Cross-domain recommendation without shared users or items by sharing latent vector distributions

Tomoharu Iwata **Koh Takeuchi**
NTT Communication Science Laboratories

Abstract

We propose a cross-domain recommendation method for predicting the ratings of items in different domains, where neither users nor items are shared across domains. The proposed method is based on matrix factorization, which learns a latent vector for each user and each item. Matrix factorization techniques for a single-domain fail in the cross-domain recommendation task because the learned latent vectors are not aligned over different domains. The proposed method assumes that latent vectors in different domains are generated from a common Gaussian distribution with a full covariance matrix. By inferring the mean and covariance of the common Gaussian from given cross-domain rating matrices, the latent factors are aligned, which enables us to predict ratings in different domains. Experiments conducted on rating datasets from a wide variety of domains, e.g., movie, books and electronics, demonstrate that the proposed method achieves higher performance for predicting cross-domain ratings than existing methods.

1 Introduction

Recommender systems are widely used in online stores because they help users to find the users' favorite items from the huge number of items available. They usually suggest items to users in a single domain, e.g. the items available in that store. However, if a store recommends items from different stores, the store can improve its profit. Moreover, presenting items that are not those of the store can allow users to find unexpected, but well-appreciated, items through serendipity [31].

To create this functionality, a number of cross-domain recommendation methods have been proposed [6, 8, 14, 23]. Most of the methods assume that there are shared users or items. However, shared users and items might be unavailable if the different domains have totally different items, different stores use different databases, and the sharing of user information might be restricted to preserve privacy.

Our task in this paper is to recommend items in different domains, where neither users nor items are shared across the domains, as shown in Figure 1. For this task, we propose a cross-domain recommendation method based on matrix factorization [18, 25], which has been successfully used for single-domain recommendation. Matrix factorization learns a latent vector for each user and each item given a user-item rating matrix, and predicts missing ratings using the latent vectors. However, standard matrix factorization techniques fail to predict cross-domain ratings in the absence of shared users and items, because the latent vectors are inferred individually in each domain, and the latent factors are not aligned across different domains.

The proposed method assumes that latent vectors in different domains are generated from a common Gaussian distribution with an unknown mean vector and full covariance matrix as shown in Figure 2. By inferring the shared mean and covariance of the Gaussian from given cross-domain rating matrices, the latent factors are aligned across different domains, which enables us to predict ratings in different domains.

When two domains are related, e.g., movie and book ratings, their latent factors would be similar. Suppose that each latent factor represents a genre; for example a high value of a factor in an item latent vector indicates the item is SF movie, and a high value of the factor in a user latent vector indicates the user likes SF movie. We can expect that users who like SF movie also like SF books. By aligning latent factors, e.g.,

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

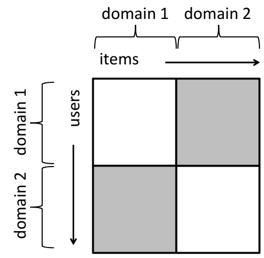


Figure 1: Two domain rating matrices without shared users or items. Ratings in unshaded parts can be observed, and those in shaded parts are missing. The task in our paper is, given rating matrices without shared users or items, to predict missing ratings in off-diagonal blocks.

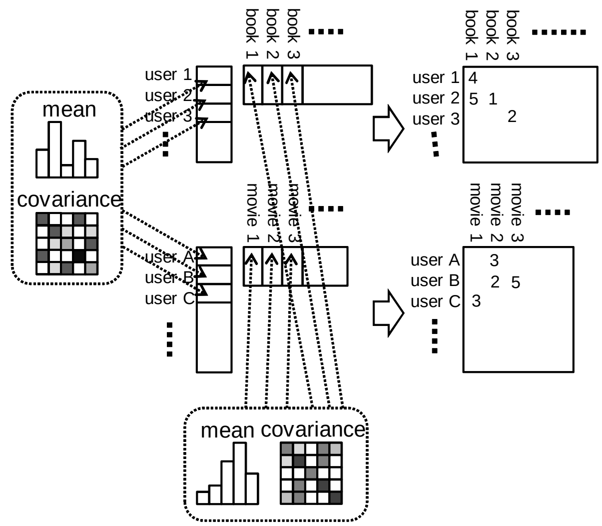


Figure 2: The proposed method is based on matrix factorization, where mean and covariance structures of latent vectors are shared across different domains. The first domain contains items (book1, book2, $\cdots$) and users (user1, user2, $\cdots$). The second domain contains items (movie1, movie2, $\cdots$) and users (userA, userB, $\cdots$). The task is to predict ratings in different domains, e.g., a rating of userA to book1, and that of user2 to movie1.

the SF movie factor and SF book factor, across different domains, SF movies can be recommended to users who like SF books. The mean and covariance values help to align the latent factors. The mean value represents popularity of the latent factor. The popularity of romance movies would be similar to the popularity of romance books. The covariance value represents correlation between two latent factors. The correlation between genres would be similar across different but related domains; users who like fantasy movies like SF movies, and also users who like fantasy books like SF books. The proposed method finds the correspondence between latent factors in different domains by using the mean and covariance information with shared Gaussian distributions.

The remainder of this paper is organized as follows. Section 2 reviews related work. In Section 3, we propose a cross-domain recommendation method based on matrix factorization with shared latent vector distributions, and present its inference procedures based on Gibbs sampling. Section 4 demonstrates the effectiveness of the proposed method with experiments on rating datasets in a wide variety of domains, e.g., movie, books and electronics. Finally, we present concluding remarks and a discussion of future work in Section 5.

