Improving Top-N Recommendation for Cold-Start Users via Cross-Domain Information

NIMA MIRBAKHSH and CHARLES X. LING, Western University

Making accurate recommendations for cold-start users is a challenging yet important problem in recommendation systems. Including more information from other domains is a natural solution to improve the recommendations. However, most previous work in cross-domain recommendations has focused on improving prediction accuracy with several severe limitations. In this article, we extend our previous work on clustering-based matrix factorization in single domains into cross domains. In addition, we utilize recent results on unobserved ratings. Our new method can more effectively utilize data from auxiliary domains to achieve better recommendations, especially for cold-start users. For example, our method improves the recall to 21% on average for cold-start users, whereas previous methods result in only 15% recall in the cross-domain Amazon dataset. We also observe almost the same improvements in the Epinions dataset. Considering that it is often difficult to make even a small improvement in recommendations, for cold- start users in particular, our result is quite significant.

Categories and Subject Descriptors: H.3 [**Information Storage and Retrieval**]: Retrieval Models, Information Filtering, Clustering

General Terms: Collaborative Filtering, Recommendation System

Additional Key Words and Phrases: Collaborative filtering, recommendation system, matrix factorization, cold start

ACM Reference Format:
Nima Mirbakhsh and Charles X. Ling. 2015. Improving top-N recommendation for cold-start users via cross-domain information. ACM Trans. Knowl. Discov. Data 9, 4, Article 33 (June 2015), 19 pages.
DOI: http://dx.doi.org/10.1145/2724720

1. INTRODUCTION

Collaborative filtering–based recommendation systems employ observed ratings to generate a list of relevant items for each user. These systems explore the interests of similar users for modeling user preferences. Netflix, Amazon, and YouTube are examples of big companies that have successfully integrated collaborative filtering into their recommendation engines. However, the performance of these models is highly sensitive to the number of observed ratings that they are employing. Thus, these models do not achieve good recommendations for new users who have made no or very few observed ratings. Finding accurate recommendations for these low-profile users, referred to as cold-start users, is a challenging yet important problem in recommendation systems [Rashid et al. 2008; Koren et al. 2009; Schein et al. 2002].

Including more information from auxiliary domains is a natural solution to improve personalized recommendations in the target domain. These auxiliary domains can be categorized according to their users and items overlap, full overlap, users overlap, items

Authors' addresses: N. Mirbakhsh and C. X. Ling, Department of Computer Science, Western University, London, ON, Canada; emails: nmirbakhsh@gmail.com, cling@csd.uwo.ca (C. X. Ling is the contact author).

DOI: http://dx.doi.org/10.1145/2724720

overlap, or no overlap [Cremonesi et al. 2010]. Many algorithms have been proposed in cross-domain recommendation systems, which successfully address this problem by considering any or all of these overlapping scenarios [Cremonesi et al. 2010; Tang et al. 2012; Li et al. 2009; Li 2011]. However, most of these proposed models focus on improving rating prediction accuracy, often observed in terms of the root mean square error (RMSE). However, the major role of a recommendation system is to make a short list of 10 or 20 relevant items for each user. Hence, rating prediction for all nonobserved ratings is definitely a less important task than accurately ranking the top relevant items as a short list of recommendations, often measured by top-N recommendation task. Thus, optimizing the recommendation model regarding RMSE is highly criticized [Koren 2008; Steck 2013, 2010; Cremonesi et al. 2011], as they do not necessarily improve the top-N recommendation task.

In addition, it is empirically proved that because of unbalanced distribution of observed ratings, employing unobserved ratings improves top-N recommendation tasks dramatically [Steck 2013, 2010; Cremonesi et al. 2011]. For instance, Steck [2010] shows that employing unobserved ratings in matrix factorization achieves 64% of recall in the well-known Netflix dataset. However, both matrix factorization excluding unobserved ratings and the well-known integrated model proposed in Koren [2008] result in 39% and 43% recall, respectively. In this article, we take advantage of unobserved ratings on two levels: the traditional user-item level and our proposed cluster level of users and items in a latent space [Mirbakhsh and Ling 2013].

In our previous works [Mirbakhsh and Ling 2013, 2014], we define cluster-level "coarse" matrices in *single domains*. These coarse matrices capture the shared interests among the cluster of users and the cluster of items. Thus, it generalizes the preferences of users on items (into a cluster-level rating matrix) to reduce the sparsity of the original rating matrix. Here, we extend our previous work from single-domain into cross-domain recommendation systems by employing the partially overlapped users and items between multiple domains to transfer their cluster-level preferences as the auxiliary sources of information.

In sum, this article makes several novel contributions:

—We utilize the information of unobserved ratings in cross-domain recommendation systems via a cluster-level space.
—Cross-domain methods mostly need heavy computations to find a transferring function between each pair of domains [Cremonesi et al. 2010; Tang et al. 2012; Li et al. 2009], whereas our proposed method simply transfers the knowledge of several auxiliary domains between each other in one step (Section 2.2).
—We practically show that integrating the transferred cluster-level and the traditional transferred user-item level knowledge can significantly improve the recommendation quality.

To validate our new method, we set up our empirical experiment on two datasets: the Amazon dataset and the Epinions dataset. Both datasets include multiple domains, such as DVD, video, and electronics. Our experiments show that our proposed clustering-based matrix factorization model significantly increases the recall in top-N recommendation for all users, particularly cold-start users. For example, our method achieves a recall of 43% on average for all users compared to 39% using the previous methods in the Amazon dataset. We also observe 25% improvements of top-N recommendation in the Epinions dataset. For cold-start users, our method improves recall to 21% on average, whereas previous methods result in only 15% recall by including data from other domains (see Section 3 for more details) in the Amazon dataset. We observed almost the same improvements in the Epinions dataset as well. Note that it is often difficult to make even a small improvement in recommendations, especially for cold-start users. For instance, the difference between the biased matrix factorization and the well-observed integrated model [Koren 2008] was only 3% of recall for top-N recommendation tasks in the Netflix dataset. Hence, our improved rate of recall is quite significant.

2. PROPOSED METHOD

In a general collaborative filtering–based recommendation system, we have a set of users $U = \{u_1, u_2, \ldots, u_n\}$ and a set of items $I = \{i_1, i_2, \ldots, i_m\}$ that are accompanied by a rating matrix $R = [r_{ui}]_{n\times m}$, where r_{ui} represents the rating of user u on item i. A collaborative filtering–based recommendation system consists of making a short list of relevant items to user u based on the ratings inside rating matrix R. While the quality of these recommendations is sensitive to the number of observed ratings by user u. Hence, collaborative filtering–based recommendation systems are unable to generate accurate recommendations for users who have made no or very few observed ratings, known as cold-start users. Consequently, more information is needed to achieve a better quality of recommendations for cold-start users.

A major source of extra information is the ratings that these users have made in the other domains. For example assume an e-store Web site with different departments, including "books," "movies," "computers," and so forth. User u may have many observed ratings in the books domain but no ratings in the movies domain. Thus, a natural solution is using observed ratings from the books domain to generate a better list of recommendations in the movies domain for user u. This solution is referred to as cross-domain recommendations for a set of domains $D = \{d_1, d_2, \ldots, d_t\}$. These auxiliary domains can be categorized according to their users and items overlap, from full-overlap, users-overlap, and items-overlap domains to no-overlap domains [Cremonesi et al. 2010]. Our proposed cross-domain recommendation model is based on a partial overlap of users and/or items between the target domains and the auxiliary domains.

