0.953 / 0.759	0.936 / 0.743	0.918 / 0.728	0.913 / 0.722
RMGM [10]	*0.944 / 0.747*	0.933 / 0.740	0.923 / 0.732	0.915 / 0.725	0.912 / 0.721
ROST ($\lvert\mathcal{C}^{\mathcal{I}}\rvert = 0$)	*0.947 / 0.750*	0.936 / 0.743	0.924 / 0.733	0.916 / 0.724	0.913 / 0.721
ROST ($\lvert\mathcal{C}^{\mathcal{I}}\rvert = 100$)	*0.943 / 0.746*	0.930 / 0.736	0.920 / 0.729	0.915 / 0.724	0.912 / 0.720
ROST ($\lvert\mathcal{C}^{\mathcal{I}}\rvert = 1000$)	***0.935* / *0.740***	**0.924** / **0.731**	**0.915** / **0.724**	**0.912** / **0.720**	**0.910** / 0.719

in Table II, where the separate validation data set is a subset of the data set used in Section VII-A.

3) Complexities: The computational complexity for training a ROST model is $\mathcal{O}(X(R+K+L)iter)$, where X is the number of all the observed ratings in $\{\mathbf{X}^{(t)}\}_{t=1}^{T}$, and $\mathcal{O}(R+K+L)$ is consumed for sampling from the mixing proportions of R rating scales, K user groups, and L item groups. The number of Gibbs sampling epoches *iter* is empirically set to 200, where the first 100 are for burn-in and the rest 100 are the samples for estimating model parameters. The spatial complexity is $\mathcal{O}(KLR+NKT+MLT+2X)$, which are allocated for three histograms h_{klr}, h_{nkt}, h_{mlt}, and $2X$ latent variables $\{z^{\mathcal{U}}, z^{\mathcal{I}}\}$ during the sampling process. All the tests of the ROST models are performed on a laptop with 2.67GHz CPU and 4G RAM. In our experiments, the empirical training time of a ROST model is $1 \sim 2$ minutes in MATLAB.

A. Cross-Domain CF Over Sites

1) Data: We use MovieLens 1M[6] to simulate cross-domain CF over sites. The entire data set comprises over $1M$ ratings provided by 6040 users on 3952 movies. In this experiment, we divide the entire data set into four parts as four different but related CF domains (one target domain and three auxiliary domains). We preprocess the data set as follows.

1) Randomly select 3020 users from the entire user set for each of the four CF domains. Randomly select 1976 movies from the entire movie set for each of the four CF domains. There may exist common users/items among these domains but we hide these correspondences using random permutations.
2) Construct four rating matrices based on the four selected user/item sets. View the first rating matrix as the target domain and the rest three as the auxiliary domains. Pick 1000 most popular movies (with most ratings) from the target domain and identify there correspondences (i.e., common movies) in the auxiliary domains to form the correspondence item set[7] $\mathcal{C}^{\mathcal{I}}$.

[6]http://www.grouplens.org/node/12

[7]We do not consider user correspondences in this experiment since user correspondence information is difficult to obtain in real-world scenarios.

As a result, we obtain four 3020×1976 rating matrices with a density around 5.9%. We investigate different density settings of the target domain by randomly selecting a subset of ratings from the target rating matrix to make its density in $\{0.3\%, 0.5\%, 0.8\%, 1.0\%, 1.2\%\}$ as the training set, and the rest ratings are used for test. It is worth noting that, in the proposed ROST models, we hide all user and item correspondences between the target domain and the auxiliary domains except a small set of item correspondences $\mathcal{C}^{\mathcal{I}}$. The identified movies with correspondences in $\mathcal{C}^{\mathcal{I}}$ are viewed as different but related items. We do not directly concatenate the two parts of ratings of a corresponding movie in our method.

2) Methods: We compare the following methods: 1) Bayesian probabilistic matrix factorization (BPMF) [6]; 2) Bi-LDA [14]; 3) transitive closure based on user-to-user similarities (TransClosure) [37]; 4) rating matrix generative model (RMGM) [10]; and 5) ROST (sites) with different sizes of $\mathcal{C}^{\mathcal{I}}$. BPMF is a state-of-the-art CF method which is often used as a baseline. Bi-LDA can be viewed as the single-domain version of the basic ROST model. TransClosure is a memory-based cross-domain CF method by augmenting paths between domains. RMGM is a cross-domain CF method based on the same rating knowledge transfer style as the ROST models do except for disregarding user/item dependence. The single-domain methods (BPMF and Bi-LDA) are trained only on the target domain while the cross-domain methods (TransClosure, RMGM, and ROST) are trained on both the auxiliary and the target domains.

3) Results: We first investigate the results of ROST (sites) in terms of number of auxiliary domains and density of the target domain in Table III. In both RMSE and MAE, we find that the ROST models trend to perform better as more auxiliary domains are involved. This result suggests that more related auxiliary domains can transfer more useful rating knowledge to the target domain. We can also observe that the performance improves as the density of the target domain increases. This observation implies that a denser target domain can be better matched to the auxiliary domains to acquire more rating knowledge. Another important result is that a more obvious performance gain can be achieved at a lower density (e.g., 0.3% in Table III) by incorporating more auxiliary domains.