2 Related work

Collaborative filtering is a technique to predict missing user preferences by using observed user-item preference data. Single-domain collaborative filtering methods, such as matrix factorization [18, 25] and nearest neighbor based methods [27], can be used for cross-domain recommendations by aggregating datasets from multiple domains into a single user-item rating matrix if there are shared users or items. However, the overlap of users and items across different domains is small and single-domain collaborative filtering methods are biased [6]. To alleviate the bias, a number of cross-domain collaborative filtering methods have been proposed [6, 14, 23]; these methods also assume shared users or items.

A limited number of cross-domain methods that do not require shared users and items have been proposed [9, 20, 30]. Multi-domain collaborative filtering [30] is based on matrix factorization, in which the relationships among user latent factors in different domains are modeled by a matrix-variate Gaussian distribution, and latent vectors are generated from Gaussian distributions with zero mean and diagonal covariance. Although this method is close to the proposed method, the proposed method can find relationships across domains by utilizing both the mean vectors and full covariance matrices for both the user and item latent vectors. A rating-matrix generative model [20] transfers information across domains by sharing implicit cluster-level rating patterns. A cluster-level latent factor model [9] assumes common latent and domain-specific rating patterns in matrix tri-factorization. These existing methods try to improve the performance of predicting ratings in a target domain by using data in other domains to alleviate the data sparsity problem in the target domain. On the other hand, our task is to predict ratings of user-item pairs in different domains. To our knowledge, this paper is the first study to demonstrate the prediction of ratings in different domains without shared users or items. In [6], it is noted that when there is no domain overlap, the only approach available is non-personalized collaborative filtering, e.g. recommending the most popular items, or predicting item ratings by the average ratings over users.

Content-based cross-domain recommendation methods have also been proposed [3, 22]. They require auxiliary information, such as user demographics and item content information, to calculate similarities between users/items in different domains. In contrast, the proposed method does not require any information other than the rating matrices from the domains.

The proposed method is based on Bayesian probabilistic matrix factorization [25], which has achieved high predictive performance for single-domain recommendation. In particular, the propose method applies, in effect, Bayesian probabilistic matrix factorization to a single large rating matrix constructed by combining ratings in all of the domains, where ratings among different domains are assumed to be missing values.

The proposed method is related to unsupervised object matching methods, which find correspondence between objects, or users and items, in different domains without alignment information [11, 24, 29, 17, 15]. The proposed method does not directly find the correspondence between user and item, but finds the correspondence between latent factors by sharing mean and covariance structures.

Latent semantic matching [13] learns latent vectors individually in each domain, and then uses unsupervised object matching methods to find the alignment of latent factors across different domains. Because of its two stage procedure, the learned latent vectors might not be suitable when aligned, and the accumulation of errors in the latent vector learning stage cannot be corrected in the factor alignment stage. On the other hand, since the proposed method simultaneously infers latent vectors and their factor alignments in a single framework, latent vectors are inferred so as to be optimal when their factors are aligned across multiple domains.

3 Proposed method

3.1 Model

We assume that we are given user-item rating matrices in D domains $\mathbf{R} = (\mathbf{R}_1, \cdots, \mathbf{R}_D)$, where $\mathbf{R}_d \in \mathbb{R}^{N_d \times M_d}$ is the rating matrix of domain d, N_d is the number of users in domain d, and M_d is the number of items in domain d. The (n, m) element of $\mathbf{R}_d$, r_{dnm}, represents the rating of user n to item m in domain d; $\mathbf{R}_d$ can contain missing values. The task is to predict ratings of user-item pairs from different domains.

We assume that each user has latent vector $\mathbf{u}_{dn} \in \mathbb{R}^{K\times 1}$, and each item has latent vector $\mathbf{v}_{dm} \in \mathbb{R}^{K\times 1}$, where K is the number of latent factors. Rating r_{dnm} is generated from a Gaussian distribution with mean $\mathbf{u}_{dn}^\top \mathbf{v}_{dm}$ and variance α^{-1}, $r_{dnm} \sim \mathcal{N}(\mathbf{u}_{dn}^\top \mathbf{v}_{dm}, \alpha^{-1})$. Inferring the latent vectors individually in each domain fails to predict cross-domain ratings because latent factors are randomly aligned if the overlap is nil. We try to align latent factors by assuming that user latent vectors in all domains are generated from a common full covariance Gaussian distribution, $\mathbf{u}_{dn} \sim \mathcal{N}(\boldsymbol{\mu}_\mathrm{u}, \boldsymbol{\Lambda}_\mathrm{u}^{-1})$, and item latent vectors in all domains are as well, $\mathbf{v}_{dm} \sim \mathcal{N}(\boldsymbol{\mu}_\mathrm{v}, \boldsymbol{\Lambda}_\mathrm{v}^{-1})$. This means that latent factors have the same mean vector and covariance matrix across different domains. By inferring the mean and covariance from given cross-domain rating matrices, the latent factors are aligned, which enables us to predict ratings in different domains. For the parameters of the shared Gaussian distributions, $(\boldsymbol{\mu}_\mathrm{u}, \boldsymbol{\Lambda}_\mathrm{u})$ and $(\boldsymbol{\mu}_\mathrm{v}, \boldsymbol{\Lambda}_\mathrm{v})$, conjugate Gaussian-Wishart priors are used.