In our previous works [Mirbakhsh and Ling 2013, 2014], we define cluster-level coarse matrices in single domains (Section 2.1). These coarse matrices generalize the relation of users and items into a cluster level by capturing the mean rating between the cluster of users and the cluster of items. In Section 2.2, we extend our previous work to cross-domain recommendation systems. In a cross-domain scenario, we find the relations among these clusters in different domains. Thus, we can predict the preferences of the users in a target domain by employing their cluster-level preferences in the auxiliary domains. In Section 2.3 we define our final ranking function by aggregating the results of these two recommendation levels. Finally, in Section 2.4, we quickly review the factorization technique that we use for factorizing the original rating matrix and our novel cross-domain coarse matrix.

2.1. Making Single-Domain Coarse Matrices

The clustering-based matrix factorization in single domains [Mirbakhsh and Ling 2013, 2014] is briefly reviewed here. This method applies clustering algorithms to the *latent* vectors of users and items. It captures the shared interests and similarities among users and the items in a cluster-level rating matrix—coarse matrix. Clustering-based matrix factorization uses the mean ratings of the users inside a cluster of users on the items inside a cluster of items to capture the relations among these clusters in a coarse matrix. Factorizing this coarse matrix achieves cluster-level rating predictions.

More specifically, to generate a single-domain coarse matrix R^c, we apply the following two steps on each domain $d \in D$ to cluster users and items into domain-specific clusters:

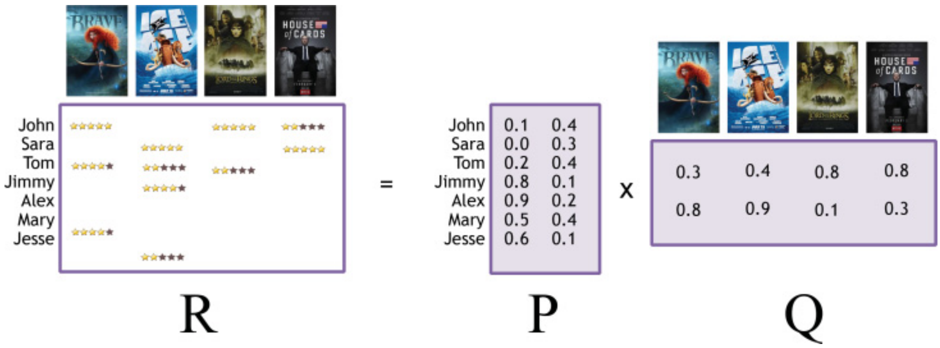

Fig. 1. Factorizing single-domain rating matrix R for each domain into lower dimension matrices P and Q.

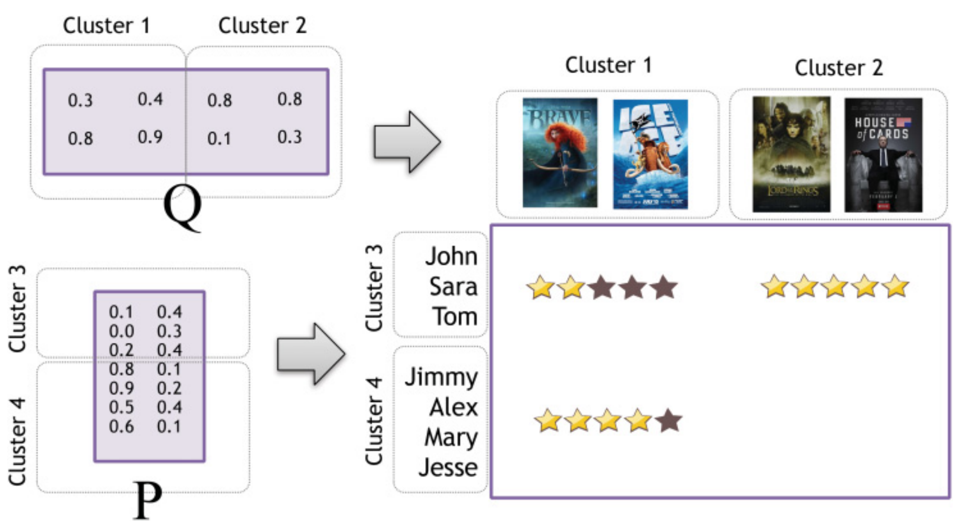

Fig. 2. Clustering latent matrices P and Q to achieve clusters of users and items and producing the coarse matrix. The coarse matrix generalizes preferences of users into a cluster level that leads to less sparsity in R^c.

(1) *Transforming:* We employ the biased matrix factorization that is proposed in Koren et al. [2009] to map rating matrix R into lower-dimension latent spaces $P \in \mathbb{R}^{n \times l}$ and $Q \in \mathbb{R}^{m \times l}$, where $R = PQ^T$ (Figure 1).
(2) *Clustering:* We then apply the K-means clustering algorithm on user latent space P and item latent space Q to categorize users and items into different clusters (Figure 2). In the transforming step, we reduce both the sparsity and the size of matrix R, which makes the clustering step computationally efficient.

Let us assume that m' is the number of clusters for items and n' is the number clusters for users. These selected values should not be too large or too small [Mirbakhsh and Ling 2014]. We guess different selections of possible categories by changing the number of clusters. Finally, by trying different sizes of clusters on the proposed models, the number of clusters that achieves more accurate recommendations on the validation set will be selected as the best choice of m' and n'.

We show in Mirbakhsh and Ling [2013, 2014] that aggregating the user-item and cluster-level rating predictions improves the quality of recommendations in single domains. For instance, our proposed clustering-based matrix factorization outperforms

Fig. 3. (Left) Cross-domain rating matrix R including rating matrices of the music and movies domains with overlapped users (dashed area). The rating matrix is very sparse, as many entries in the top right and lower left are missing values. (Right) Coarse matrix R^c including mean ratings between the cluster of users and the cluster of items. As shown, the coarse matrix reduces the sparsity of R by propagating the observed ratings into unobserved ratings. Note that the white area (missing values) is much reduced.

well-known collaborative filtering–based methods such as SVD++ [Koren 2008] and Asymmetric SVD [Koren 2008] in top-N recommendation tasks.

2.2. Making a Cross-Domain Coarse Matrix

As discussed earlier, cross-domain auxiliary information can be employed in the recommendation model to increase the recommendation accuracy further. In this work, we build a cross-domain cluster-level coarse matrix R^c, which captures the shared interests among the cluster of users and the cluster of items between multiple domains. Figure 3 illustrates rating matrix R including ratings from the music and movies domains. As shown, it is assumed that these two domains have partially overlapped users (dashed area). Let us assume that user u is a shared user between these domains. User u may rate item i in the music domain and item i' in the movies domain. In a classic matrix factorization, only these shared ratings will be transferred between the two domains. However, all entries in the top right and lower left (the white areas in the left-hand side of Figure 3) are missing values, and thus the rating matrix is too sparse.

In our model, we propagate these shared ratings into a cluster level (right-hand side of Figure 3). Hence, we reduce the sparsity of rating matrix R by propagating the observed ratings into unobserved ratings in coarse matrix R^c. In Figure 3, for instance, we may propagate the individual interest of users inside cluster 4 on a couple of cartoons into a cluster-level interest from cluster 4 on the cluster of cartoons as new entities. Note that the white area (missing values) is much reduced in Figure 3 (right). As overlapped users are separately clustered in each domain, they belong to more than one cluster in a cross-domain scenario. By factorizing this new matrix, the coarse matrix, we will have cluster-level preference prediction for each cluster of users.