Table IV reports the results of the compared methods in terms of density of the target domain. The first observation is that ROST ($|\mathcal{C}^{\mathcal{I}}| = 1000$) performs best in almost all cases; and it outperforms ROST ($|\mathcal{C}^{\mathcal{I}}| = 100$) while ROST ($|\mathcal{C}^{\mathcal{I}}| = 100$) further outperforms ROST ($|\mathcal{C}^{\mathcal{I}}| = 0$). The results imply that more item correspondences are beneficial for group-level rating pattern matching. The second observation is that the ROST models clearly outperform their single-domain version Bi-LDA. These results validate that related auxiliary domains indeed can transfer useful rating knowledge to the target domain to alleviate its sparsity problem. RMGM and ROST ($|\mathcal{C}^{\mathcal{I}}| = 0$) have similar performance since they use the same rating knowledge transfer style and have not considered item dependence between domains. TransClosure even performs worse than the single-domain method BPMF because memory-based method is sensitive to sparsity such that it may hardly augment paths. The last important result is that the superiority of the ROST models is more obvious if the target domain is relatively sparse (e.g., 0.3% in Table IV).

B. Cross-Domain CF Over Time

1) Data: We use Netflix[8] to simulate cross-domain CF over time. The entire data set comprises over 100M ratings provided by 480K users on 17K movies between 1999 and 2005. Each rating is associated with a time-stamp. To better investigate how users' interests drift over time, we conduct the following data preparation steps.[9]

a) Discard the ratings before 2002 and divide the remaining time span 2002.01 $\sim$ 2005.12 into 16 time-slices, each of which corresponds to three months, and associate each rating with a time-slice in $\{1, \ldots, 16\}$.
b) Select users who registered in Netflix before 2002 and were still active in 2005 (based on the time-stamps of their first and last ratings). Further select users who have more than 100 ratings in total and have at least 15 ratings in 4 time-slices. Obtain 6784 users.
c) Select the movies which were imported into Netflix before 2002. Further select the movies with more than 50 ratings. Obtain 3287 movies.

After data preprocessing, we obtain a 6784×3287 rating matrix (density 4.5%) whose elements are associated with 16 time-slices (temporal domains). We pick the first four temporal domains for validation and the rest twelve for evaluation.

2) Methods: We compare the following methods: 1) Bayesian probabilistic matrix factorization (BPMF) [6]; 2) Bayesian probabilistic tensor factorization (BPTF) [18]; 3) Bi-LDA [14]; 4) TimeSVD++ [7]; and 5) ROST (time). BPMF is a state-of-the-art CF method and BPTF is an extension of BPMF by considering the temporal dimension. Bi-LDA can be viewed as the single-domain version of the basic ROST model. TimeSVD++ is a well-known temporal CF method that won the Netflix prize. Among them, BPTF, TimeSVD++, and ROST (time) consider temporal information.

[8]http://www.netflix.com

[9]In the entire Netflix data set, 53.8% (or 73.7%) users were active only in half (or one) year, 80% ratings of 53.9% (or 74.8%) users were provided in three (or six) months. These observations suggest that the majority of users are not suitable for investigating interest drift during a long time period.

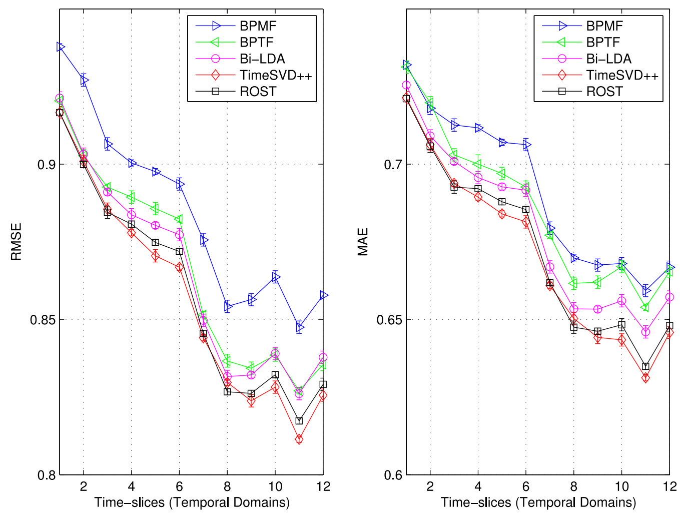


Fig. 10. Comparison results for cross-domain CF over time (12 temporal domains) in terms of RMSE (left) and MAE (right). The performance curves are average results over 5 splits with standard deviations.

