Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence

Cross-Domain Collaborative Filtering over Time

Bin Li†, Xingquan Zhu†, Ruijiang Li‡, Chengqi Zhang†, Xiangyang Xue‡, and Xindong Wu§♯
†QCIS Centre, FEIT, University of Technology, Sydney, NSW 2007, Australia
‡School of Computer Science, Fudan University, Shanghai 200433, China
§Department of Computer Science, Hefei University of Technology, Hefei 230009, China
♯Department of Computer Science, University of Vermont, Burlington, VT 05405, USA
bin.li-1@uts.edu.au, xqzhu@it.uts.edu.au, 081024031@fudan.edu.cn
chengqi@it.uts.edu.au, xyxue@fudan.edu.cn, xwu@cs.uvm.edu

Abstract

Collaborative filtering (CF) techniques recommend items to users based on their historical ratings. In real-world scenarios, user interests may drift over time since they are affected by moods, contexts, and pop culture trends. This leads to the fact that a user's historical ratings comprise many aspects of user interests spanning a long time period. However, at a certain time slice, one user's interest may only focus on one or a couple of aspects. Thus, CF techniques based on the entire historical ratings may recommend inappropriate items. In this paper, we consider modeling user-interest drift over time based on the assumption that each user has multiple counterparts over temporal domains and successive counterparts are closely related. We adopt the cross-domain CF framework to share the static group-level rating matrix across temporal domains, and let user-interest distribution over item groups drift slightly between successive temporal domains. The derived method is based on a Bayesian latent factor model which can be inferred using Gibbs sampling. Our experimental results show that our method can achieve state-of-the-art recommendation performance as well as explicitly track and visualize user-interest drift over time.

1 Introduction

In recommender systems, most collaborative filtering (CF) techniques infer users' preferences based on their historical ratings. Memory-based methods [Resnick *et al.*, 1994] find K-nearest neighbors who have similar historical interests with the active user. Model-based methods [Hofmann, 1999; Si and Jin, 2003; Porteous *et al.*, 2008] learn preference models for the users who have similar historical interests. Recommendations can thus be made based on the many aspects of user interests which were collected during a long time period.

However, in real-world scenarios, user interests are not static and may drift over time since they are continuously affected by moods, contexts, and pop culture trends. For example, a user recently under great working pressure may prefer comedies to relax him/herself, while he/she used to prefer suspenseful crime and thriller movies. The second example is that a user's interest is very likely to drift to sweet family movies when she is going to have a baby, though she has watched many horror movies before. Another example is that, although many people don't like animations, they may still have interests in emerging 3-D animations because of the fantastic 3-D visual effects. These observations show that, although many aspects of user interests can be found based on users' historical ratings, at a certain time slice, one user's interest may only focus on one or a couple of aspects. Thus, the static CF methods built on the entire historical ratings are inadequate to capture user-interest drift. In order to track user interests and create comprehensive user profiles such that different recommendation strategies can be used for consistent-taste users and changing-taste users, a CF method that can model user interests over time is required.

Recently, researchers have observed the underlying temporal dynamics in CF and some temporal and evolutionary CF methods have been reported [Koren, 2009; Liu *et al.*, 2010; Xiong *et al.*, 2010]. However, these methods do not target explicitly modeling user-interest drift. Their temporal features/components cannot be used for interpreting user interests. In order to explicitly model user-interest distribution over items, one possible approach is to use probabilistic topic models [Steyvers and Griffiths, 2007], which can model user (document) interests (topics) over items (words). But in this paper, we are given a more complex problem setting which involves a series of temporal domains. Two questions can thus be raised immediately: 1) what components in a rating matrix can be shared across CF temporal domains? and 2) how to model drifting user-interest components over time?

In this paper, we consider modeling user-interest drift over time based on the assumption that each user has multiple counterparts across temporal domains and the counterparts of successive temporal domains are closely related. We adopt the cross-domain CF framework [Li *et al.*, 2009] to share the static group-level rating matrix across temporal domains. The group-level rating matrix is a compressed rating pattern representation whose element can be viewed as an expected rating provided by a user prototype on an item prototype, while each real user/item is a convex combination of these prototypes. This answers the first question and the details are introduced in Section 3. In the group-level rating matrix, each user prototype (group) has a set of expected ratings on item prototypes (groups), which indeed reflect the interests of user groups over item groups. Since a real user is a convex combination of the user prototypes, we can easily obtain the individual users' interests over item groups. By modeling the relatedness of user-group memberships between successive temporal domains, we can capture user-interest drift. The derived method is based on a Bayesian latent factor model which can be inferred using collapsed Gibbs sampling. This answers the second question and the details are introduced in Section 4. The experimental results show that our method can achieve state-of-the-art recommendation performance, compared with the well-known TimeSVD++ [Koren, 2009]. Moreover, our method can explicitly track and visualize user-interest drift over time.