Here is a formal description of our proposed method. Let us assume that $C^{d_j}_{U,u}$ is the uth cluster of users in domain d_j and $C^{d_j}_{I,i}$ is the ith cluster of items in domain d_j. After finding the domain-specific clusters (Section 2.1), we define the cross-domain coarse matrix R^c as follows:

$$R^c = \begin{bmatrix} R^c_{d_1,d_1} & R^c_{d_1,d_2} & \cdots \\ R^c_{d_2,d_1} & R^c_{d_2,d_2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \tag{1}$$

where

$$R^c_{d_j,d_{j'}} = \begin{bmatrix} r_{C^{d_j}_{U,1} C^{d_{j'}}_{I,1}} & r_{C^{d_j}_{U,1} C^{d_{j'}}_{I,2}} & \cdots \\ r_{C^{d_j}_{U,2} C^{d_{j'}}_{I,1}} & r_{C^{d_j}_{U,2} C^{d_{j'}}_{I,2}} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \tag{2}$$

and each $r_{C^{d_j}_{U,u} C^{d_{j'}}_{I,i}}$ is defined as follows:

$$r_{C^{d_j}_{U,u} C^{d_{j'}}_{I,i}} = \begin{cases} \frac{\sum_{u',i'} r_{u'i'}}{N(C^{d_j}_{U,u}, C^{d_{j'}}_{I,i})} & N\big(C^{d_j}_{U,u}, C^{d_{j'}}_{I,i}\big) > thr \\ \text{unobserved} & N\big(C^{d_j}_{U,u}, C^{d_{j'}}_{I,i}\big) \le thr \end{cases}. \tag{3}$$

Hence, $i' \in C^{d_{j'}}_{I,i}$ and $u' \in C^{d_j}_{U,u}$; $N(C^{d_j}_{U,u}, C^{d_{j'}}_{I,i})$ is the number of observed ratings that the users inside cluster $C^{d_j}_{U,u}$ have made on the items inside cluster $C^{d_{j'}}_{I,i}$. Let us remember that each overlapped user or item belongs to more than one cluster because each are clustered separately inside two or more different domains. For example, assume that user u belongs to $C^{d1}_{U,u}$ in domain $d1$ and also belongs to cluster $C^{d2}_{U,u}$ in domain $d2$. In this article, we suppose that $r_{C^{d1}_{U,u} C^{d}_{I,i}} = r_{C^{d2}_{U,u} C^{d}_{I,i}}$ (Figure 3). As is shown in Figure 3, this assumption propagates observed ratings into unobserved ratings. Thus, cluster-level rating matrix R^c (right-hand side of Figure 3) reduces the sparsity of R (left-hand side of Figure 3). However, sometimes there is not enough evidence to support this propagation. Hence, we employ a fixed threshold value thr to remove low-confidence relations between the clusters (we take $thr = 5$ in our experiment).

2.3. Generating Recommendations

In Section 2.4, we will show how to factorize matrices R, and R^c to rank the unobserved ratings. Here, we will describe the way that we make the cluster-level recommendations and how we aggregate them with traditional user-item–level recommendations. Let us define ${N_u}^C$ as the number of clusters (in different domains) to which user u belongs, ${N_i}^C$ as the number of clusters (in different domains) to which item i belongs, predicted rating matrix $\hat{R} = [\hat{r}_{ui}]$, and its cluster-level predicted rating matrix $\hat{R}^c = [\hat{r}^c_{ui}]$, where $\hat{r}_{ui} = r_m + P_u Q_i^T$, and

$$\hat{r}^c_{ui} = r_m + \frac{\sum_{d \in D} P^c_{C^d_{U,u}} Q^c_{C^d_{I,i}}}{N^C_u . N^C_i}. \tag{4}$$

The cluster-level predicted ratings in $\hat{R}^c$ are too general to be used solely. Hence, we integrate these two matrices linearly to achieve our final predictions as follows:

$$R^* = \alpha \hat{R}^c + (1 - \alpha)\hat{R}, \tag{5}$$

where $\alpha \in [0, 1]$ is a fixed tuning parameter and optimized via cross-validation. Thus, we employ matrix $R^* = [r^*_{ui}]$ to rank relevant items to each user.

For evaluating these recommendations, we use the top-N recommendation metric that is proposed by Koren [2008]. Assume T as the set of all ratings in the test set that $r_{ui} = 5$. For each test example $< u, i >$ in T, we randomly select 1,000 items from the set of items. We then predict the preference of user u on those selected items plus item i. We form a ranked list by ordering all 1,001 items according to their predicted ranks value in matrix R^*. Finally, we form a top-N recommendation list by picking the N top-ranked items from the list. If we have item i in the top-N list, we have a hit (the test item i is recommended to the user). Otherwise, we have a miss. Chances of a hit obviously increase with N. We measure the recall based on the number of hits in a list of N recommendations as follows:

$$Recall(N) = \frac{\#hits}{|T|}.$$

Thus, by increasing the recall, we will have more interesting items for each user in her personalized top-N list.

2.4. Factorizing Matrices Considering Unobserved Ratings

Usual collaborative filtering–based recommendation models are based on observed ratings. However, Steck [2010] shows that the distribution of usual datasets in recommendation systems are unbalanced and that any unobserved ratings can be considered as low-confidence nonpreference feedback from users. Thus, those observed-ratings–based algorithms ignore much useful feedback from users. Steck shows that by considering these unobserved ratings as a low rating value, r_m (e.g., $r_m = 2$ or any other value lower than the mean of observed ratings), the accuracy of recommendations increases dramatically. For example, he empirically shows that highly complex methods, such as the proposed integrated method [Koren 2008], achieve a recall of 42% based on top-N recommendation ($N = 20$) in the well-known Netflix dataset, but his unobserved ratings–integrated model increases this number to 64%.

However, factorizing a full-filled rating matrix (filled by replacing all unobserved ratings with r_m) with many numbers of users and items will be computationally expensive. Hence, he proposed a new alternative least squares (ALS)-based learning model to factorize this full-filled rating matrix into matrices P and Q more efficiently. He defines the following objective function to learn these factorized matrices:

$$\sum_{\text{all}\,u}\sum_{\text{all}i} W_{ui}.\left\{(R_{ui} - \hat{R}_{ui})^2 + \lambda.\left(\sum_{j=1:l} P_{i,j}^2 + Q_{u,j}^2\right)\right\}, \tag{6}$$

where $\hat{R} = R_m + P.Q^T$ is the predicted rating, and W_m is defined as follows:

$$W_{ui} = \begin{cases} w_{obs} & \text{if } R_{ui} \text{ is observed} \\ w_m & otherwise. \end{cases} \tag{7}$$

W_{ui} and λ are fixed parameters and will be optimized via a cross-validation task to maximize recommendation accuracy. ALS is then applied to find a (close to) minimum solution of Equation (6) by employing gradient descent. At each step, one of two matrices P and Q is assumed fixed, which turns the updating process of the other matrix into a quadratic optimization problem that can be solved exactly through equating the gradient of Equation (6) to zero. This results in the following updating equation for each corresponded latent vector Q_i (P is assumed fixed):

$$Q_i = (R_i - r_m)\hat{W}^{(i)}P\left(P^T\hat{W}^{(i)}P + \lambda.tr\left(\hat{W}^{(i)}\right)I\right)^{-1}, \tag{8}$$

where vector R_i includes all ratings on item i from rating matrix R, $\hat{W}^{(i)} \in \mathbb{R}^{n\times n}$ is the diagonal matrix containing the ith row of matrix W, and $I \in \mathbb{R}^{l\times l}$ is the identity matrix.

In the next step, Q will be assumed to be a fixed value, which turns the updating equation for each P_u as follows:

$$P_u = (R_u - r_m)\hat{W}^{(u)}Q\big(Q^T\hat{W}^{(u)}Q + \lambda.tr\big(\hat{W}^{(u)}\big)I\big)^{-1}, \tag{9}$$

where $\hat{W}^{(u)} \in \mathbb{R}^{m\times m}$ is the diagonal matrix containing the uth row of matrix W. Note that for unobserved ratings, we have $R_{i,u} - r_m = 0$. Thus, updating Equations (8) and (9) can be rewritten simpler and more computationally efficient, which is thoroughly discussed in Steck [2010]. Parallelization ability and simple updating process for new-coming users are two advantages of this updating mechanism.