3) Results: Fig. 10 plots the results of the compared methods over 12 time-slices. The two panels plot the performance curves in RMSE and MAE, respectively. We can see that ROST (time) achieves the similar performance as the well-known TimeSVD++ and outperforms other three compared methods in all 12 temporal domains. BPTF performs better than BPMF since it takes into account an additional temporal factor. The reason that ROST (time) outperforms BPTF and achieves comparable performance with TimeSVD++ might be that ROST (time) can learn time-dependent components for individual users/items as TimeSVD++ does while BPTF cannot. TimeSVD++ performs a little better than ROST (time) since it considers many effective heuristics in practice (e.g., time-dependent user bias). The curves of Bi-LDA and ROST (time) are similar in shape, which implies that ROST (time) is an enhanced version of Bi-LDA by taking into account temporal knowledge. The overall performance on the test data over 12 temporal domains indicates that ROST (time) can achieve state-of-the-art performance.

Apart from the state-of-the-art performance, a unique ability of ROST (time) is user-interest drift tracking. We can visualize user-interest drift based on the learning results of ROST (time). In Fig. 11, we plot some examples of user-group membership components (in Fig. 11(a), each subplot shows a matrix $\left[[\mathbf{P}_n^{(1)}]^\top, \ldots, [\mathbf{P}_n^{(12)}]^\top\right]$) and the accompanied user-interest components (in Fig. 11(b), each subplot shows a matrix $\left[[\mathbf{P}_n^{(1)}\mathbf{B}]^\top, \ldots, [\mathbf{P}_n^{(12)}\mathbf{B}]^\top\right]$). Through visualization, we can investigate how users switch their user-groups from time to time and change their interests over movie groups accordingly. These results show that ROST (time) can explicitly track and visualize user-interest drift over time.

C. Case Study

1) Data: Finally we conduct a case study of rating knowledge sharing over sites and time simultaneously on

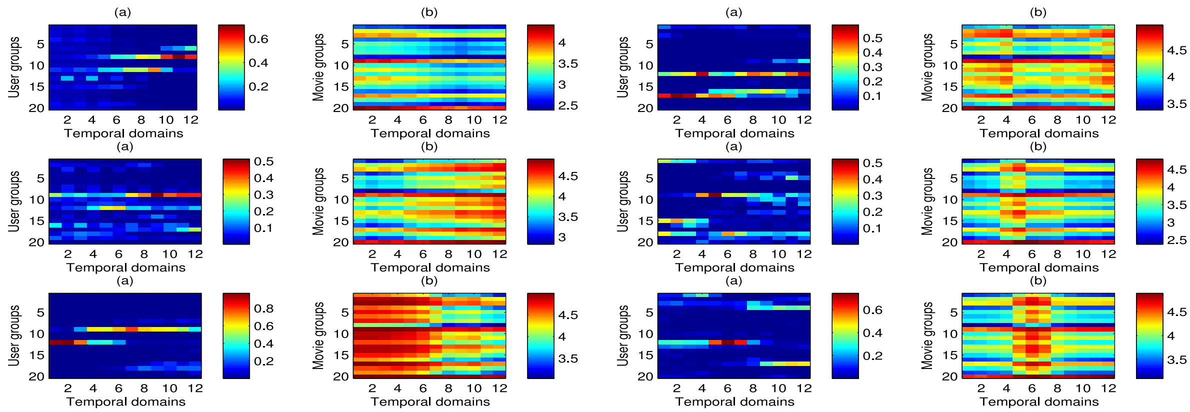


Fig. 11. (a) Examples of visualized user-group membership change (color scales indicate mixing proportions in [0, 1]), and its (b) accompanied user-interest drift (color scales indicate expected rating values in [1, 5]).

Douban [41], which is a composite social network data set containing user ratings on movies, music, and books. The original data set comprises 50*K* users and 7.45*M* ratings. For simplicity, we only investigate movies and music for three years, and preprocess the data as follows.

a) Divide the time span 2009.01 $\sim$ 2011.12 into 12 time-slices, each of which corresponds to three months. Associate each rating with a time-slice index in $\{1, \ldots, 12\}$.
b) Select users who have more than 100 ratings in total during the time span. Obtain 1616 users for the movie site and 675 users for the music site.
c) Select the movies and soundtracks with more than 100 ratings in total during the time span. Obtain 1580 movies and 741 soundtracks.

After data preprocessing, we obtain a 1616 $\times$ 1580 movie rating matrix, say $\mathbf{X}_{\text{movie}}$, with density 1%, and a 675 $\times$ 741 music rating matrix, say $\mathbf{X}_{\text{music}}$, with density 2.7%. The two rating matrices are treated as two sites without user/item overlaps.

2) Methods: We investigate our ROST model under three configurations.

a) ROST (sites) is performed on both $\mathbf{X}_{\text{movie}}$ and $\mathbf{X}_{\text{music}}$ disregarding time-slices.
b) ROST (time) is performed on $\mathbf{X}_{\text{movie}}$ and $\mathbf{X}_{\text{music}}$, respectively, incorporating time-slices.
c) ROST (sites+time) is performed on both $\mathbf{X}_{\text{movie}}$ and $\mathbf{X}_{\text{music}}$ as well as considering time-slices.