2 Problem Formulation

We are given an $N \times M$ rating matrix $\mathbf{X}$, each element $\mathbf{X}_{ij} \in \{1, \ldots, R\}$ denotes a rating provided by user u_i on item v_j, where $\{1, \ldots, R\}$ are rating scales. Besides, $\mathbf{X}_{ij}$ is also associated with a time-stamp. We split the whole time span of ratings into T time slices and let the matrix $\mathbf{Y}$ with the same size of $\mathbf{X}$ denote the corresponding time slice indices of ratings, where $\mathbf{Y}_{ij} \in \{1, \ldots, T\}$. We call these time slices temporal domains. Each temporal domain comprises a fraction of the entire rating data, we denote by $\mathbf{X}^{(t)}$ the ratings in the tth temporal domain, and $\mathbf{X} = \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(T)}\}$.

Through the whole time span of ratings, there are only N users and M items. However, as we argued in Section 1, the users in different temporal domains may have different interests and their interests can drift over temporal domains. Thus, we let each user u_i have T counterparts in T temporal domains, denoted by $\{u_i^{(1)}, \ldots, u_i^{(T)}\}$. One can simply understand this manipulation as that we have $M * T$ users. The key point is that each series of T counterparts for a certain user are temporally related. Similarly, we also let each item v_j have T counterparts $\{v_j^{(1)}, \ldots, v_j^{(T)}\}$ since items may also have slight character drift over time.

By splitting one user into T counterparts and modeling each counterpart separately, we can also capture different user interests in different temporal domains. However, in real-world scenarios, the rating data in each temporal domain for a certain user are very sparse. Learning a model for $u_i^{(t)}$ merely based on $\mathbf{X}^{(t)}$ is a big challenge. As we have observed that the user counterparts of successive temporal domains should be closely related, we resort to sharing rating knowledge across temporal domains for addressing the sparsity problem in individual temporal domains.

3 A Framework for Temporal-Domain CF

Looking from macroscopic view, the interest distribution of the large user population should be consistent over time. Although we have shown that individual users may have interest drift from time to time, the overall interest distribution of the whole user population should remain stable since different users' interest drift can be counterbalanced (i.e., some users' interests drift from $\mathcal{A}$ to $\mathcal{B}$ while some others' drift from $\mathcal{B}$ to $\mathcal{A}$). Thus, the user-interest distributions over items in different temporal domains are similar. In other words, if we group

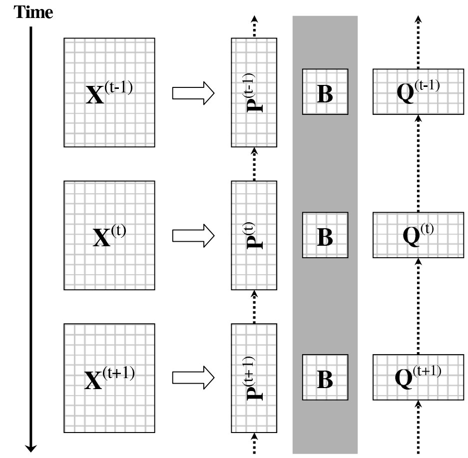

Figure 1: A framework for temporal-domain CF. Shaded rectangle means sharing and dotted arrows mean dependence.

users based on their interests, the user groups (prototypes) in different temporal domains should be same.

The above observation implies that, we can share the group-level rating knowledge across temporal domains while let the user-specific components drift over time and constrained by the temporal relatedness. This problem setting fits the cross-domain CF framework, i.e., rating-matrix generative model (RMGM) [Li *et al.*, 2009]. In RMGM, a group-level rating matrix $\mathbf{B}$ is learned from and shared across multiple related CF domains. Each element $\mathbf{B}_{kl}$ denotes the expected rating provided by user prototype k on item prototype l. Putting our problem in this framework, the temporal domains are related domains, and a group-level rating matrix $\mathbf{B}$ is shared across temporal domains. In addition, we also impose dependence for successive user/item components such that temporal relatedness of user-interest drift also can be captured. Our framework is illustrated in Figure 1.