We extend Steck's approach by applying this factorizing technique in two levels:

—To factorize cross-domain user-item–level rating matrix R.
—To factorize cross-domain cluster-level rating matrix R^c (Equation (1)).

Thus, to incorporate unobserved ratings into our method, we employ similar learning steps as shown in Equations (8) and (9) to factorize R^c into matrices $P^c \in \mathbb{R}^{n'_D\times l}$ and $Q^c \in \mathbb{R}^{m'_D\times l}$. We change Equation (6) regarding factorization of R^c as follows:

$$\sum_{\text{all } C_U^d}\sum_{\text{all } C_I^{d'}} W^c_{C_U^d C_I^{d'}}.\left\{\left(R^c_{C_U^d C_I^{d'}} - \hat{R}^c_{C_U^d C_I^{d'}}\right)^2 + \lambda^c.\left(\sum_{j=1:l} {P^c_{C^{d'}_{I,j}}}^2 + {Q^c_{C^d_{U,j}}}^2\right)\right\}, \tag{10}$$

where m'_D and n'_D are the total number of clusters for items and users, respectively; $R^c_{C_U^d C_I^{d'}}$ includes observed and nonobserved ratings using Equation (3); $\hat{R}^c = R_m + P^c {Q^c}^T$ is the cluster-level predicted rating; and

$$W^c_{C_U^d C_I^{d'}} = \begin{cases} w^c_{obs} & \text{if } R^c_{C_u C_i} \text{ is observed by Equation (3)} \\ w^c_m & otherwise. \end{cases} \tag{11}$$

We execute Equations (8) and (9) step by step for Equation (10) to learn latent matrices P^c and Q^c. Note that we use the same r_m but different λ and w_m in the learning process of Equation (10) because of the different rate of sparsity and the much lower dimensionality of R^c.

3. EXPERIMENTS

We employ two cross-domain datasets in our experiments: the Amazon dataset [Leskovec et al. 2007] and the Epinions dataset [Meyffret et al. 2012; Massa and Avesani 2006; Jamali and Ester 2010].

The Amazon dataset was collected from June 2001 to May 2003. In total, 548,523 products were recommended, where 68% of them belong to the books domain Thus, we ignore this domain to save our computations and also have more balanced distribution of observed ratings among domains. Figure 4 illustrates the number of observed ratings in the remaining domains. We select the top six domains with the largest numbers of observed ratings to employ in our cross-domain experiment. These six domains include DVD, music, video, electronics, kitchen and housewares, and toys and games. We refer to kitchen and housewares as "kitchen" and toys and games as "toys" for simplicity in the rest of this article. Table I presents the number of users, items, and observed ratings in each of these selected domains. As shown, the training sets are very sparse, and less than 1% of ratings are observed.

The Epinions datset [Meyffret et al. 2012] is extracted from Epinions[1] in June 2011. It contains reviews from users on items, trust values between users, item categories,

[1]http://www.epinions.com.

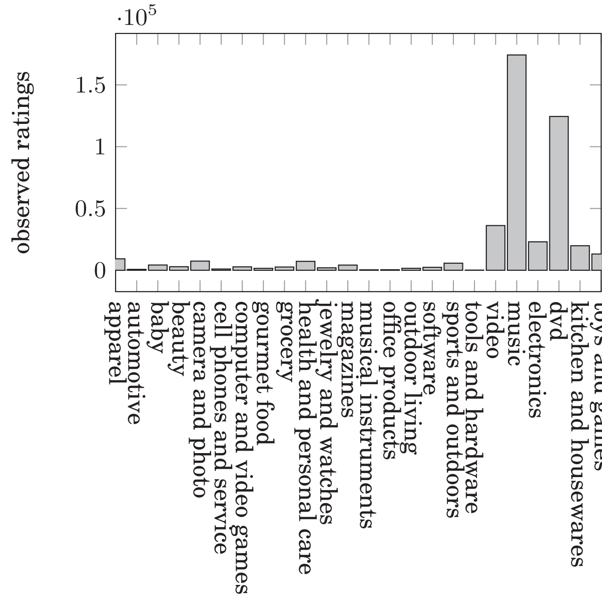

Fig. 4. Number of observed ratings in different domains in the Amazon dataset.

Table I. Number of Users, Items, and Observed Ratings in Six Selected Domains in the Amazon Dataset

Domains	Users (#)	Items (#)	Ratings (#)
electronics	18,649	3,975	23,009
kitchen	16,114	5,511	19,856
toys	9,924	3,451	13,147
dvd	49,151	14,608	124,438
music	69,409	24,159	174,180
video	11,569	5,223	36,180

Table II. Percentage of User Overlaps between Domains in the Amazon Dataset

Domains	electronics	kitchen	toys	dvd	music
kitchen	0.051				
toys	0.028	0.041			
dvd	0.040	0.037	0.031		
music	0.032	0.028	0.020	0.119	
video	0.029	0.029	0.028	0.317	0.058

category hierarchy, and so forth. This dataset contains 131,228 users, 317,755 items, and 1,127,673 reviews in total. It is a very sparse dataset with 0.003% sparsity. We employ the 10 categories with the most observed ratings of the Epinions dataset in our experiment.

The domains in the Amazon dataset only have user overlaps. Thus, there are no shared items between these domains. Table II illustrates the percentage of overlapped users between each pair of domains. As shown, the DVD and music domains have the most overlapped users (31.7%), whereas other domains have almost 3% to 5% of overlaps between their user sets. We randomly split 75% of each domain for training set and dedicate the remaining 25% to the test set.

In the following sections, we will compare these four methods:

—*Most-Pop.* This is our basic baseline [Steck 2010; Cremonesi et al. 2010], which uses the number of times that item i received the highest rating ($r_{ui} = 5$) for making the recommendation list. As personalized recommendations for cold-start users are inaccurate, recommending the most popular items seems to be a reasonable option.

—*Single-MF.* This is the single-domain matrix factorization technique employing information of unobserved ratings that is proposed in Steck [2010] and explained briefly in Section 2.4. In this method, we only employ the data from each domain's training set. Thus, comparing this single-domain method with our cross-domain methods will show us whether adding the extra information on auxiliary domains increases recommendation accuracy.

—*Cross-MF.* This is our strong baseline, which is the cross-domain extension of Single-MF. Thus, to test target domain d_j, we employ the training set of domain d_j adding all ratings from the auxiliary domains. This is the traditional way of dealing with cross-domain information [Hu et al. 2013]. Hence, it does not use cluster-level recommendation. Comparing our proposed method to Cross-MF shows whether our new method can utilize the data to achieve better recommendations.

—*Cross-CBMF.* This is our proposed cross-domain model, that aggregates the information of unobserved ratings from the user-item level and cluster level. As described in Section 2.3, our proposed model employs the same information as 'Cross-MF' but utilizes it with cluster level recommendations of coarse matrix R^c to achieve more accurate recommendations.

We first compare these methods for all users (Section 3.1). We then limit the set of users to ones with no ratings in the training set to compare the performance of these methods for cold-start users (Section 3.2). As discussed earlier, we employ top-N recommendation tasks as our evaluation metrics. Recall values are scaled in $[0, 1]$ for demonstrations. As Steck [2010] proposes, we take $w_{obs} = w^c_{obs} = 1$ and $r_m = 2$ in our experiment. Additionally, we tune our fixed parameters including α, w_m, w^c_m, λ, and λ^c via a cross-validation. We also iterate the learning process for five epochs to factorize both the user-item level and the cluster-level rating matrices. We run each experiment five times with different random initializations. Thus, we report the mean result of these five runs in the following sections.