This can be enabled straightforwardly by sharing the group-level rating patterns $\mathbf{B}$ over both sites and temporal domains.[10] We also perform Bi-LDA [14] on $\mathbf{X}_{\text{movie}}$ and $\mathbf{X}_{\text{music}}$, respectively, as the baseline.

3) Results: The overall results in Table V show that ROST (sites) performs better than Bi-LDA due to sharing ratings across the two movie and music sites. ROST (time) also performs better than Bi-LDA because it captures users' varying interests. ROST (sites) performs a little better than ROST (time) implying that sharing ratings among multiple sites is relatively more helpful for improving rating predictions. As expected, ROST (sites+time) performs best due to incorporating both the advantages of ROST (sites) and ROST (time).

TABLE V
COMPARISON RESULTS OF THE CASE STUDY

	RMSE	MAE
Bi-LDA [14]	0.888 ± 0.004	0.720 ± 0.004
ROST (sites)	0.868 ± 0.004	0.679 ± 0.003
ROST (time)	0.874 ± 0.003	0.699 ± 0.002
ROST (sites+time)	$\mathbf{0.860} \pm 0.002$	$\mathbf{0.671} \pm 0.003$

Because both movies and music share the same group-level rating patterns $\mathbf{B}$, each movie or music has a distribution over $L = 20$ item groups for $T = 12$ time-slices, denoted by $\mathbf{Q}_m \in [0, 1]^{20\times 12}$. Thus we can compare items by calculating pairwise correlations based on $\{\mathbf{Q}_m\}$ of all the items in $\mathbf{X}_{\text{movie}}$ and $\mathbf{X}_{\text{music}}$. Some examples[11] of top matched pairs are listed in Table VI. Each pair is relevant in some aspects. For example (see Table VI), the movie/music in (a) are both French classical; the movies in (b) are both about Chinese Kung Fu; the music/movie in (c) are both about romance memory; and the movies in (d) are both Japanese samurai animations.

Based on the distribution matrix $\mathbf{Q}_m \in [0, 1]^{20\times 12}$, we can also investigate the drifting of item-group distribution for each movie/music. We use $\|\mathbf{Q}_m^{(2:12)} - \mathbf{Q}_m^{(1:11)}\|_{Fro}^2$ to measure the fluctuation of the distribution between consecutive time-slices (where $\mathbf{Q}_m^{(1:11)}$ denotes the columns $1 \sim 11$ in $\mathbf{Q}_m$). An interesting phenomenon we observed is that most top ranked items involve controversial topics, such as (see Table VII) homosexuality (a), scientific ethics (d), and politically sensitive issues (g). In particular, ROST (sites+time) obtains 0.847 and 0.806 in RMSE for predicting the ratings of (a) and (g), respectively, compared to 0.883 and 0.840 obtained by Bi-LDA. The performance gain of ROST (sites+time) on (a) and (g) is around -0.035 in RMSE while that on all the items is only -0.028 in RMSE (see Table V).

[10] For estimating $\mathbf{B}$ in this setting, we only need to count the ratings from any sites and any time-slices into h_{klr} in (4).

[11] The numbers in [ ] are movie/music IDs in www.douban.com. One can find the webpage of "La Haine (1995)" by searching "douban+1306449."

TABLE VI
SOME EXAMPLES OF THE TOP 30 MATCHED ITEM PAIRS

(a)	[movie-1306449] La Haine (1995) [music-3255195] Dans mon île (by Lisa Ono)
(b)	[movie-1294991] Fists of Fury (1972) [movie-1308927] Shaolin Popey 2: Messy Temple (1994)
(c)	[music-3335199] Smile (by Fiona Sit) [movie-3586471] Best of Times (2009)
(d)	[movie-1455442] The End of Evangelion (1997) [movie-1460915] Samurai Champloo (2004)
(e)	[music-1419616] Forrest Gump: The Soundtrack [music-5293263] ...Featuring (by Norah Jones)
(f)	[movie-1306802] Stuart Little 2 (2002) [movie-1309011] Robots (2005)
(g)	[movie-2252733] Kang Xi Wei Fu Si Fang Ji 1 (1997) [movie-2144485] Ke Wang (1990)
(h)	[movie-1291823] The Interpreter (2005) [music-6090760] Judas (by Lady Gaga)

TABLE VII
SOME EXAMPLES OF THE TOP 20 FLUCTUATING ITEMS

(a)	[music-3211009] I Kissed a Girl (by Katy Perry)
(b)	[movie-1294315] Tie Me Up! Tie Me Down! (1990)
(c)	[music-1397400] Hail to the Thief (by Radiohead)
(d)	[movie-1293073] The Fly (1986)
(e)	[music-1899400] The Black Parade (by My Chemical Romance)
(f)	[movie-1872272] An American Crime (2007)
(g)	[movie-3059379] Crossing (2008)
(h)	[movie-1292230] JFK (1991)

The above case study validates our hypothesis that user ratings are indeed constantly evolving, and item-groups across related sites are also highly correlated. Leveraging rating knowledge from related sites and further capturing user rating evolution over time, like ROST (time) and ROST (sites+time) do, provide effective solutions to tackle rating sparsity and user-interest drift for collaborative filtering in highly dynamic user environments.