In summary, the proposed method generates rating matrices from multiple domains $\mathbf{R}$ according to the following process,

1. Draw user latent vector precision matrix
   $\boldsymbol{\Lambda}_\mathrm{U} \sim \mathcal{W}(\mathbf{W}_0, \nu_0)$
2. Draw user latent vector mean
   $\boldsymbol{\mu}_\mathrm{U} \sim \mathcal{N}(\boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Lambda}_\mathrm{U})^{-1})$
3. Draw item latent vector precision matrix
   $\boldsymbol{\Lambda}_\mathrm{V} \sim \mathcal{W}(\mathbf{W}_0, \nu_0)$
4. Draw item latent vector mean
   $\boldsymbol{\mu}_\mathrm{V} \sim \mathcal{N}(\boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Lambda}_\mathrm{V})^{-1})$
5. For each domain $d = 1, \cdots, D$
   (a) For each user $n = 1, \cdots, N_d$
      i. Draw user latent vector
         $\mathbf{u}_{dn} \sim \mathcal{N}(\boldsymbol{\mu}_\mathrm{U}, \boldsymbol{\Lambda}_\mathrm{U}^{-1})$
   (b) For each item $m = 1, \cdots, M_d$
      i. Draw item latent vector
         $\mathbf{v}_{dn} \sim \mathcal{N}(\boldsymbol{\mu}_\mathrm{V}, \boldsymbol{\Lambda}_\mathrm{V}^{-1})$
   (c) For each user $n = 1, \cdots, N_d$
      i. For each item $m \in \mathbf{M}_{dn}$
         A. Draw rating
            $r_{dnm} \sim \mathcal{N}(\mathbf{u}_{dn}^\top \mathbf{v}_{dm}, \alpha^{-1})$.

Here, $\mathbf{M}_{dn}$ is a set of items rated by user n in domain d, and $\mathcal{W}(\boldsymbol{\Lambda}|\mathbf{W}_0, \nu_0)$ represents a Wishart distribution, where $\mathbf{W}$ is a $K \times K$ symmetric positive

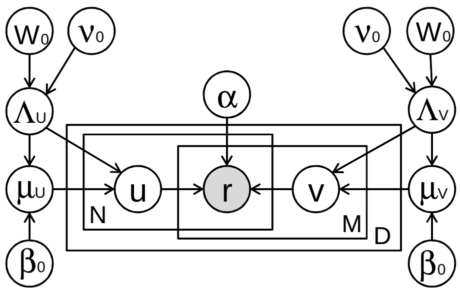


Figure 3: Graphic model representation of the proposed method.

definite matrix, and $\nu_0 > K - 1$ is the number of degrees of freedom. The mean of this distribution is $\nu\mathbf{W}$. Figure 3 shows the graphical model representation of the proposed method, where the shaded and unshaded nodes indicate observed and latent variables, respectively. The scales of ratings are assumed to be the same over all domains. If the scales are different, they can be unified by normalizing ratings for each domain in a preprocessing step.

3.2 Inference

The unknown parameters in the proposed method are user latent vectors $\mathbf{U} = ((\mathbf{u}_{dn})_{n=1}^{N_d})_{d=1}^{D}$, item latent vectors $\mathbf{V} = ((\mathbf{v}_{dm})_{m=1}^{M_d})_{d=1}^{D}$, and shared Gaussian parameters $(\boldsymbol{\mu}_\mathrm{u}, \boldsymbol{\Lambda}_\mathrm{u})$ and $(\boldsymbol{\mu}_\mathrm{v}, \boldsymbol{\Lambda}_\mathrm{v})$. They are inferred by Gibbs sampling, where each parameter is sampled from its conditional distribution given the current state of all the other parameters. Because the proposed method uses conjugate priors for the parameters, the conditional distributions can be analytically obtained as described in [25]. The conditional probability for a user latent vector is given by

$$p(\mathbf{u}_{dn}|\mathbf{R}, \mathbf{V}, \boldsymbol{\mu}_\mathrm{u}, \boldsymbol{\Lambda}_\mathrm{u}, \alpha) = \mathcal{N}(\boldsymbol{\mu}_{dn}, \boldsymbol{\Lambda}_{dn}), \tag{1}$$

where

$$\boldsymbol{\mu}_{dn} = \boldsymbol{\Lambda}_{dn}^{-1}\left(\alpha \sum_{m\in\mathbf{M}_{dn}} \mathbf{v}_{dm} r_{nm} + \boldsymbol{\Lambda}_\mathrm{u}\boldsymbol{\mu}_\mathrm{u}\right), \tag{2}$$

$$\boldsymbol{\Lambda}_{dn} = \boldsymbol{\Lambda}_\mathrm{u} + \alpha \sum_{m\in\mathbf{M}_{dn}} \mathbf{v}_m \mathbf{v}_m^\top, \tag{3}$$

are the mean and precision matrix of the posterior Gaussian distribution of user latent vector $\mathbf{u}_{dn}$, respectively. Similarly, the conditional probability for an item latent vector is given by

$$p(\mathbf{v}_{dn}|\mathbf{R}, \mathbf{U}, \boldsymbol{\mu}_\mathrm{v}, \boldsymbol{\Lambda}_\mathrm{v}, \alpha) = \mathcal{N}(\boldsymbol{\mu}_{dm}, \boldsymbol{\Lambda}_{dm}), \tag{4}$$

where

$$\boldsymbol{\mu}_{dm} = \boldsymbol{\Lambda}_{dm}^{-1}\left(\alpha \sum_{n\in\mathbf{N}_{dm}} \mathbf{u}_{dn} r_{nm} + \boldsymbol{\Lambda}_\mathrm{v}\boldsymbol{\mu}_v\right), \tag{5}$$

$$\boldsymbol{\Lambda}_{dm} = \boldsymbol{\Lambda}_\mathrm{v} + \alpha \sum_{n\in\mathbf{N}_{dm}} \mathbf{u}_n \mathbf{u}_n^\top, \tag{6}$$

are the mean and precision matrix of the posterior Gaussian distribution of item latent vector $\mathbf{v}_{dm}$, respectively. Here, $\mathbf{N}_{dm}$ is the set of users who give ratings to item m in domain d.