In our temporal domain setting, a user counterpart $u_i^{(t)}$ is a convex combination of the K user prototypes and has a discrete distribution (membership) over K user groups, which is denoted by $\mathbf{p}_i^{(t)}$ and $\sum_k \mathbf{p}_i^{(t)}[k] = 1$. Similarly, an item counterpart $v_j^{(t)}$ also has a membership over L item groups, denoted by $\mathbf{q}_j^{(t)}$. Then a rating in $\mathbf{X}^{(t)}$ can be predicted by

$$\hat{\mathbf{X}}_{ij}^{(t)} = [\mathbf{p}_i^{(t)}]^\top \mathbf{B} \mathbf{q}_j^{(t)} \tag{1}$$

It is worth noting that the group-level rating matrix $\mathbf{B}$ is shared across all the T temporal domains while user/item-group memberships, $\mathbf{p}_i^{(t)}$ and $\mathbf{q}_j^{(t)}$, are domain-dependent.

Thus far, we should address the following two problems:

1. Learn the shared group-level rating matrix $\mathbf{B}$ based on all the ratings from T temporal domains.
2. Learn the user/item-group memberships, $\mathbf{p}_i^{(t)}$ and $\mathbf{q}_j^{(t)}$, in each temporal domain and impose temporal relatedness for memberships in successive temporal domains.

Given $\mathbf{B}$ (*static* over temporal domains) and $\mathbf{p}_i^{(t)}$ (*drifting* over temporal domains), we can then predict user u_i's interest over item groups at temporal domain t and track the drift over time. We address these two problems in Section 4.

4 Algorithm

4.1 Collaborative Filtering based on Bi-LDA

Bi-LDA [Porteous *et al.*, 2008] is a Bayesian latent factor model for matrix tri-factorization: $\hat{\mathbf{X}} = \mathbf{P}\mathbf{B}\mathbf{Q}^\top$, where $\mathbf{B}$ is a two-sided low-rank representation of $\mathbf{X}$, $\mathbf{P}$ and $\mathbf{Q}$ are membership matrices, where $\mathbf{P1} = \mathbf{1}$ and $\mathbf{Q1} = \mathbf{1}$

In collaborative filtering, given an $N \times M$ rating matrix $\mathbf{X}$, Bi-LDA can be used for co-clustering users and items by assigning each rating with a pair of latent factors $(z^{\mathcal{U}}, z^{\mathcal{I}})$ (i.e., the indices of the associated user/item groups). The co-clustering result gives a $K \times L$ group-level rating matrix $\mathbf{B}$, an $N \times K$ user-group membership matrix $\mathbf{P}$, and an $M \times L$ item-group membership matrix $\mathbf{Q}$.

Bi-LDA is a generative model, where the ratings in $\mathbf{X}$ can be generated in the following generative process:

1. For user-item joint group (k, l), choose $\mathbf{\Phi}_{kl} \sim Dirichlet(\beta)$;
2. For user u_i, choose $\mathbf{p}_i \sim Dirichlet(\alpha^{\mathcal{U}})$;
3. For item v_j, choose $\mathbf{q}_j \sim Dirichlet(\alpha^{\mathcal{I}})$;
4. For rating $\mathbf{X}_{ij}$
   - Choose a user group $z^{\mathcal{U}}_{ij} \sim Multinomial(\mathbf{p}_i)$
   - Choose an item group $z^{\mathcal{I}}_{ij} \sim Multinomial(\mathbf{q}_j)$
   - Choose a rating $\mathbf{X}_{ij} \sim Multinomial(\mathbf{\Phi}_{z^{\mathcal{U}}_{ij} z^{\mathcal{I}}_{ij}})$

where $\mathbf{\Phi}$ is a $K \times L \times R$ tensor and $\mathbf{\Phi}_{kl}$ is the rating-scale mixing proportion of user-item joint group (k, l) over $\{1, \ldots, R\}$ (i.e., $\sum_{r=1}^{R} \mathbf{\Phi}_{kl}[r] = 1$); $\mathbf{p}_i$ is the membership of user u_i over K user groups and $\mathbf{q}_j$ is the membership of item v_j over L item groups (i.e., $\sum_k \mathbf{p}_i[k] = 1$ and $\sum_l \mathbf{q}_j[l] = 1$); the ith row in $\mathbf{P}$ is $\mathbf{p}_i^\top$ and the jth row in $\mathbf{Q}$ is $\mathbf{q}_j^\top$.