VIII. CONCLUSION

In this paper, we proposed a unified framework for accommodating various cross-domain CF problems. Based on this unified framework, cross-domain CF over sites and time can be formulated as the same problem, that is, to learn from a set of related rating matrices by sharing group-level rating patterns and imposing user/item dependence across domains. We instantiated the proposed framework by developing a Bayesian generative model, say ROST, which can generate and predict ratings over the site-time coordinate system for different sites at different time. We also applied ROST (sites) and ROST (time) to address two challenges in real-world CF-based recommender systems, rating sparsity problem and user-interest drift problem, respectively. The experiments on a number of real-world recommendation data sets validated the effectiveness of the proposed models in addressing these two challenges. Our case study on simultaneously sharing ratings over both sites and time-slices illustrated some interesting behaviors of items revealed by the ROST models. Our framework open the opportunities to consider additional types of CF domains, such as contexts, to formulate complicated cross-domain CF problems.

In the future work, we will study how to make our ROST models selectively share useful rating knowledge across heterogeneous domains. To this end, we can introduce a link function to each domain for weighting and transforming rating patterns for sharing across heterogeneous domains. We will also extend the current Bayesian models to Bayesian nonparametric models which can automatically choose the numbers of latent user/item groups without empirical selection.

APPENDIX
GIBBS SAMPLING FOR ROST

According to the generative process of ROST described in Section IV, we have

$$P\left(\mathbf{X}_{nm}^{(t)}|z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)}, \phi\right) = \phi_{z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)}, \mathbf{X}_{nm}^{(t)}}$$

$$P\left(\phi_{k,l}|\beta\right) = \frac{\Gamma\left(R\beta\right)}{\prod_r \Gamma\left(\beta\right)} \prod_r \left[\phi_{k,l,r}\right]^{\beta-1}$$

$$P\left(z_{nm}^{\mathcal{U}(t)}|\theta^{\mathcal{U}(t)}\right) = \theta^{\mathcal{U}(t)}_{n, z_{nm}^{\mathcal{U}(t)}}$$

$$P\left(\theta_n^{\mathcal{U}(t)}|\alpha^{\mathcal{U}}\right) = \frac{\Gamma\left(K\alpha^{\mathcal{U}}\right)}{\prod_k \Gamma\left(\alpha^{\mathcal{U}}\right)} \prod_k \left[\theta_{n,k}^{\mathcal{U}(t)}\right]^{\alpha^{\mathcal{U}}-1}$$

$$P\left(z_{nm}^{\mathcal{I}(t)}|\theta^{\mathcal{I}(t)}\right) = \theta^{\mathcal{I}(t)}_{m, z_{nm}^{\mathcal{I}(t)}}$$

$$P\left(\theta_m^{\mathcal{I}(t)}|\alpha^{\mathcal{I}}\right) = \frac{\Gamma\left(L\alpha^{\mathcal{I}}\right)}{\prod_l \Gamma\left(\alpha^{\mathcal{I}}\right)} \prod_l \left[\theta_{m,l}^{\mathcal{I}(t)}\right]^{\alpha^{\mathcal{I}}-1}$$

where $\Gamma(\cdot)$ denotes Gamma function.

By plugging the above equations into the joint distribution of all the random variables in (2), we have

$$\begin{aligned}
&P\left(\mathbf{X}, z^{\mathcal{U}}, z^{\mathcal{I}}, \phi, \theta^{\mathcal{U}}, \theta^{\mathcal{I}}|\beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}}\right) \\
&\quad\propto \prod_{k,l}\prod_r \phi_{k,l,r}^{\sum_{n,m,t}\mathbf{1}\left(z_{nm}^{\mathcal{U}(t)}=k\right)\mathbf{1}\left(z_{nm}^{\mathcal{I}(t)}=l\right)\mathbf{1}\left(\mathbf{X}_{nm}^{(t)}=r\right)} \left[\phi_{k,l,r}\right]^{\beta-1} \\
&\quad\prod_n\prod_k\prod_t \left[\theta_{n,k}^{\mathcal{U}(t)}\right]^{\sum_m \mathbf{1}\left(z_{nm}^{\mathcal{U}(t)}=k\right)} \left[\theta_{n,k}^{\mathcal{U}(t)}\right]^{\alpha^{\mathcal{U}}-1} \\
&\quad\prod_m\prod_l\prod_t \left[\theta_{m,l}^{\mathcal{I}(t)}\right]^{\sum_n \mathbf{1}\left(z_{nm}^{\mathcal{I}(t)}=l\right)} \left[\theta_{m,l}^{\mathcal{I}(t)}\right]^{\alpha^{\mathcal{I}}-1}
\end{aligned}$$

where $\mathbf{1}(\cdot)$ denotes the indicator function.