As we show in Mirbakhsh and Ling [2013], the number of clusters should not be too large or small. If the number of clusters is too small, the predictions of the coarse matrix will be too general, whereas if the number of clusters is too large, these predictions will be very close to the item-user–level predictions. We use 100 clusters of items and 100 clusters of users for each domain in our experiment. Because of the large sparsity of the Epinions dataset, applying clustering on the set of items has not achieved clusters with a good quality. Thus, we only employ the clusters of users in our final experiment in the Epinions dataset.

3.1. Performance on All Users

We include the entire six selected domains in this setup. For each domain d_j, we employ the training set of domain d_j and all ratings from other domains to learn our cross-domain models. We then test the learned models on domain d_j's test set. Figure 5 presents this experiment, which compares the results of three methods: Single-MF, Cross-MF, and Cross-CBMF in Amazon dataset. As shown, Cross-CBMF significantly outperforms the other methods in the electronics, kitchen, and toys domains. Figure 6 presents the same comparison in the Epinions dataset.

In the Amazon dataset, for instance, for the DVD and music domains, the improvements are slight, but the results of Cross-CBMF and Cross-MF are almost the same for the video domain. It seems that employing cluster-level cross-domain information

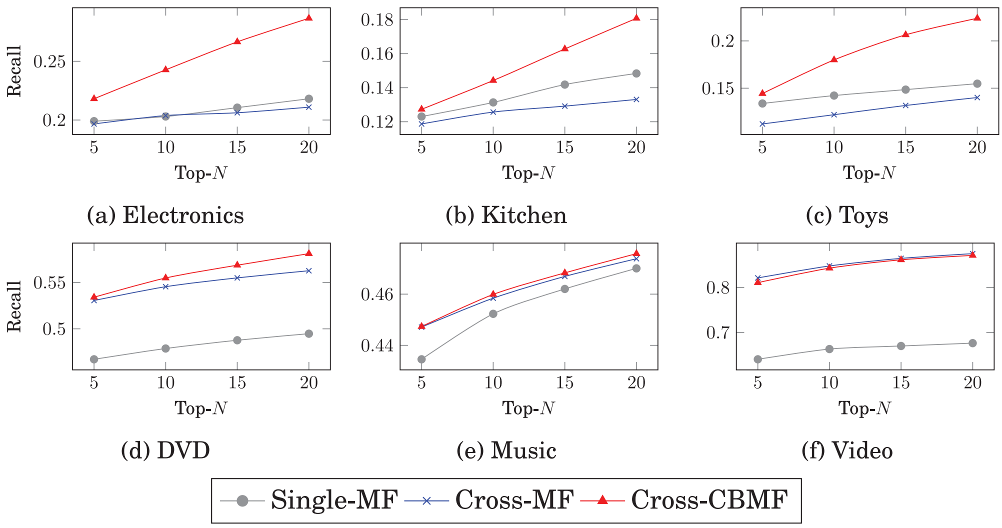

Fig. 5. Comparing Single-MF, Cross-MF, and Cross-CBMF for all users in six selected domains in the Amazon dataset. For each domain, information on the other five domains are included in the cross-domain methods.

is not helpful in this specific domain, possibly a result of the unbalanced distribution of this domain's ratings.

Note that the results of Most-Pop are removed from some figures, because this model achieves a very low recall in some setups. In the electronics domain from the Amazon dataset, for instance, Most-Pop achieves 0.041 recall for $N = 20$.

3.2. Performance on Cold-Start Users

As described earlier, collaborative filtering–based recommendation systems have a low performance for cold-start users. To evaluate the performance of our proposed method for cold-start users, we define cold-start users as ones who have made no ratings in the training set. Thus, single-domain methods have no collaborative information about these users. Figures 7 and 8 illustrate a comparison among selected methods. As shown, Cross-CBMF dramatically increases recall for all domains except the video domain (similar to Section 3.1) in the Amazon dataset. In the Epinions dataset, for 5 out of 10 selected domains, the improvements are significant but only slightly improved for the other domains.

In our experiment, we observed that the weight of unobserved ratings is much higher in the learning process of the coarse matrix than in the learning process of the user-item rating matrix. In the Amazon dataset, for instance, we use these fixed parameters (found via a cross-validation) in the setup that all domains are combined: $w_m = 0.0001$ and $\lambda = 0.1$ in Cross-MF for all domains, and $w_m^c = 0.9$ and $\lambda^c = 0.9$ in the Cross-CBMF model for most of the domains. The higher values are probably due to lower dimensionality and sparsity of the coarse matrices.

Figures 9 and 10 illustrate the change of recall for the Cross-CBMF method by employing different values of α. As defined in Equation (5), for $\alpha = 0$, we do not consider the effect of cluster-level recommendations, and the results are similar to Cross-MF. By increasing α, cluster-level recommendations have more influence in the aggregated result. As shown, other than in the video domain, we see improvement

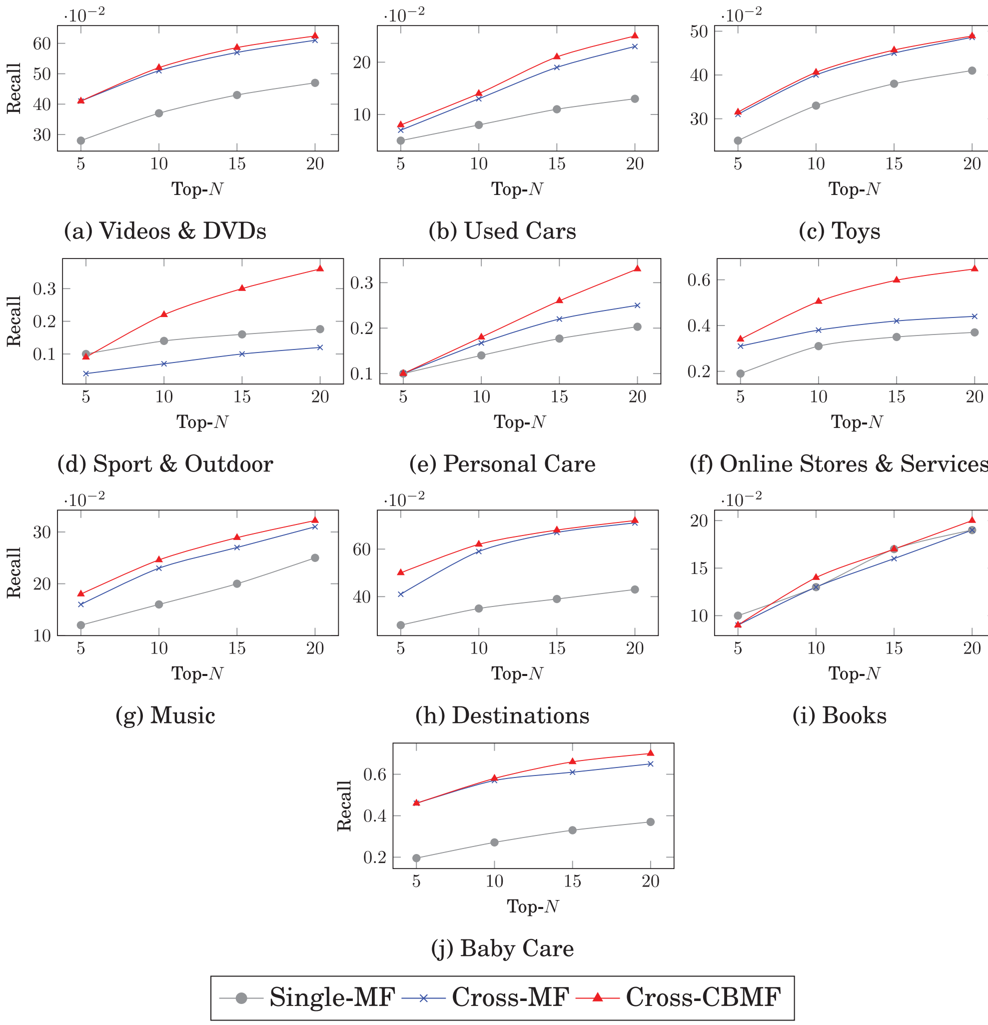

Fig. 6. Comparing Single-MF, Cross-MF, and Cross-CBMF for all users in the 10 selected domains in the Epinions dataset. For each domain, information on the other 9 domains are included in the cross-domain methods.

of recall in other domains taking appropriate values of α. In the video domain of the Amazon dataset, the value is the same for an alphavalue between 0 and 0.3.