We can use the collapsed Gibbs sampler for inferring the latent variables $\mathbf{z} = \{\mathbf{z}^{\mathcal{U}}, \mathbf{z}^{\mathcal{I}}\}$, where $\mathbf{z}^{\mathcal{U}} = \{z^{\mathcal{U}}_{ij}\}_{(i,j)=(1,1)}^{(N,M)}$ and $\mathbf{z}^{\mathcal{I}} = \{z^{\mathcal{I}}_{ij}\}_{(i,j)=(1,1)}^{(N,M)}$. Here we directly gives the conditional distribution for each latent variable pair $(z^{\mathcal{U}}_{ij}, z^{\mathcal{I}}_{ij})$

$$P(z^{\mathcal{U}}_{ij} = k, z^{\mathcal{I}}_{ij} = l | \mathbf{z}^{\neg(ij)}, \mathbf{X}) \propto \left(\frac{n_{klr}^{\neg(ij)} + \beta/R}{\sum_r n_{klr}^{\neg(ij)} + \beta}\right) \left(n_{ik}^{\neg(ij)} + \alpha^{\mathcal{U}}/K\right)\left(n_{jl}^{\neg(ij)} + \alpha^{\mathcal{I}}/L\right) \quad (2)$$

where $n_{klr}^{\neg(ij)}$ denotes the counter of all the ratings which fall in the cell (k, l, r), except for $\mathbf{X}_{ij}$. Similar definitions are applicable for the other counters in (2) and (7).

We can predict a rating in $\mathbf{X}$ by $\hat{\mathbf{X}}_{ij} = \mathbf{p}_i^\top \mathbf{B} \mathbf{q}_j$, where

$$\mathbf{B}_{kl} = \sum_r r \mathbf{\Phi}_{kl}[r] = \sum_r r \Big(\frac{n_{klr} + \beta/R}{\sum_r n_{klr} + \beta}\Big) \quad (3)$$

$$\mathbf{p}_i[k] = \frac{n_{ik} + \alpha^{\mathcal{U}}/K}{\sum_k n_{ik} + \alpha^{\mathcal{U}}}, \quad \mathbf{q}_j[l] = \frac{n_{jl} + \alpha^{\mathcal{I}}/L}{\sum_l n_{jl} + \alpha^{\mathcal{I}}} \quad (4)$$

$\mathbf{B}$ is the group-level rating matrix and $\mathbf{p}_i$ is the user-group membership of user u_i over K user groups.

4.2 Cross-Domain CF Over Time

As we have shown at the end of Section 3, our goal is to learn a group-level rating matrix $\mathbf{B}$ shared across temporal domains and $\mathbf{p}_i^{(t)}$ for user u_i's user-group membership at temporal domain t, for $i = \{1, \ldots, N\}$ and $t = \{1, \ldots, T\}$. If we simply view the counterparts of each user as different users, we can directly use Bi-LDA to simultaneously model T rating matrices $\{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(T)}\}$ within the cross-domain CF framework [Li *et al.*, 2009]. The result by doing so is that we can still learn $\mathbf{B}$ and $\{\mathbf{P}^{(1)}, \ldots, \mathbf{P}^{(T)}\}$, but the user-group memberships of a certain user's counterparts over temporal domains will have no relatedness. Besides, in this case, since the rating knowledge can not be transferred by explicitly connecting user counterparts of successive temporal domains, the data sparsity problem may become even severe.

To build the relatedness of a user's counterparts in successive temporal domains, we can impose dependence on user-group memberships of the successive counterparts. Given that our base algorithm is Bi-LDA which is a Bayesian model, a straightforward method for feeding knowledge is to use priors. We adopt a simple strategy that can transfer the user-group membership knowledge of user counterpart $u_i^{(t-1)}$ to $u_i^{(t)}$. That is to draw $\mathbf{p}_i^{(t)}$ from a Dirichlet prior parametrized by $\mathbf{p}_i^{(t-1)}$, i.e., $\mathbf{p}_i^{(t)} \sim Dirichlet(\mathbf{p}_i^{(t-1)})$. However, since membership vector is normalized (i.e., $\sum_k \mathbf{p}_i^{(t)}[k] = 1, \mathbf{p}_i^{(t)}[k] \leq 1$), which can make the prior distribution highly concentrated on few components. To avoid this, we introduce a weighting parameter λ to scale the prior

$$\mathbf{p}_i^{(t)} \sim Dirichlet(\lambda \mathbf{p}_i^{(t-1)}) \quad (5)$$

$$\mathbf{q}_j^{(t)} \sim Dirichlet(\lambda \mathbf{q}_j^{(t-1)}) \quad (6)$$

The intuition of $\lambda \mathbf{p}_i^{(t-1)}$ and $\lambda \mathbf{q}_j^{(t-1)}$ can be interpreted as the prior observed counts of ratings in K user groups before any rating from current temporal domain is observed [Steyvers and Griffiths, 2007; Wei *et al.*, 2007]. Due to the changes of the priors for generating user/item-group memberships, the conditional distribution (2) is changed accordingly