By marginalizing out $\{\phi, \theta^{\mathcal{U}}, \theta^{\mathcal{I}}\}$, we have

$$\begin{aligned}
&P\left(\mathbf{X}, z^{\mathcal{U}}, z^{\mathcal{I}}|\beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}}\right) \\
&\quad\propto \prod_{k,l} \frac{\prod_r \Gamma\left(\sum_{n,m,t}\mathbf{1}(z_{nm}^{\mathcal{U}(t)}=k)\mathbf{1}\left(z_{nm}^{\mathcal{I}(t)}=l\right)\mathbf{1}\left(\mathbf{X}_{nm}^{(t)}=r\right)+\beta\right)}{\Gamma\left(\sum_r\sum_{n,m,t}\mathbf{1}\left(z_{nm}^{\mathcal{U}(t)}=k\right)\mathbf{1}\left(z_{nm}^{\mathcal{I}(t)}=l\right)\mathbf{1}\left(\mathbf{X}_{nm}^{(t)}=r\right)+R\beta\right)} \\
&\quad\prod_n\prod_t \frac{\prod_k \Gamma\left(\sum_m \mathbf{1}\left(z_{nm}^{\mathcal{U}(t)}=k\right)+\alpha^{\mathcal{U}}\right)}{\Gamma\left(\sum_k\sum_m \mathbf{1}\left(z_{nm}^{\mathcal{U}(t)}=k\right)+K\alpha^{\mathcal{U}}\right)} \\
&\quad\prod_m\prod_t \frac{\prod_l \Gamma\left(\sum_n \mathbf{1}\left(z_{nm}^{\mathcal{I}(t)}=l\right)+\alpha^{\mathcal{I}}\right)}{\Gamma\left(\sum_l\sum_n \mathbf{1}\left(z_{nm}^{\mathcal{I}(t)}=l\right)+L\alpha^{\mathcal{I}}\right)} \\
&\quad= \prod_{k,l}\frac{\prod_r \Gamma\left(h_{klr}+\beta\right)}{\Gamma\left(\sum_r h_{klr}+R\beta\right)} \\
&\quad\prod_n\prod_t \frac{\prod_k \Gamma\left(h_{nkt}+\alpha^{\mathcal{U}}\right)}{\Gamma\left(\sum_k h_{nkt}+K\alpha^{\mathcal{U}}\right)} \prod_m\prod_t \frac{\prod_l \Gamma\left(h_{mlt}+\alpha^{\mathcal{I}}\right)}{\Gamma\left(\sum_l h_{mlt}+L\alpha^{\mathcal{I}}\right)}.
\end{aligned}$$

For Gibbs sampling, we need to compute the conditional distribution of $\{z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)}\}$ given the rest of the latent variables $\{z_{\neg(nmt)}^{\mathcal{U}}, z_{\neg(nmt)}^{\mathcal{I}}\}$

$$\begin{aligned} P&\left(z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)} | z_{\neg(nmt)}^{\mathcal{U}}, z_{\neg(nmt)}^{\mathcal{I}}, \mathbf{X}; \beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}}\right) \\ &= \frac{P\left(z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)}, z_{\neg(nmt)}^{\mathcal{U}}, z_{\neg(nmt)}^{\mathcal{I}}, \mathbf{X}; \beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}}\right)}{P\left(z_{\neg(nmt)}^{\mathcal{U}}, z_{\neg(nmt)}^{\mathcal{I}}, \mathbf{X}; \beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}}\right)} \\ &\propto P\left(z_{nm}^{\mathcal{U}(t)}, z_{nm}^{\mathcal{I}(t)}, z_{\neg(nmt)}^{\mathcal{U}}, z_{\neg(nmt)}^{\mathcal{I}}, \mathbf{X}; \beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}}\right) \\ &\propto \prod_{k,l} \frac{\Gamma\left(h_{kl\mathbf{X}_{nm}^{(t)}} + \beta\right)}{\Gamma\left(\sum_r h_{klr} + R\beta\right)} \\ &\quad \prod_k \frac{\Gamma\left(h_{nkt} + \alpha^{\mathcal{U}}\right)}{\Gamma\left(\sum_k h_{nkt} + K\alpha^{\mathcal{U}}\right)} \prod_l \frac{\Gamma\left(h_{mlt} + \alpha^{\mathcal{I}}\right)}{\Gamma\left(\sum_l h_{mlt} + L\alpha^{\mathcal{I}}\right)}. \end{aligned}$$

The above equation can be further simplified by treating the terms not dependent on (k, l) as constants