To conclude, our experiments show that our proposed clustering-based matrix factorization model significantly increases the recall in top-N recommendation tasks for all users, particularly cold-start users. For example, for $N = 20$ in the Amazon dataset, our Cross-CBMF method achieves a recall of 43% on average for all users compared to 39% using Cross-MF. For cold-start users, our method improves recall to 21% on average, whereas including data from other domains using Cross-MF results in only 15% recall (for $N = 20$). In the Epinions dataset, our experiment shows an almost 25% total improvement in recall for cold-start users using our proposed Cross-CBMF. Note that it is often difficult to make even a small improvement of recommendations, especially for cold-start users. Hence, our result is quite significant.

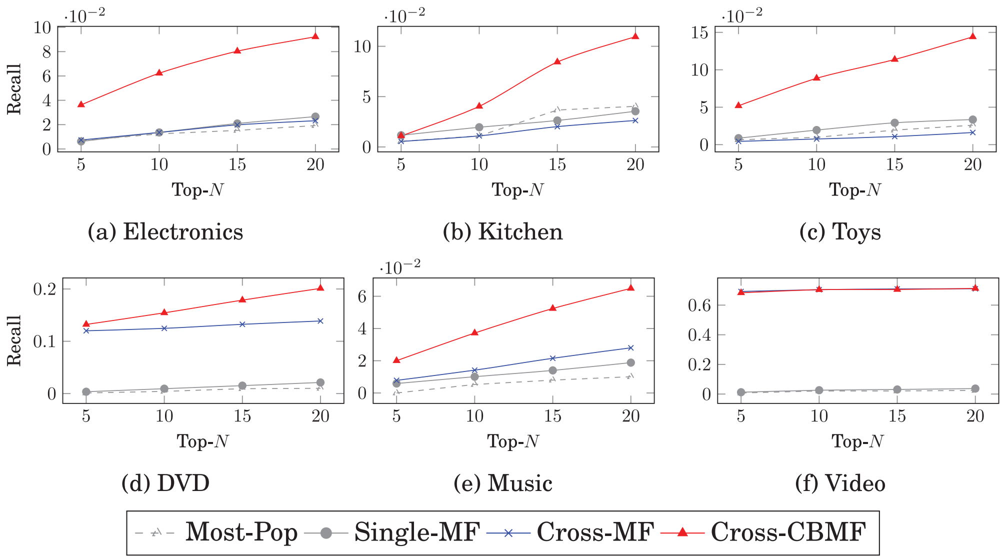

Fig. 7. Comparing the selected methods on cold-start users combining all six domains in the Amazon dataset.

4. RELATION TO PREVIOUS WORKS

As described earlier, employing unobserved ratings and efficient calculation of relations among the entire set of domains are two major contributions of this work. These two novelties distinguish our proposed method from current cross-domain recommendation methods. Employing unobserved ratings is shown to be dramatically effective in increasing recommendation accuracy [Steck 2010, 2013; Cremonesi et al. 2010; Ning and Karypis 2012; Rendle et al. 2009]. Consequently, we expect that our proposed method will outperform the current cross-domain methods on top-N recommendation tasks, as they are learned only based on observed ratings and with respect to improving prediction accuracy. This is because (as shown in Cremonesi et al. [2010]) methods with a good prediction accuracy do not always result in good recommendation accuracy. Moreover, many well-observed cross-domain methods, such as the proposed models in Cremonesi et al. [2011], Tang et al. [2012], and Li et al. [2009], require a heavy computation to find possible relations between each of the two domains. However, as we described in Section 2.2, we find these relations among the entire set of domains in one efficient step.

Li et al. [2009] propose a similar cluster-level integration of ratings in a cross-domain recommendation system. They adopt the orthogonal nonnegative matrix trifactorization algorithm to construct a cluster-level rating matrix, called *Codebook*.Our proposed coarse matrix (Equation (1)) is similar to this Codebook with two differences. First, the Codebook does not capture unobserved rating values among cluster of users and items. Second, these Codebooks are domain specific. Thus, for any two domains, they have to apply an expensive transferring algorithm, called *Codebook Transfer*, to find possible relations between pairs of Codebooks. However, our proposed method employs the extra information of unobserved ratings in the cluster level and user-item level. Our method also finds the relations among the coarse matrices in one step. Gao et al. [2013] propose an almost similar method based on Li's proposed Codebook. They consider similar explicit cluster-level latent space for users from different domains while the items in each domain may hold their domain-specific latentvectors. Moreno et al. [2012] also

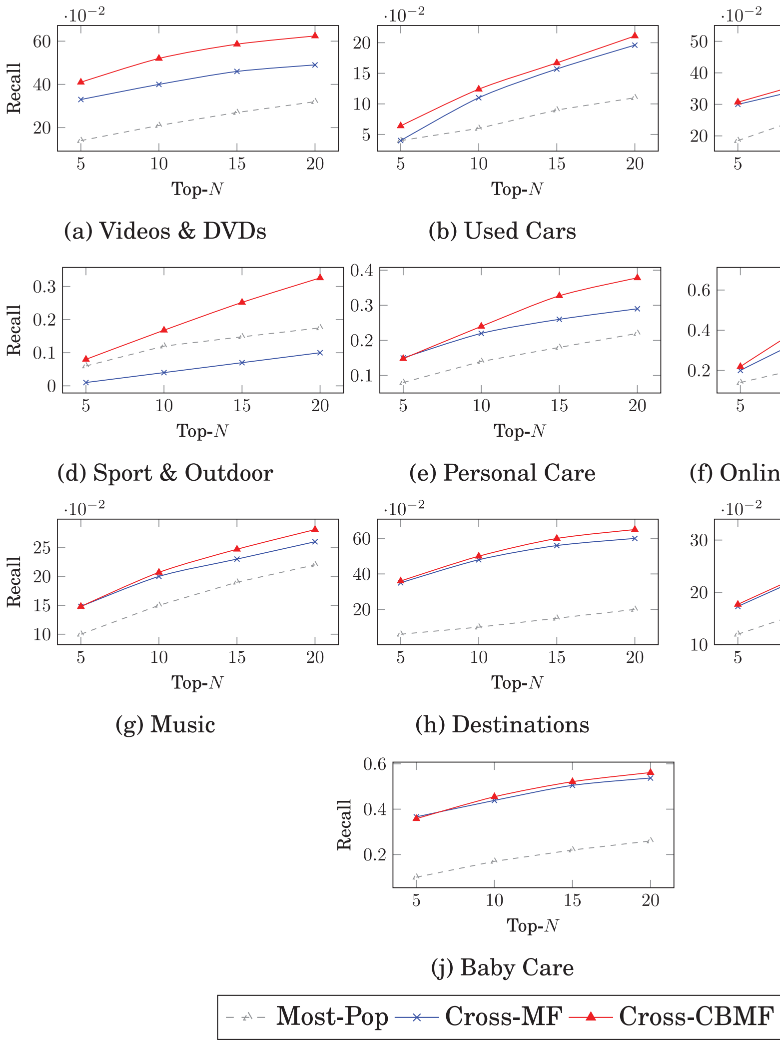

Fig. 8. Comparing the selected methods on cold-start users combining all 10 domains in the Epinions dataset.

generalize Li's method to transfer the auxiliary knowledge from multiple domains into one domain in contrast to Li's model, which is based on transferring knowledge from one domain to another. However, both methods suffer from the same limitations as Li's method.