The shared mean and covariance matrix for user latent vectors are sampled from the following posterior Gaussian-Wishart distribution

$$\begin{aligned} &p(\boldsymbol{\mu}_\mathrm{u}, \boldsymbol{\Lambda}_\mathrm{u}|\mathbf{U}, \mathbf{W}_0, \nu_0, \beta_0, \boldsymbol{\mu}_0) \\ &\quad = \mathcal{N}(\boldsymbol{\mu}_\mathrm{u}|\boldsymbol{\mu}_{0\mathrm{u}}, (\beta_{0\mathrm{u}}\boldsymbol{\Lambda}_\mathrm{u})^{-1})\mathcal{W}(\boldsymbol{\Lambda}_\mathrm{u}|\mathbf{W}_{0\mathrm{u}}, \nu_{0\mathrm{u}}), \end{aligned} \tag{7}$$

where

$$N = \sum_{d=1}^{D} N_d, \quad \boldsymbol{\mu}_{0\mathrm{u}} = \frac{\beta_0\boldsymbol{\mu}_0 + N\bar{\mathbf{u}}}{\beta_0 + N}, \tag{8}$$

$$\beta_{0\mathrm{u}} = \beta_0 + N, \quad \nu_{0\mathrm{u}} = \nu_0 + N, \tag{9}$$

$$\mathbf{W}_{0\mathrm{u}}^{-1} = \mathbf{W}_0^{-1} + N\bar{\mathbf{S}}_\mathrm{u} + \frac{\beta_0 N}{\beta_0 + N}(\boldsymbol{\mu}_0 - \bar{\mathbf{u}})(\boldsymbol{\mu}_0 - \bar{\mathbf{u}})^\top, \tag{10}$$

$$\bar{\mathbf{u}} = \frac{1}{N}\sum_{d=1}^{D}\sum_{n=1}^{N_d} \mathbf{u}_{dn}, \tag{11}$$

$$\bar{\mathbf{S}}_\mathrm{u} = \frac{1}{N}\sum_{d=1}^{D}\sum_{n=1}^{N_d} (\mathbf{u}_{dn} - \bar{\mathbf{u}})(\mathbf{u}_{dn} - \bar{\mathbf{u}})^\top. \tag{12}$$

In the same way, the shared mean and covariance matrix for item latent vectors are sampled from the posterior distribution

$$\begin{aligned} &p(\boldsymbol{\mu}_\mathrm{v}, \boldsymbol{\Lambda}_\mathrm{v}|\mathbf{V}, \mathbf{W}_0, \nu_0, \beta_0, \boldsymbol{\mu}_0) \\ &\quad = \mathcal{N}(\boldsymbol{\mu}_\mathrm{v}|\boldsymbol{\mu}_{0\mathrm{v}}, (\beta_{0\mathrm{v}}\boldsymbol{\Lambda}_\mathrm{v})^{-1})\mathcal{W}(\boldsymbol{\Lambda}_\mathrm{v}|\mathbf{W}_{0\mathrm{v}}, \nu_{0\mathrm{v}}), \end{aligned} \tag{13}$$

where

$$M = \sum_{d=1}^{D} M_d, \quad \boldsymbol{\mu}_{0\mathrm{v}} = \frac{\beta_0\boldsymbol{\mu}_0 + M\bar{\mathbf{v}}}{\beta_0 + M}, \tag{14}$$

$$\beta_{0\mathrm{v}} = \beta_0 + M, \quad \nu_{0\mathrm{v}} = \nu_0 + M, \tag{15}$$

$$\mathbf{W}_{0\mathrm{v}}^{-1} = \mathbf{W}_0^{-1} + M\bar{\mathbf{S}}_\mathrm{v} + \frac{\beta_0 M}{\beta_0 + M}(\boldsymbol{\mu}_0 - \bar{\mathbf{v}})(\boldsymbol{\mu}_0 - \bar{\mathbf{v}})^\top, \tag{16}$$

$$\bar{\mathbf{v}} = \frac{1}{M}\sum_{d=1}^{D}\sum_{m=1}^{M_d}\mathbf{v}_{dm}, \tag{17}$$

$$\bar{\mathbf{S}}_\mathrm{v} = \frac{1}{M}\sum_{d=1}^{D}\sum_{m=1}^{M_d}(\mathbf{v}_{dm} - \bar{\mathbf{v}})(\mathbf{v}_{dm} - \bar{\mathbf{v}})^\top. \tag{18}$$