$$\begin{aligned} P&\left(z_{nm}^{\mathcal{U}(t)} = k, z_{nm}^{\mathcal{I}(t)} = l | z_{\neg(nmt)}^{\mathcal{U}}, z_{\neg(nmt)}^{\mathcal{I}}, \mathbf{X}; \beta, \alpha^{\mathcal{U}}, \alpha^{\mathcal{I}}\right) \\ &\propto \frac{\Gamma\left(h_{kl\mathbf{X}_{nm}^{(t)}}^{\neg(nmt)} + \beta + 1\right)}{\Gamma\left(\sum_r h_{klr}^{\neg(nmt)} + R\beta + 1\right)} \\ &\quad \frac{\Gamma\left(h_{nkt}^{\neg(nmt)} + \alpha^{\mathcal{U}} + 1\right)}{\Gamma\left(\sum_k h_{nkt}^{\neg(nmt)} + K\alpha^{\mathcal{U}} + 1\right)} \frac{\Gamma\left(h_{mlt}^{\neg(nmt)} + \alpha^{\mathcal{I}} + 1\right)}{\Gamma\left(\sum_l h_{mlt}^{\neg(nmt)} + L\alpha^{\mathcal{I}} + 1\right)} \\ &\propto \frac{\left(h_{kl\mathbf{X}_{nm}^{(t)}}^{\neg(nmt)} + \beta\right)}{\left(\sum_r h_{klr}^{\neg(nmt)} + R\beta\right)} \left(h_{nkt}^{\neg(nmt)} + \alpha^{\mathcal{U}}\right) \left(h_{mlt}^{\neg(nmt)} + \alpha^{\mathcal{I}}\right) \end{aligned}$$

which is the conditional distribution in (3) for Gibbs sampling.

References

[1] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooper. Work (CSCW)*, Chapel Hill, NC, USA, 1994, pp. 175–186.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web (WWW)*, May 2001, pp. 285–295.

[3] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. 15th Conf. Uncertainty Artif. Intell. (UAI)*, 1999, pp. 289–296.

[4] L. Si and R. Jin, "Flexible mixture model for collaborative filtering," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, Washington, DC, USA, 2003, pp. 704–711.

[5] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, Washington, DC, USA, 2003, pp. 720–727.

[6] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, Helsinki, Finland, 2008, pp. 880–887.

[7] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proc. Int. Conf. Knowl. Discov. Data Mining (KDD)*, Paris, France, 2009, pp. 447–456.

[8] B. Li, Q. Yang, and X. Xue, "Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction," in *Proc. 21st Int. Joint Conf. Artif. Intell. (IJCAI)*, 2009, pp. 2052–2057.

[9] B. Li *et al.*, "Cross-domain collaborative filtering over time," in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI)*, 2011, pp. 2293–2298.

[10] B. Li, Q. Yang, and X. Xue, "Transfer learning for collaborative filtering via a rating-matrix generative model," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 617–624.

[11] T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in *Proc. 16th Int. Joint Conf. Artif. Intell. (IJCAI)*, 1999, pp. 688–693.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.

[13] M. Steyvers and T. Griffiths, "Probabilistic topic models," in *Handbook of Latent Semantic Analysis: A Road to Meaning*, T. Landauer, D. Mcnamara, S. Dennis, and W. Kintsch, Eds. Hillsdale, NJ, USA: Lawrence Erlbaum, 2007.

[14] I. Porteous, E. Bart, and M. Welling, "Multi-HDP: A non parametric Bayesian model for tensor factorization," in *Proc. 23rd Nat. Conf. Artif. Intell. (AAAI)*, 2008, pp. 1487–1490.

[15] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, Pittsburgh, PA, USA, 2006, pp. 113–120.

[16] X. Wei, J. Sun, and X. Wang, "Dynamic mixture models for multiple time series," in *Proc. 20th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2007, pp. 2909–2914.

[17] L. AlSumait, D. Barbará, and C. Domeniconi, "On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 3–12.

[18] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proc. SIAM Data Mining (SDM)*, 2010, pp. 211–222.

[19] Z. Lu, D. Agarwal, and I. S. Dhillon, "A spatio-temporal approach to collaborative filtering," in *Proc. 3rd ACM Conf. Recommender Syst.*, New York, NY, USA, Oct. 2009, pp. 13–20.

[20] B. Li, "Cross-domain collaborative filtering: A brief survey," in *Proc. 23rd IEEE Int. Conf. Tools Artif. Intell. (ICTAI)*, Boca Raton, FL, USA, Nov. 2011, pp. 1085–1086.

[21] O. Moreno, B. Shapira, L. Rokach, and G. Shani, "TALMUD: Transfer learning for multiple domains," in *Proc. 21st ACM Int. Conf. Inform. Knowl. Manag. (CIKM)*, 2012, pp. 425–434.

[22] S. Gao *et al.*, "Cross-domain recommendation via cluster-level latent factor model," in *Proc. ECML/PKDD*, Prague, Czech Republic, Sep. 2013, pp. 161–176.

[23] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[24] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, Corvallis, OR, USA, 2007, pp. 759–766.

[25] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, 2008, pp. 650–658.

[26] D. Zhou *et al.*, "Learning multiple graphs for document recommendations," in *Proc. 17th Int. Conf. World Wide Web*, Beijing, China, 2008, pp. 141–150.

[27] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, "Transfer learning in collaborative filtering for sparsity reduction," in *Proc. 24th Conf. Artif. Intell. (AAAI)*, 2010, pp. 230–235.

[28] W. Pan, N. N. Liu, E. W. Xiang, and Q. Yang, "Transfer learning to predict missing ratings via heterogeneous user feedbacks," in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI)*, 2011, pp. 2318–2323.