$$P(z^{\mathcal{U}}_{ij} = k, z^{\mathcal{I}}_{ij} = l | \mathbf{z}^{\neg(ij)}, \mathbf{X}, \mathbf{Y}) \propto \left(\frac{n_{klr}^{\neg(ij)} + \beta/R}{\sum_r n_{klr}^{\neg(ij)} + \beta}\right) \left(n_{ikt}^{\neg(ij)} + \lambda \mathbf{p}_i^{(t-1)}[k]\right)\left(n_{jlt}^{\neg(ij)} + \lambda \mathbf{q}_j^{(t-1)}[l]\right) \quad (7)$$

where the counters n_{ikt} and n_{jlt} get an additional temporal dimension. Note that the first term remains the same as that in (2) since the group-level rating matrix $\mathbf{B}$ is independent of time and shared across temporal domains. So in the cross-temporal-domain setting, $\mathbf{B}$ can still be constructed by using (3). While user/item-group memberships for user/item counterparts over temporal domains are constructed differently

$$\mathbf{p}_i^{(t)}[k] = \frac{n_{ikt} + \lambda \mathbf{p}_i^{(t-1)}[k]}{\sum_k n_{ikt} + \lambda \sum_k \mathbf{p}_i^{(t-1)}[k]} \quad (8)$$

$$\mathbf{q}_j^{(t)}[l] = \frac{n_{jlt} + \lambda \mathbf{q}_j^{(t-1)}[l]}{\sum_l n_{jlt} + \lambda \sum_l \mathbf{q}_j^{(t-1)}[l]} \quad (9)$$

The main idea of the proposed approximation inference strategy for CF across temporal domains is straightforward. We name it rating-matrix generative model over time (RMGM-OT), which comprises the following two alternating steps:

Step 1 [Cross-Domain CF] Perform one epoch of Gibbs sampling over $\{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(T)}\}$ using (7) by viewing all the user/item counterparts as different users/items (i.e., RMGM over temporal domains). Go to Step 2.

Step 2 [Time-Dependent Priors Update] After an epoch of Gibbs sampling, update $\{\mathbf{P}^{(1)}, \ldots, \mathbf{P}^{(T)}\}$ and $\{\mathbf{Q}^{(1)}, \ldots, \mathbf{Q}^{(T)}\}$ using (8) and (9), respectively. These will be the user/item-group membership priors for the next Gibbs sampling epoch. Go to Step 1.

In the first Gibbs sampling epoch, the user/item-group membership prior parameters are initialized as $\alpha^{\mathcal{U}}$ and $\alpha^{\mathcal{I}}$. The prior for the mixing proportion of rating scales of user-item joint group is set to β through all the Gibbs sampling epoches. In practice, the Gibbs sampler converges after hundreds of epoches. After inference, we can construct $\mathbf{B}$ using (3), and predict the rating $\mathbf{X}_{ij}^{(t)}$, which is provided by user u_i on item v_j in temporal domain t, by using (1).

Modeling User-Interest Drift

With $\mathbf{B}$ and $\{\mathbf{P}^{(1)}, \ldots, \mathbf{P}^{(T)}\}$, we can finally model user-interest drift over time. The group-level rating matrix $\mathbf{B}$ reflects each user prototype's interest over item groups, while $\mathbf{p}_i^{(t)}$ is the user-group membership for user u_i at temporal domain t. Thus, the user-interest component of user u_i at temporal domain t should be $\mathbf{B}^\top \mathbf{p}_i^{(t)}$. By constructing all the user-interest components in T temporal domains for user u_i, we can track user u_i's interest drift over time

$$\mathbf{D}_i = \left[\mathbf{B}^\top \mathbf{p}_i^{(1)}, \ldots, \mathbf{B}^\top \mathbf{p}_i^{(T)}\right] \tag{10}$$

where $\mathbf{D}_i$ is an $L \times T$ interest-drift matrix for user u_i over T temporal domains. Each column in $\mathbf{D}_i$ can be interpreted as the expected ratings provided by user u_i on all the L item groups at temporal domain t. If we list some top items in each item group, we can intuitively find which kinds of items that a user may like most at certain time slices.

5 Related Work

The proposed method is based on the latent factor model (LFM). The first well-known work which applied LFM for collaborative filtering is probabilistic Latent Semantic Analysis (pLSA) [Hofmann, 1999]. Later, the LFMs with two sets of latent variables, associated to users and items respectively, were also proposed for CF [Si and Jin, 2003; Porteous *et al.*, 2008]. Recently, many Bayesian extensions of pLSA [Steyvers and Griffiths, 2007] have been extensively adopted for CF, including Bi-LDA [Porteous *et al.*, 2008], which is used as the base algorithm in our method.