To initialize the latent vectors we use the result of the maximum a posteriori (MAP) estimation based on the probabilistic matrix factorization model [21]. Then, we iterate the sampling of shared means and covariances $\boldsymbol{\mu}_\mathrm{u}, \boldsymbol{\Lambda}_\mathrm{u}, \boldsymbol{\mu}_\mathrm{v}, \boldsymbol{\Lambda}_\mathrm{v}$ by (7) and (13), user latent vectors $\mathbf{U}$ by (1), and item latent vectors $\mathbf{V}$ by (4). The rating of user n in domain d to item m in domain d' is estimated by averaging over the samples after a burn-in period

$$\hat{r}_{dnd'm} = \frac{1}{S}\sum_{s=1}^{S} r^{(s)}_{dnd'm}, \tag{19}$$

where S is the number of samples, and $r^{(s)}_{dnd'm}$ is the sth sample that is sampled by the following Gaussian distribution

$$r_{dnd'm} \sim \mathcal{N}(\mathbf{u}_{dn}^\top \mathbf{v}_{d'm}, \alpha^{-1}). \tag{20}$$

The averaging means that ratings are predicted by the posterior distribution of the latent vectors in a Bayesian framework, but not by the point estimates. For all of the experiments, we used the following hyperparameters: $\mathbf{W}_0 = \mathbf{I}, \nu_0 = K, \beta_0 = 1, \boldsymbol{\mu}_0 = \mathbf{0}, \alpha = 2$.

4 Experiments

4.1 Datasets

We evaluated the proposed method on the following three movie rating datasets: Movielens [12], EachMovie and Netflix [2], and Amazon review rating datasets from the following six categories: Book, DVD, Electronics, Kitchen, Music and Video [5]. The numbers of users, items and ratings for each dataset are shown in Table 1.

For the movie datasets, we generated two rating matrices for each dataset with no overlap. In detail, users and items are randomly split into two groups of equal size, and the ratings between users and items assigned into the same group are used for training, and those assigned to different groups are used for testing, as in Figure 1.

Table 1: Statistics of datasets used in our experiments.

	#users	#items	#ratings
Movielens	943	1,682	100,000
Eachmovie	61,265	1,623	2,811,718
Netflix	480,189	17,770	100,480,507
Book	21,237	65,044	248,829
DVD	3,915	10,868	35,350
Elec.	372	2,148	797
Kitchen	317	1,563	727
Music	4,157	14,738	39,350
Video	1,816	3,339	12,070

For the Amazon datasets, we generated two rating matrices using each pair of categories with no overlap. For each category pair, users are randomly split into two groups, and the ratings of users in the first (second) group to items in the first (second) category are used for training, and the rest are used for testing.

4.2 Comparing the methods

We compared the proposed method with seven methods: CLFM, MF-MM, LSM, MF, Mean-U, Mean-I and Mean, which are explained below. CLFM (cluster-level latent factor model) [9] is a cross-domain recommendation method that assumes common latent and domain-specific rating patterns in matrix tri-factorization. With MF-MM (matrix factorization with mean matching), latent vectors for each domain are obtained individually by MAP estimation based on the probabilistic matrix factorization model [21]. Then, the latent factors in different domains are aligned by sorting them by their mean values. LSM (latent semantic matching) [13] finds alignment of the latent factors across different domains by convex kernelized sorting [7]. Convex kernelized sorting is a kernel based unsupervised object matching method that finds correspondence by maximizing the dependency measured in terms of Hilbert-Schmidt information criterion [10]. The MF (probabilistic matrix factorization) does not align latent factors. We included MF to show how well a single-domain method works for our task. The Mean-U (mean value for each user) method predicts ratings by the mean value for the user. It cannot be used for recommendation because it predicts the same rating for all the items for each user. The Mean-I (mean value for each item) method predicts ratings by the mean value for the item. It cannot personalize recommendations, and it suggests popular items. The Mean method predicts ratings by the mean value of all ratings. For all of the matrix factorization based methods (the proposed method, CLFM, MF-MM and LSM), the number of latent factors was set at $K = 30$.

4.3 Results

The task is to predict ratings in different domains. Table 2 shows the root mean squared error (RMSE) for test data, as averaged over 30 experiments. The proposed method achieved the lowest RMSE in 15 of the 18 datasets. CLFM achieved low error in the movie datasets because the statistical properties of two domains in each movie dataset are the same, which are generated from a single rating matrix. However, CLFM failed to predict ratings in different categories in the Amazon datasets, because different categories have different properties. On the other hand, the RMSE values of the proposed method were consistently low for all datasets examined, which indicates that the proposed method can well handle domains with different properties. The RMSE of MF-MM was also relatively low in the movie datasets, but high in the Amazon datasets. LSM yielded worse RMSE than MF-MM, which indicates that the mean values of latent factors are informative for aligning factors. The proposed method improves the predictive performance by using the mean values as well as the covariance information for aligning factors. MF had high RMSE for all datasets, which shows that single-domain methods fail at cross-domain rating prediction given the lack of shared users and items.

Table 3 shows the average test RMSE of the proposed method for each pair of categories. The difficulty of rating prediction largely depends on the item category. For example, the RMSEs for the Electronics items were generally higher than those for the Music items. The RMSE for the Video (DVD) users to the DVD (Video) items were lower than the average of the DVD (Video) items. This is reasonable because DVD and Video are closely related categories. The RMSEs for the Book items did not vary when user category was changed, and the RMSEs of the Book users were high for all item categories. This is because there were many users and items in the Book category, and latent factors were learned mainly for the Book category but not for the other category.

Figure 4 shows the average test RMSEs with different numbers of latent factors in the Movielens dataset. When there were more than a few latent factors, the proposed method achieved the best performance. The proposed method did not overfit even when there were many latent factors.