Hu et al. [2013] integrate information from unobserved ratings into a cross-domain triadic factorization model. They merge domain-specific item-user rating matrices into a cubic user-item-domain rating matrix. Their proposed trifactorization model is then applied to factorize this cubic rating matrix into user, item, and domain latent space. They show empirically that their method outperforms unobserved ratings–integrated matrix factorization (same as the model that we refer to as Cross-MF in our experiment). However, cold-start users are ignored in their experiments, where they run their experiment over users with at least 30 observed ratings. Moreover, instead of finding user-item relations among different domains, we consider relations between

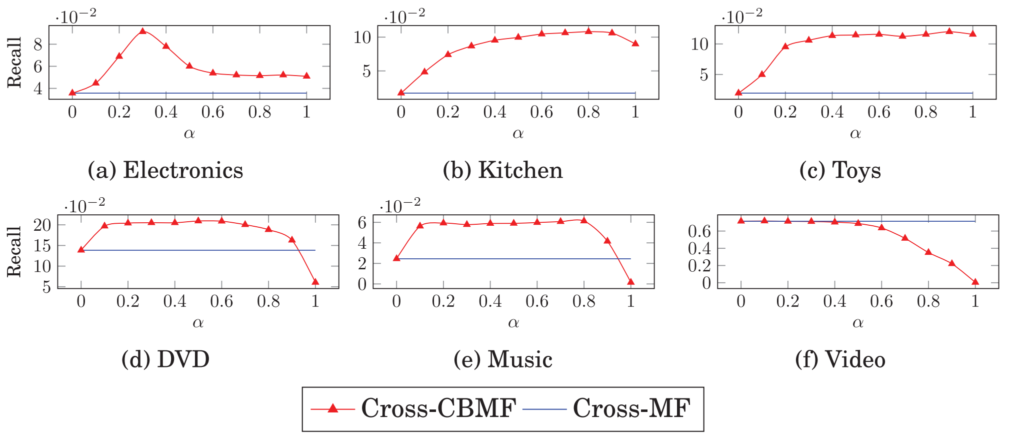

Fig. 9. Effect of changing α value on aggregated recommendations employing top-N evaluation ($N = 20$) in the Amazon dataset. Note that for $\alpha = 0$, the recall result is same as Cross-MF's result. The effect of cluster-level recommendations increases as α increases.

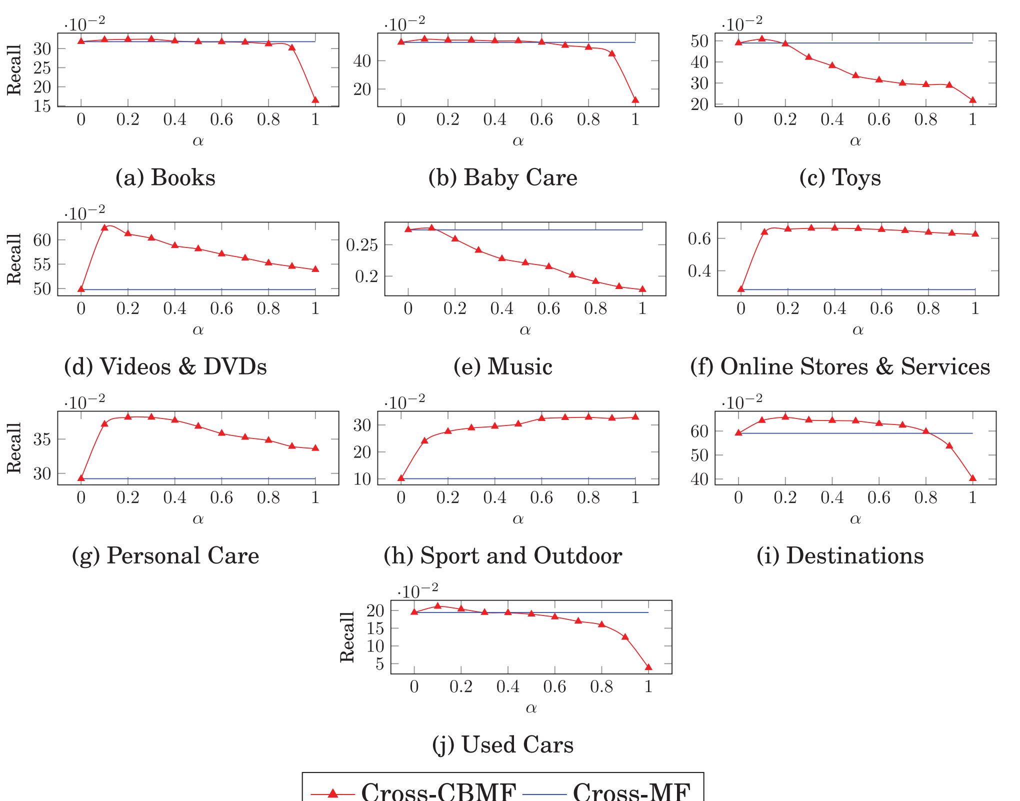

Fig. 10. Effect of changing the α value on aggregated recommendations employing top-N evaluation ($N = 20$) in the Epinions dataset. Note that for $\alpha = 0$, the recall result is the same as Cross-MF's result. The effect of cluster-level recommendations increases as α increases.

the cluster of users and items among different domains. In Section 3, we showed that our proposed method makes a significant improvement in top-N recommendation tasks for cold-start users.

Other related papers can be categorized into cross-domain transfer learning for recommendation and cross-domain collaborative filtering. Zhu et al. [2010] propose a heterogeneous transfer learning for image classification. Their proposed method employs matrix factorization to transfer useful knowledge in texts into an image classification model. Although they do not address the recommendation problem directly, their interesting model can be combined with our proposed model to make a context-aware recommendation system. Jiang et al. [2012] propose a novel hybrid random walk (HRW) to integrate multiple heterogeneous domains, such as users' social networks, to improve the recommendation accuracy. Random walk–based models generally provide simple solutions to integrate knowledge of different domains. However, they are sensitive to the number of steps in their random walk models, and their complexity dramatically increases by growing the number of steps. Lu et al. [2012] propose a Gaussian probabilistic latent semantic analysis (GPLSA) model that considers the consistency between the knowledge in two domains and only transfers the consistence auxiliary information between cross-domains. Their proposed selective transfer learning transfers the knowledge in user-item level, which can be compared to the proposed model of Hu et al. [2013]. However, we generalize the user-item matrix to reduce the sparsity. We then transfer the knowledge in two levels: cluster level and user-item level.

Chen et al. [2013] employthe same idea of clustering items and users in latent space and transferring the clustering-level knowledge between domains. However, in their proposed method, they need to learn a transition model between the clustering representation of each two domains, which is expensive to be generalized for multiple domains. This methods may be compared to Li's Codebooks Li et al. [2009]. They also employ the auxiliary information of items' contents and also users' social network into their cross-domain recommendation model. Again, our proposed model can easily handle including the auxiliary information from multiple domains without a need for learning a domain–domain mapping function.

In addition, Shi et al. [2014] and Li [2011] provide a complete survey on cross-domain recommendation systems. Pan and Yang [2010] present a complete survey on transfer learning in general for transferring knowledge between multiple domains in machine learning. Adomavicius and Tuzhilin [2005] also provide a good survey about state-of-the-art methods in recommendation systems.