Our work is also related to temporal and evolutionary CF methods. The most well-known work in this line is TimeSVD++ [Koren, 2009], in which a user/item feature is a combination of one static component and one time-dependent component. Online evolutionary CF [Liu *et al.*, 2010] is memory-based and able to run in incremental mode. [Xiong *et al.*, 2010] views temporal domain as a third dimension and use tensor factorization to factorize the temporal components. [Khoshneshin and Street, 2010] dynamically assigns users/items to different clusters based on evolutionary co-clustering. However, the temporal features/components in these methods cannot be interpreted for user interests.

Another related topic is transfer learning. We have shown in Section 3 that the proposed method is within the cross-domain CF framework [Li *et al.*, 2009] in which a group-level rating matrix is shared. Moreover, we impose temporal dependence on user/item memberships such that temporal knowledge can be taken into account. Our method can combine rating data from multiple domains and simultaneously learn model for each domain by sharing parameters, so it fits the multi-task learning setting [Pan and Yang, 2010]. [Zheng *et al.*, 2008] also consider knowledge transfer over temporal domains, but it is based on hidden Markov models.

Finally, our work is related to user-interest tracking in CF. Few works have been reported in this line. [Ma *et al.*, 2007] is one such work but it is a hybrid CF method using content information. To the best of our knowledge, the proposed method is the first work that can *explicitly* model user-interest drift over item groups in collaborative filtering.

6 Experiments

6.1 Data Preparing

We use the Netflix[1] prize data set in our experiments. The entire data set comprises over 100 million ratings provided by ∼480,000 users on ∼17,000 movies between 1999 and 2005. Each rating is associated with a time-stamp. In order to vividly demonstrate the drifting of user interests over time, we conduct the following data preprocessing steps:

1. We discard the ratings before 2002 and split the remaining time span (Jan 2002 – Dec 2005) into 16 equal time slices. Each time slice corresponds to exact three months. Then, we replace the original time-stamp of each rating with the time-slice index in $\{1, \ldots, 16\}$.

2. We only consider the users who registered in Netflix before 2002 and were still active in 2005 (based on the time-stamps of their first/last rating). We further select the users who have more than 100 ratings in total and have at least 15 ratings in 4 time slices. As a result, we obtain 6784 users for our experiments.

3. We only consider the movies which were imported into Netflix before 2002. We further select the movies whose associated ratings are more than 50. As a result, we obtain 3287 movies in our experiments.

After data pruning, we obtain a 6784×3287 rating matrix whose elements are associated to 16 time slices (temporal domains). We pick the first 4 temporal domains for validation and the remaining 12 temporal domains for evaluation.

[1] www.netflix.com

Table 1: Parameter selection for (K, L) and λ.

$(K, L)\backslash\lambda$	0.1	1	10	100	1000
$(20, 20)$	0.970	0.970	0.968	0.967	0.969
$(20, 50)$	0.976	0.974	0.973	0.971	0.974
$(20, 100)$	0.980	0.979	0.977	0.975	0.979
$(50, 20)$	0.974	0.973	0.970	0.969	0.972
$(50, 50)$	0.978	0.977	0.976	0.973	0.975
$(50, 100)$	0.984	0.984	0.978	0.977	0.979
$(100, 20)$	0.976	0.975	0.972	0.971	0.973
$(100, 50)$	0.983	0.980	0.979	0.976	0.978
$(100, 100)$	0.989	0.988	0.984	0.982	0.984

6.2 Evaluation Protocol

We randomly extract 20% ratings from the evaluation data set for training (density 0.9%) while the remaining ratings are used for test. This procedure is repeated for 5 times and all the reported results in Figure 2 and Table 2 are the average performance over 5 trials. The performance evaluation metric we adopt in our experiments is the root mean squared error (RMSE): $\sqrt{\sum_{i\in\mathcal{S}}(r_i - \hat{r}_i)^2/|\mathcal{S}|}$, where $\mathcal{S}$ denotes the set of test ratings, r_i is the ground truth and $\hat{r}_i$ is the predicted rating. A smaller value of RMSE indicates a better performance.

6.3 Parameter Selection

We simply set the hyper-parameters of the Dirichlet priors, $\alpha^{\mathcal{U}}$, $\alpha^{\mathcal{I}}$, and β, to 1, according to [Porteous *et al.*, 2008].