Figure 5 shows the test RMSEs over inference iteration with the Movielens dataset. The burn-in period was 1,000. In the burn-in period, the test RMSEs were calculated by a sample; after burn-in, they were calculated by averaging over samples. At the first iteration, the RMSE was very high, where the MAP estimate by the probabilistic matrix factorization was used. With a small number of iterations, the RMSE decreased drastically due to the use of shared mean and covariance values. After the burn-in period, the RMSE decreased again by averaging over multiple samples with the Bayesian prediction framework.

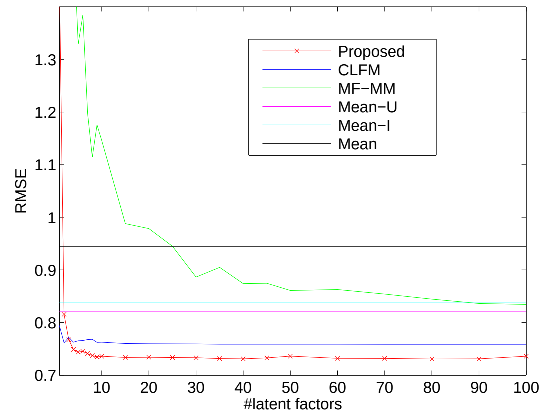


Figure 4: Average test RMSEs with different numbers of latent factors with the Movielens dataset.

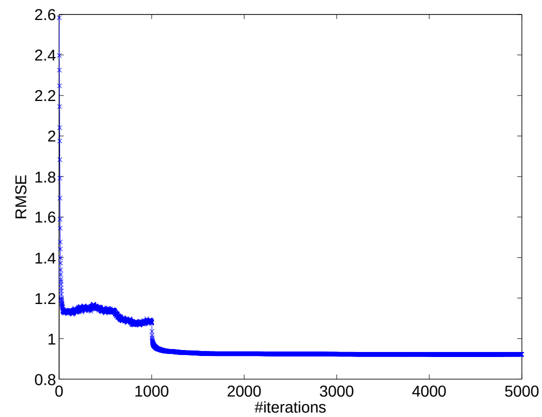


Figure 5: Test RMSEs over inference iteration number with the Movielens dataset. The burn-in period was 1,000.

Figure 6 shows the empirical mean vectors and covariance matrices of latent vectors for each domain estimated by (a) MF and (b) the proposed method. With MF, the latent factors are not aligned; the mean and covariance values of different domains are different. On the other hand, with the proposed method, the latent factors are aligned, which enables the latent vectors to be used in predicting cross-domain ratings.

5 Conclusion

We proposed a cross-domain recommendation method that can predict ratings in different domains even if

Table 2: Average test RMSE. Values in bold typeface are statistically better (at the 5% level) from those in normal typeface as indicated by a paired t-test.

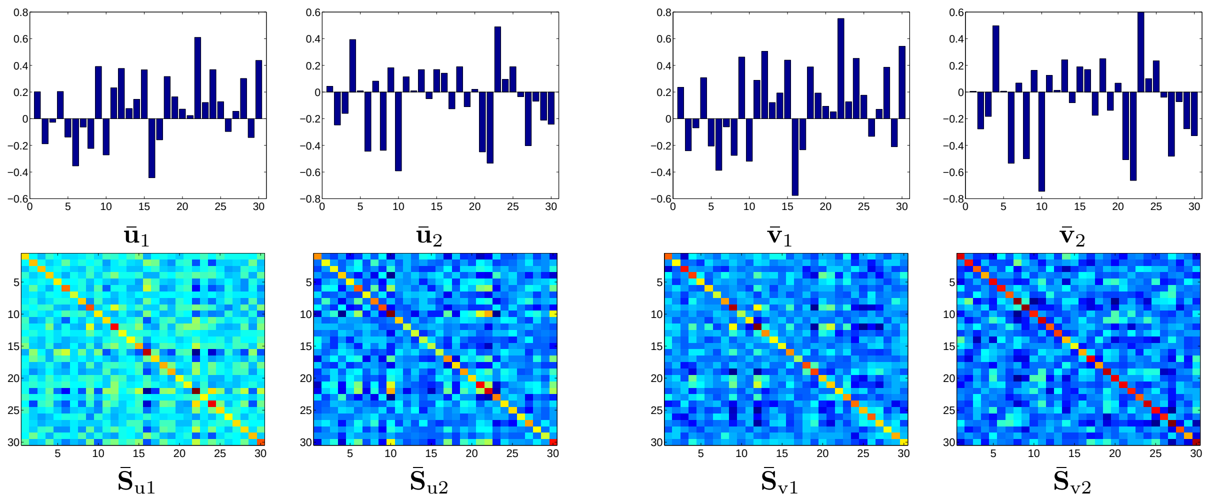
	Proposed	CLFM	MF-MM	LSM	MF	Mean-U	Mean-I	Mean
Movielens	**0.925**	0.963	1.090	1.948	2.696	1.063	1.093	1.266
Eachmovie	**0.961**	1.036	1.188	2.217	2.696	1.324	1.230	1.564
Netflix	1.043	**0.927**	1.086	2.081	2.819	1.000	1.022	1.176
Book-DVD	**1.178**	3.158	3.134	3.453	3.406	1.507	1.616	1.338
Book-Elec.	**1.180**	3.558	3.487	3.488	3.488	1.451	1.618	1.341
Book-Kitchen	**1.181**	3.557	3.485	3.487	3.487	1.442	1.617	1.343
Book-Music	**1.138**	3.284	3.225	3.476	3.507	1.388	1.544	1.278
Book-Video	**1.184**	3.385	3.332	3.471	3.477	1.479	1.629	1.353
DVD-Elec.	**1.135**	3.452	3.415	3.420	3.419	1.416	1.605	1.315
DVD-Kitchen	**1.128**	3.473	3.427	3.430	3.430	1.404	1.585	1.295
DVD-Music	**1.041**	2.945	2.951	3.412	3.514	1.328	1.344	1.110
DVD-Video	**1.053**	2.118	2.502	3.117	3.391	1.125	1.664	1.392
Elec.-Kitchen	**1.369**	3.600	3.436	3.437	3.441	2.024	1.930	1.787
Elec.-Music	0.953	4.465	3.596	3.599	3.597	1.056	1.139	**0.914**
Elec.-Video	**1.317**	3.470	3.326	3.329	3.328	1.827	2.061	1.764
Kitchen-Music	0.948	4.503	3.601	3.603	3.604	1.035	1.123	**0.900**
Kitchen-Video	**1.300**	3.490	3.340	3.346	3.347	1.796	2.080	1.711
Music-Video	**1.045**	3.273	3.245	3.527	3.540	1.333	1.334	1.106
average	**1.112**	2.940	2.937	3.214	3.344	1.389	1.513	1.331