[29] L. Hu *et al.*, "Personalized recommendation via cross-domain triadic factorization," in *Proc. 22nd Int. Conf. World Wide Web (WWW)*, 2013, pp. 595–606.

[30] S. Berkovsky, T. Kuflik, and F. Ricci, "Cross-domain mediation in collaborative filtering," in *Proc. 11th Int. Conf. User Model. (UM)*, Corfu, Greece, Jul. 2007, pp. 355–359.

[31] B. Cao, N. N. Liu, and Q. Yang, "Transfer learning for collective link prediction in multiple heterogenous domains," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, 2010, pp. 159–166.

[32] Y. Zhang, B. Cao, and D.-Y. Yeung, "Multi-domain collaborative filtering," in *Proc. 26th Conf. Uncertainty Artif. Intell. (UAI)*, 2010, pp. 725–732.

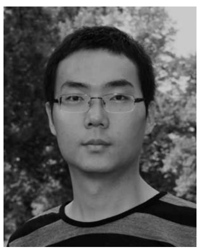
[33] Y. Shi, M. Larson, and A. Hanjalic, "Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering," in *Proc. 19th Int. Conf. UMAP*, Girona, Spain, Jul. 2011, pp. 305–316.

[34] B. Mehta and T. Hofmann, "Cross system personalization and collaborative filtering by learning manifold alignments," in *Proc. 29th Annu. Ger. Conf. Artif. Intell. KI*, Bremen, Germany, Jun. 2006, pp. 244–259.

[35] Y. Zhao, X. Feng, J. Li, and B. Liu, "Shared collaborative filtering," in *Proc. 5th ACM Conf. Rec. Syst.*, 2011, pp. 29–36.

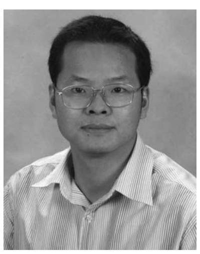
[36] B. Li, L. Chen, X. Zhu, and C. Zhang, "Noisy but non-malicious user detection in social recommender systems," *World Wide Web*, vol. 16, nos. 5–6, pp. 677–699, 2013.

[37] P. Cremonesi, A. Tripodi, and R. Turrin, "Cross-domain recommender systems," in *Proc. IEEE 11th Int. Conf. Data Mining Workshops (ICDMW)*, Vancouver, BC, Canada, 2011, pp. 496–503.
[38] T. Hofmann, J. Puzicha, and M. I. Jordan, "Learning from dyadic data," in *Proc. Adv. Neural Inf. Process Syst. (NIPS)*, 1999, pp. 466–472.
[39] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, 2006, pp. 126–135.
[40] G. Chen, F. Wang, and C. Zhang, "Collaborative filtering using orthogonal nonnegative matrix tri-factorization," in *Proc. 7th IEEE Int. Conf. Data Mining Workshops (ICDM)*, Omaha, NE, USA, Oct. 2007, pp. 303–308.
[41] E. Zhong, W. Fan, J. Wang, L. Xiao, and Y. Li, "ComSoc: Adaptive transfer of user behaviors over composite social network," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, 2012, pp. 696–704.



Bin Li received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2009.

He is a Lecturer with the Center for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Ultimo, NSW, Australia, since 2011. Prior to this, he was a Post-Doctoral Research Fellow at Institute TELECOM SudParis, Évry, France, from 2009 to 2010. His current research interests include machine learning and its applications to recommender systems and big data analytics.

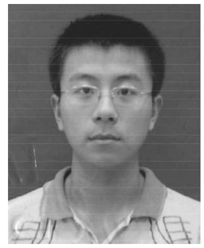


Xingquan Zhu (SM'12) received the Ph.D. degree in computer science from Fudan University, Shanghai, China.

He is an Associate Professor with the Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA. Prior to that, he was with the Center for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Ultimo, NSW, Australia.

His current research interests include data mining, machine learning, and multimedia systems. Since 2000, he has published over 170 refereed journal and conference papers in these areas.

Dr. Zhu is an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2008 to 2012 and since 2014. He received two Best Paper Awards and one Best Student Paper Award.



Ruijiang Li received the bachelor's degree in physics from Fudan University, Shanghai, China, in 2004, where he is currently pursuing the Ph.D. degree with the School of Computer Science, Fudan University, Shanghai.

His current research interests include matrix factorization, probabilistic models, and approximate inference methods.

Chengqi Zhang (SM'95) received the Ph.D. degree in computer science from the University of Queensland, Brisbane, QLD, Australia, in 1991, and the Dr.Sc. degree from Deakin University, Geelong, VIC, Australia, in 2002.

He is a Professor of Information Technology with the University of Technology, Sydney (UTS), Ultimo, NSW, Australia, where he has been the Director of the UTS Priority Investment Research Center for Quantum Computation and Intelligent Systems, since 2008.

He is a fellow of the Australian Computer Society. He was an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2005 to 2008, and has been as the General Chair, the Program Committee Chair, or the Organizing Chair for five international Conferences, including the IEEE International Conference on Data Mining and the Web Intelligence and Intelligent Agent Technology.