The numbers of user and item groups, K and L, are selected in $\{20, 50, 100\}$ and the results are in Table 1. We find that the performance reduces gradually as K and L become larger. This result may be caused by model overfitting if the latent dimensions are too many. We set $K = 20$ and $L = 20$.

Another parameter λ is used for tuning the concentration of the Dirichlet priors. We select λ in $\{0.1, 1, 10, 100, 1000\}$ and report the results in Table 1. We find that the performance increases gradually until $\lambda = 100$ and then reduces a little, which implies λ is a tradeoff parameter: A larger λ will feed more knowledge of domain $t-1$ to domain t and smooth $\mathbf{P}^{(t)}$ and $\mathbf{Q}^{(t)}$ in all temporal domains, but it may discard too much rating knowledge of the current domain t; while a smaller λ will make the model of domain t emphasize the data in domain t and absorb less knowledge from domain $t-1$, but it may lead to model overfitting. We set $\lambda = 100$.

6.4 Performance Comparison

We compare the performance of the following methods: 1) Weighted low-rank approximations (WLRA) [Srebro and Jaakkola, 2003], 2) Bi-LDA [Porteous *et al.*, 2008], 3) TimeSVD++ [Koren, 2009] (temporal), and 4) our method RMGM-OT (temporal). WLRA is an effective yet simple method which is broadly adopted as a baseline. Bi-LDA is the non-temporal version of our method. TimeSVD++ is the state-of-the-art temporal CF technique and can thus be the baseline of temporal CF methods.

Figure 2 and Table 2 report the comparison results on the evaluation data set. In Figure 2, we calculate RMSE in each temporal domain and plot the performance curves of the compared methods. We find that the four methods have similar performance in the first 6 temporal domains and widen the difference in the last 6 temporal domains. The curves of Bi-LDA and RMGM-OT are very similar. TimeSVD++ performs a little worse than our method on this data set. The overall performance over all 12 temporal domains in Table 2 also gives the same result, i.e., Bi-LDA and RMGM-OT have the best performance. The experimental results show that our method can achieve state-of-the-art performance.

Table 2: Overall performance comparison.

Methods	RMSE (Mean±Std)
WLRA	0.942±0.001
Bi-LDA	0.918±0.004
TimeSVD++	0.929±0.002
RMGM-OT	0.919±0.002

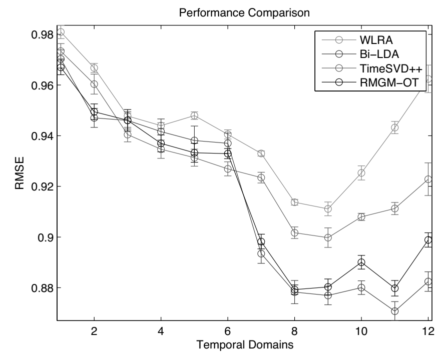

Figure 2: Performance comparison over temporal domains.

6.5 User-Interest Drift Analysis

Finally, we visualize the RMGM-OT learning results to show that the proposed method can indeed capture users' interest drift over temporal domains. In Figure 3, we plot some user-group membership components $\mathbf{P}^\top = [\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(12)}]$ (upper row, each subplot shows a $\mathbf{P}^\top$) and the corresponding user-interest components $\mathbf{D} = [\mathbf{B}^\top\mathbf{p}^{(1)}, \ldots, \mathbf{B}^\top\mathbf{p}^{(12)}]$ (bottom row, each subplot shows a $\mathbf{D}$). From these plots, we can investigate how user switch their user groups from time to time and change their interests over movie groups accordingly. For example, the second user switches his/her group membership between user groups 13 and 16. When he/she belongs more to user group 13, he/she would show more interest in movie group 5. The third user spans three user groups. When he/she belongs more to user group 17, he/she would show a broad interest in most movie groups.

We can also observe that, a user's user-group memberships in successive temporal domains are smooth. In other words, users don't switch their groups frequently. This phenomenon should be the result of the imposed dependence between successive user-group memberships (5).