Table 3: Average test RMSE of the proposed method for each pair of categories in Amazon datasets. For example, the value in the Book-row of the DVD-column shows the RMSE for the rating prediction of users in the Book category to items in the DVD category. Values in bold typeface indicate the lowest values in each item category (column).

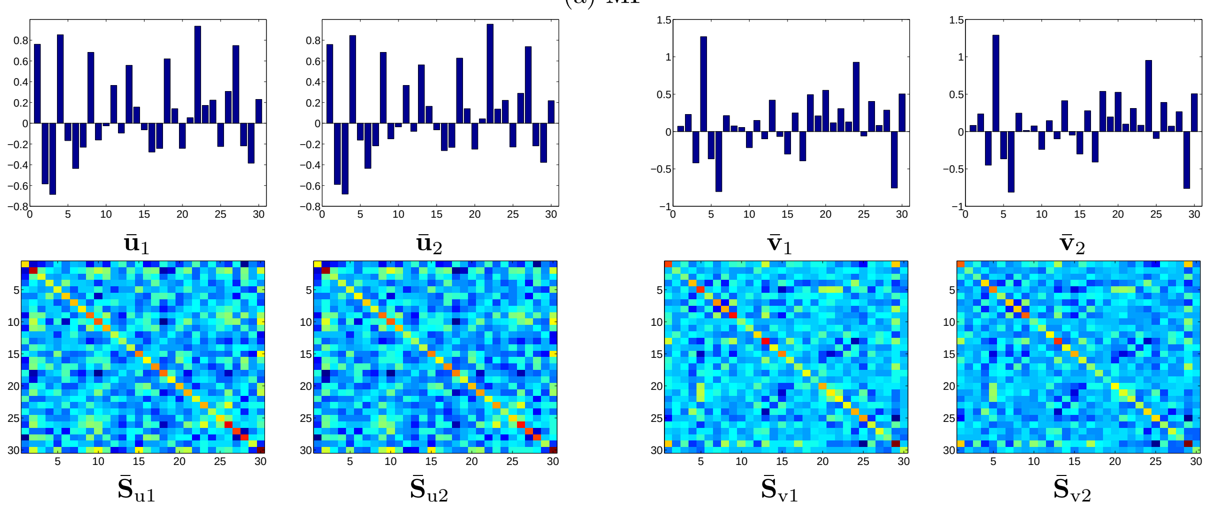
user \ item	Book	DVD	Elec.	Kitchen	Music	Video	average
Book	-	1.201	1.551	1.402	1.038	1.310	1.300
DVD	1.175	-	1.489	1.376	0.962	**1.094**	1.219
Elec.	1.178	1.126	-	1.295	**0.938**	1.312	1.170
Kitchen	1.180	1.121	1.437	-	**0.938**	1.300	1.195
Music	**1.155**	1.125	1.493	1.381	-	1.303	1.291
Video	1.177	**1.033**	**1.405**	**1.282**	0.955	-	1.170
average	1.173	1.121	1.475	1.347	0.966	1.264	1.047

neither shared users nor items are available. The proposed method aligns latent factors in different domains by assuming the latent factors have common mean and covariance structures that are shared by all domains. Experiments on real-world data sets confirmed that the proposed method achieved better predictive performance for ratings in different domains.

Although our results are encouraging, we must extend our approach in a number of directions. First, the interpretability of the inferred latent factors must be improved. The inferred latent factors in our experiments were not interpretable as genres or topics. It is because the proposed method allows latent factors to have arbitrary sign, and involves complex cancellations between positive and negative latent factors [19]. The interpretability might be improved by using non-negativity constraints. Second, we would like to extend the proposed method to handle domain heterogeneity. The proposed method described here assumes that all domains share a common distribution for generating latent vectors. However, some latent factors are used only for a certain domain, and relationships between domains would be different. By introducing domain-specific latent factors as well as shared, domain heterogeneity could be modeled as introduced in canonical correlation analysis [1, 16, 26, 28]. Third, the proposed framework can be applied to different types of datasets by using other distributions for the observation. The proposed method assumes Gaussian observation noise for rating matrices. In the case of bag-of-words data, multiple topic models with shared mean and covariance structures can be an option, where a correlated topic model [4] is used as a component. Finally, the performance of the proposed method can be