Social networks are another important source of knowledge that can be included as auxiliary domains into cross-domain recommendation models. Jamali and Ester [2010] provide a promising model to include the social trusts to improve the recommendation accuracy. As mentioned earlier, Chen et al. [2013] include the social information as well as other auxiliary information for this purpose. Our proposed method may also be applied in social networks to include cluster-level information from social/trust networks for building a more accurate cross-domain personalization model.

5. CONCLUSIONS AND FUTURE WORK

To summarize, collaborative filtering–based recommendation systems are unable to make accurate recommendations for cold-start users. Employing the ratings that users have made in other domains is a natural solution to improve recommendations for cold-start users. However, most previous works in cross-domain recommendations ignore a significant part of available information—unobserved ratings. These methods also mainly focus on improving prediction accuracy, often observed in terms of RMSE, which has been highly criticized over the past few years.

In this article, we extend our previous work on clustering-based matrix factorization in single domains into cross-domains. We utilize recent results on unobserved ratings on two levels: the traditional user-item level and our novel cluster level. We define a cross-domain coarse matrix, which captures the shared preferences between clusters of users and cluster of items in same or different domains. Using this coarse matrix, we propagate the observed ratings into the cluster-level unobserved ratings to reduce the sparsity of traditional rating matrices. Finally, our proposed clustering-based matrix factorization aggregates the recommendations from these two levels. It effectively utilizes cross-domain data to improve recommendation accuracy. Our experiments show that our method improves top-N recommendation accuracy for all users, particularly cold-start users. For instance, our method achieves a recall of 43% on average for all users compared to 39% using the previous methods. For cold-start users, our method improves recall to 21% on average, whereas those previous methods result in 15% recall. We also observe almost 25% improvement of recall in the Epinions dataset. It is often difficult to make even a small improvement in recommendations, especially for cold-start users. Thus, our result is quite significant.

We consider a partial user and/or item overlap between domains. Knowledge transferring between nonoverlapped domains is emerging research in cross-domain recommendation systems. As future work, we are going to extend our method into nonoverlapped domain scenarios.

ACKNOWLEDGMENTS

This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca) and Compute/Calcul Canada. The authors would also like to thank the reviewers and the editor of TKDD for their helpful comments.

REFERENCES

Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6, 734–749. DOI:http://dx.doi.org/10.1109/TKDE.2005.99

Wei Chen, Wynne Hsu, and Mong Li Lee. 2013. Making recommendations from multiple domains. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*. ACM, New York, NY, 892–900. DOI:http://dx.doi.org/10.1145/2487575.2487638

Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-N recommendation tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys'10)*. ACM, New York, NY, 39–46. DOI:http://dx.doi.org/10.1145/1864708.1864721

Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. 2011. Cross-domain recommender systems. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW'11)*. 496–503. DOI:http://dx.doi.org/10.1109/ICDMW.2011.57

Sheng Gao, Hao Luo, Da Chen, Shantao Li, Patrick Gallinari, and Jun Guo. 2013. Cross-domain recommendation via cluster-level latent factor model. In *Machine Learning and Knowledge Discovery in Databases*. Lecture Notes in Computer Science, Vol. 8189. Springer, 161–176. DOI:http://dx.doi.org/10.1007/978-3-642-40991-2_11.

Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*. 595–606. http://dl.acm.org/citation.cfm?id=2488388.2488441.

Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys'10)*. ACM, New York, NY, 135–142. DOI:http://dx.doi.org/10.1145/1864708.1864736

Meng Jiang, Peng Cui, Fei Wang, Qiang Yang, Wenwu Zhu, and Shiqiang Yang. 2012. Social recommendation across multiple relational domains. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM'12)*. ACM, New York, NY, 1422–1431. DOI:http://dx.doi.org/10.1145/2396761.2398448

Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*. ACM, New York, NY, 426–434. DOI:http://dx.doi.org/10.1145/1401890.1401944

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8, 30–37. DOI:http://dx.doi.org/10.1109/MC.2009.263

Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. 2007. The dynamics of viral marketing. *ACM Transactions on the Web* 1, 1, Article No. 5. DOI:http://dx.doi.org/10.1145/1232722.1232727

Bin Li. 2011. Cross-domain collaborative filtering: A brief survey. In *Proceedings of the 2011 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI'11)*. 1085–1086. DOI:http://dx.doi.org/10.1109/ICTAI.2011.184

Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*. 2052–2057. http://dl.acm.org/citation.cfm?id=1661445.1661773

Zhongqi Lu, ErHeng Zhong, Lili Zhao, Evan Wei Xiang, Weike Pan, and Qiang Yang 0001. 2012. Selective transfer learning for cross domain recommendation. arXiv:1210.7056. Available at http://dblp.uni-trier.de/db/journals/corr/corr1210.html#abs-1210-7056.

Paolo Massa and Paolo Avesani. 2006. Trust-aware bootstrapping of recommender systems. In *Proceedings of the ECAI Workshop on Recommender Systems (ECAI'06)*. 29–33.

Simon Meyffret, Emmanuel Guillot, Lionel Medini, and Fredrique Laforest. 2012. *RED: A Rich Epinions Dataset for Recommender Systems*. Technical Report RR-LIRIS-2012-014. LIRIS UMR 5205 CNRS/INSA de Lyon/Universit Claude Bernard Lyon 1/Universit Lumire Lyon 2/cole Centrale de Lyon. http://liris.cnrs.fr/publis/?id=5787.

Nima Mirbakhsh and Charles X. Ling. 2013. Clustering-based factorized collaborative filtering. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys'13)*. ACM, New York, NY, 315–318. DOI:http://dx.doi.org/10.1145/2507157.2507233

Nima Mirbakhsh and Charles X. Ling. 2014. Leveraging clustering to improve collaborative filtering (submitted and under review).

Orly Moreno, Bracha Shapira, Lior Rokach, and Guy Shani. 2012. TALMUD: Transfer learning for multiple domains. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM'12)*. ACM, New York, NY, 425–434. DOI:http://dx.doi.org/10.1145/2396761.2396817

Xia Ning and George Karypis. 2012. Sparse linear methods with side information for top-N recommendations. In *Proceedings of the 6th ACM Conference on Recommender Systems (RecSys'12)*. ACM, New York, NY, 155–162. DOI:http://dx.doi.org/10.1145/2365952.2365983

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10, 1345–1359. DOI:http://dx.doi.org/10.1109/TKDE.2009.191

Al Mamunur Rashid, George Karypis, and John Riedl. 2008. Learning preferences of new users in recommender systems: An information theoretic approach. *SIGKDD Exploration Newsletter* 10, 2, 90–100. DOI:http://dx.doi.org/10.1145/1540276.1540302

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI'09)*. 452–461. http://dl.acm.org/citation.cfm?id=1795114.1795167

Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02)*. ACM, New York, NY, 253–260. DOI:http://dx.doi.org/10.1145/564376.564421

Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys* 47, 1, Article No. 3. DOI:http://dx.doi.org/10.1145/2556270

Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. ACM, New York, NY, 713–722. DOI:http://dx.doi.org/10.1145/1835804.1835895

Harald Steck. 2013. Evaluation of recommendations: Rating-prediction and ranking. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys'13)*. ACM, New York, NY, 213–220. DOI:http://dx.doi.org/10.1145/2507157.2507160

Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. 2012. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. ACM, New York, NY, 1285–1293. DOI: http://dx.doi.org/10.1145/2339530.2339730

Yin Zhu, Yuqiang Chen, Zhongqi Lu, Sinno Jialin Pan, Gui Rong Xue, Yong Yu, and Qiang Yang. 2010. Heterogeneous transfer learning for image classification. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10)*.

Received June 2014; revised December 2014; accepted January 2015