7 Conclusion

In this paper, we proposed a cross-domain CF method over temporal domains, in which each user has multiple counterparts over temporal domains and successive counterparts are

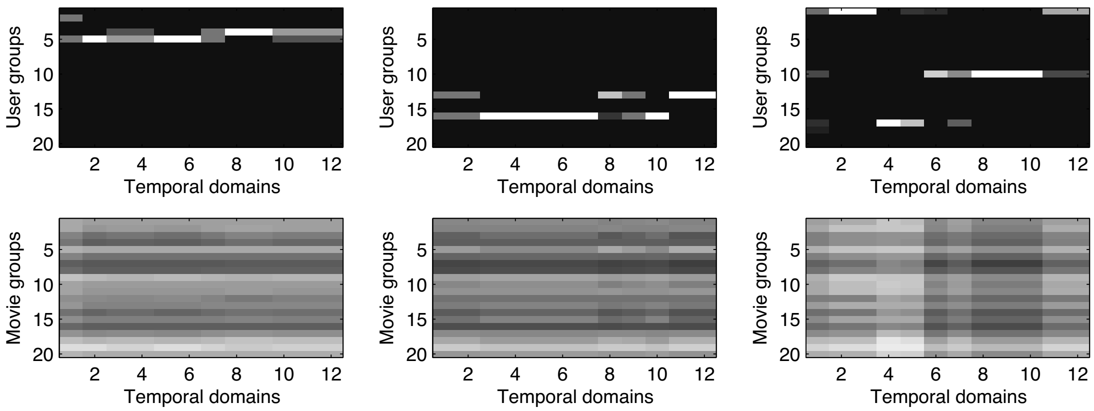

Figure 3: Some examples of user-group membership drift (upper row, the gray scales indicate membership proportions) and corresponding user-interest drift over movie groups (bottom row, gray scales indicate expected rating values in $[1, 5]$).

closely related. We adopted the cross-domain CF framework to share the static group-level rating matrix across temporal domains while let user-group memberships drift slightly between successive temporal domains. The derived method is based on a latent factor model. The experimental results have shown that our method can not only achieve state-of-the-art recommendation performance but also explicitly track and visualize user-interest drift over time.

8 Acknowledgments

This work was supported by Australian Research Council's Discovery Project (No. DP1093762) and Future Fellowship (No. FT100100971), the National 973 Program of China (Nos. 2010CB327906 and 2009CB326203), and the NSF of China (Nos. 60873178 and 60828005).

References

[Hofmann, 1999] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proc of the 15th Conf. on Uncertainty in Artificial Intelligence*, pages 289–296, 1999.

[Khoshneshin and Street, 2010] Mohammad Khoshneshin and W. Nick Street. Incremental collaborative filtering via evolutionary co-clustering. In *Proc. of the Fourth ACM Conf. on Recommender Systems*, pages 325–328, 2010.

[Koren, 2009] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 447–456, 2009.

[Li *et al.*, 2009] Bin Li, Qiang Yang, and Xiangyang Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proc. of the 26th Int'l Conf. on Machine Learning*, pages 617–624, 2009.

[Liu *et al.*, 2010] Nathan N. Liu, Min Zhao, Evan Xiang, and Qiang Yang. Online evolutionary collaborative filtering. In *Proc. of the Fourth ACM Conf. on Recommender Systems*, pages 95–102, 2010.

[Ma *et al.*, 2007] Shanle Ma, Xue Li, Yi Ding, and Maria E. Orlowska. A recommender system with interest-drifting. In *Proc. of the 8th Int'l Conf. on Web Information Systems Engineering*, pages 633–642, 2007.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[Porteous *et al.*, 2008] Ian Porteous, Evgeniy Bart, and Max Welling. Multi-HDP: A non parametric Bayesian model for tensor factorization. In *Proc. of the 23rd National Conf. on Artificial Intelligence*, pages 1487–1490, 2008.

[Resnick *et al.*, 1994] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. of the ACM Conf. on Computer Supported Cooperative Work*, pages 175–186, 1994.

[Si and Jin, 2003] Luo Si and Rong Jin. Flexible mixture model for collaborative filtering. In *Proc. of the 20th Int'l Conf. on Machine Learning*, pages 704–711, 2003.

[Srebro and Jaakkola, 2003] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proc. of the 20th Int'l Conf. on Machine Learning*, pages 720–727, 2003.

[Steyvers and Griffiths, 2007] Mark Steyvers and Tom Griffiths. Probabilistic topic models. In *Handbook of Latent Semantic Analysis: A Road to Meaning*. Lawrence Erlbaum, 2007.

[Wei *et al.*, 2007] Xing Wei, Jimeng Sun, and Xuerui Wang. Dynamic mixture models for multiple time series. In *Proc of the 20th Int'l Joint Conf. on Artificial Intelligence*, pages 2909–2914, 2007.

[Xiong *et al.*, 2010] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G. Carbonell. Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In *Proc. of the SIAM Int'l Conf. on Data Mining*, pages 211–222, 2010.

[Zheng *et al.*, 2008] Vincent W. Zheng, Evan W. Xiang, Qiang Yang, and Dou Shen. Transferring localization models over time. In *Proc. of the 23rd National Conf. on Artificial Intelligence*, pages 1421–1426, 2008.