(a) MF



(b) Proposed

Figure 6: Empirical mean vectors and covariance matrices of latent vectors for each domain estimated by (a) single-domain matrix factorization and (b) the proposed cross-domain method. The empirical mean vector of user latent vectors for domain d is calculated by $\bar{\mathbf{u}}_d = \frac{1}{N_d}\sum_{n=1}^{N_d}\mathbf{u}_{dn}$, and the covariance matrix for domain d is calculated by $\bar{\mathbf{S}}_{\text{u}d} = \frac{1}{N_d}\sum_{n=1}^{N_d}(\mathbf{u}_{dn} - \bar{\mathbf{u}}_d)(\mathbf{u}_{dn} - \bar{\mathbf{u}}_d)^\top$. The empirical mean vector of item latent vectors for domain d is calculated by $\bar{\mathbf{v}}_d = \frac{1}{M_d}\sum_{m=1}^{M_d}\mathbf{v}_{dm}$, and the covariance matrix for domain d is calculated by $\bar{\mathbf{S}}_{\text{v}d} = \frac{1}{M_d}\sum_{m=1}^{M_d}(\mathbf{v}_{dm} - \bar{\mathbf{v}}_d)(\mathbf{v}_{dm} - \bar{\mathbf{v}}_d)^\top$. With regard to the bar graph for the mean, the horizontal axis represents latent factor indices, and the vertical axis represents their mean values. For covariance, the horizontal and vertical axes represent latent factor indices, and the color shows their covariance values. They were calculated by the last sample in the inference with the proposed method.

improved in a semi-supervised setting by using a small number of shared users or items, and auxiliary information, such as user demographics and item content information.

References

[1] C. Archambeau and F. R. Bach. Sparse probabilistic projections. In *Advances in Neural Information Processing Systems*, pages 17–24, 2009.

[2] J. Bennett and S. Lanning. The N etflix prize. In *Proceedings of KDD cup and workshop*, 2007.

[3] S. Berkovsky, T. Kuflik, and F. Ricci. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3):245–286, 2008.

[4] D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems*, pages 147–154, 2006.

[5] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, pages 440–447, 2007.

[6] P. Cremonesi, A. Tripodi, and R. Turrin. Cross-domain recommender systems. In *IEEE International Conference on Data Mining Workshops*, pages 496–503, 2011.

[7] N. Djuric, M. Grbovic, and S. Vucetic. Convex kernelized sorting. In *AAAI*, 2012.

[8] I. Fernández-Tobías, I. Cantador, M. Kaminskas, and F. Ricci. Cross-domain recommender systems: A survey of the state of the art. In *Spanish Conference on Information Retrieval*, 2012.

[9] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari, and J. Guo. Cross-domain recommendation via cluster-level latent factor model. In *Machine Learning and Knowledge Discovery in Databases*, pages 161–176. Springer, 2013.

[10] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic learning theory*, pages 63–77. Springer, 2005.

[11] A. Haghighi, P. Liang, T. Berg-Kirkpatrick, and D. Klein. Learning bilingual lexicons from monolingual corpora. In *ACL:HLT*, pages 771–779, 2008.

[12] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*, pages 230–237, 1999.

[13] T. Hirao, T. Iwata, and M. Nagata. Latent semantic matching: Application to cross-language text categorization without alignment information. In *ACL*, pages 212–216, 2013.

[14] L. Hu, J. Cao, G. Xu, J. Wang, Z. Gu, and L. Cao. Cross-domain collaborative filtering via bilinear multilevel analysis. In *IJCAI*, pages 2626–2632, 2013.

[15] T. Iwata, T. Hirao, and N. Ueda. Unsupervised cluster matching via probabilistic latent variable models. In *AAAI*, pages 445–451, 2013.

[16] Y. Jia, M. Salzmann, and T. Darrell. Factorized latent spaces with structured sparsity. *Advances in Neural Information Processing Systems*, pages 982–990, 2010.

[17] A. Klami. Variational Bayesian matching. In *ACML*, pages 205–220, 2012.

[18] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[19] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[20] B. Li, Q. Yang, and X. Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *ICML*, pages 617–624, 2009.

[21] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2007.

[22] M. Nakatsuji, Y. Fujiwara, A. Tanaka, T. Uchiyama, and T. Ishida. Recommendations over domain specific user graphs. In *ECAI*, pages 607–612, 2010.

[23] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI*, volume 10, pages 230–235, 2010.

[24] N. Quadrianto, A. J. Smola, L. Song, and T. Tuytelaars. Kernelized sorting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1809–1821, 2010.

[25] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *ICML*, pages 880–887. ACM, 2008.

[26] M. Salzmann, C. H. Ek, R. Urtasun, and T. Darrell. Factorized orthogonal latent spaces. In *AISTATS*, pages 701–708, 2010.

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

[28] S. Virtanen, A. Klami, and S. Kaski. Bayesian CCA via group sparsity. In *ICML*, pages 457–464, 2011.

[29] M. Yamada and M. Sugiyama. Cross-domain object matching with model selection. In *AISTATS*, pages 807–815, 2011.

[30] Y. Zhang, B. Cao, and D.-Y. Yeung. Multi-domain collaborative filtering. In *UAI*, pages 725–732, 2010.

[31] Y. C. Zhang, D. Ó. Séaghdha, D. Quercia, and T. Jambor. Auralist: introducing serendipity into music recommendation. In *WSDM*, pages 13–22, 2012.