

Top-N Recommendation via Joint Cross-Domain User Clustering and Similarity Learning

Dimitrios Rafailidis¹ and Fabio Crestani²

¹ Department of Informatics, Aristotle University of Thessaloniki, Greece
draf@csd.auth.gr

² Faculty of Informatics, Università della Svizzera italiana (USI), Switzerland
fabio.crestani@usi.ch

Abstract. A cross-domain recommendation algorithm exploits user preferences from multiple domains to solve the data sparsity and cold-start problems, in order to improve the recommendation accuracy. In this study, we propose an efficient Joint cross-domain user Clustering and Similarity Learning recommendation algorithm, namely JCSL. We formulate a joint objective function to perform adaptive user clustering in each domain, when calculating the user-based and cluster-based similarities across the multiple domains. In addition, the objective function uses an $L_{2,1}$ regularization term, to consider the sparsity that occurs in the user-based and cluster-based similarities between multiple domains. The joint problem is solved via an efficient alternating optimization algorithm, which adapts the clustering solutions in each iteration so as to jointly compute the user-based and cluster-based similarities. Our experiments on ten cross-domain recommendation tasks show that JCSL outperforms other state-of-the-art cross-domain strategies.

Keywords: Collaborative filtering, cross-domain recommendation, alternating optimization.

1 Introduction

The collaborative filtering strategy has been widely followed in recommendation systems, where users with similar preferences tend to get similar recommendations [13]. User preferences are expressed explicitly in the form of ratings or implicitly in the form of number of views, clicks, purchases, and so on. Representative collaborative filtering strategies are matrix factorization techniques, which factorize the data matrix with user preferences in a single domain (e.g., music or video), to reveal the latent associations between users and items [14]. However, data sparsity and cold-start problems degrade the recommendation accuracy, as there are only a few preferences on which to base the recommendations in a single domain [5, 13].

With the advent of social media platforms and e-commerce systems, such as Amazon and Epinions, users express their preferences in multiple domains. For example, in Amazon users can rate items from different domains, such as

books and DVDs, or users express their opinion on different social media platforms, such as Facebook and Twitter. In the effort to overcome the data sparsity and cold-start problems, several cross-domain recommendation strategies have been proposed, which exploit the additional information of user preferences in multiple auxiliary domains to leverage the recommendation accuracy in a target domain [15]. However, generating cross-domain recommendations is a challenging task [5, 23]; for example, if the auxiliary domains are richer than the target domain, algorithms learn how to recommend items in the auxiliary domains and consider the target domain as noise. Moreover, the auxiliary domains might be a potential source of noise, for example, if user preferences differ in the multiple domains, the auxiliary domains introduce noise in the learning of the target domain. Therefore, a pressing challenge resides on how to transfer the knowledge of user preferences from different domains.

In cross-domain recommendation, the auxiliary domains can be categorized based on users and items overlap, that is, full-overlap, and partial or non user/item overlap between the domains [5]. In this study, we focus on partial user overlap between the target and the auxiliary domains, as it reflects on the real-world setting [8]. Relevant methods, such as [8, 20], form user and item clusters to capture the relationships between multiple domains at a cluster level, thus tackling the sparsity problem; and then weigh the cluster-based and user-based preferences to generate the top- N recommendations in the target domain. However, existing cluster-based cross-domain strategies have the following limitations, they form *non-adaptive* user and item clusters in a common latent space, when computing the cluster-based associations [8]; or they *linearly combine* the cluster-based and user-based relationships in the target domain [20].

1.1 Contribution

In this study, we overcome the aforementioned limitations in a novel approach for joint cross-domain recommendation based on user adaptive clustering and similarity learning. Our main contribution is summarized as follows, (i) we formulate an objective function to *jointly learn the user-based and cluster-based similarities across multiple domains, while adapting the user clusters in each domain at the same time*. To account for the fact that the user-based and cluster-based similarities across multiple domains are sparse, we use an $L_{2,1}$ -norm regularization to force the similarities to be sparse. (ii) We propose *an efficient alternating optimization algorithm to minimize the joint objective function*, thus computing the user-based similarities across the multiple domains. The user latent factors are weighted based on the calculated user-based similarities, to generate the final top- N recommendations. Our experiments on ten cross-domain recommendation tasks demonstrate the superiority of the proposed approach over competitive cross-domain strategies.

The remainder of the paper is organized as follows, Section 2 reviews the related study and in Section 3 we formally define the cross-domain recommendation problem. Section 4 formulates the proposed joint objective function, Section 5 presents our alternating optimization algorithm, and in Section 6 we elab-

orate on how to generate the top- N cross-domain recommendations. Finally, Section 7 presents the experimental results and Section 8 concludes the study.

2 Related Work

Cross-domain recommendation algorithms differ in how the knowledge of user preferences from the auxiliary domains is exploited, when generating the recommendations in the target domain [15, 23]. For example, various cross-domain approaches aggregate user preferences into a unified matrix, on which weighted single-domain techniques are applied, such as user-based kNN [2]. The graph-based method of [6] models the similarity relationships as a direct graph and explore all possible paths connecting users or items to capture the cross-domain relationships. Other methods exploit side information when linking multiple domains, on condition that the domains are linked by common knowledge, such as overlap of user/item attributes [4], social tags [1], and semantic networks [12]. However, such side information is not always available [20].

Other cross-domain techniques assume that the auxiliary and target domains are related by means of shared latent features. Representative methods are tri-matrix co-factorization, where user and item latent factors are shared between domains with different user preferences patterns. For example, Pan et al. [22] transform the knowledge of user preferences from different domains with heterogeneous forms of user feedback, that is, explicit or implicit feedback, to compute the shared latent features. Hu et al. [10] model a cubic user-item-domain matrix (tensor), and by applying factorization the respective latent space is constructed, based on which the cross-domain recommendations are generated.

More closely related to our approach, cross-domain strategies transfer patterns of user preferences between domains at a cluster level. Li et al. [16] calculate user and item clusters for each domain, and then encode the cluster-based patterns in a shared codebook; finally, the knowledge of user preferences is transferred across domains through the shared codebook. Gao et al. [8] compute the latent factors of user-clusters and item-clusters to construct a common latent space, which represents the preference patterns e.g., rating patterns, of user clusters on the item clusters. Then, the common cluster-based preference pattern that is shared across domains is learnt following a subspace strategy, so as to control the optimal level of sharing among multiple domains. Mirbakhsh and Ling [20] factorize a cluster-based coarse matrix, to capture the shared interests among user and item clusters. By factorizing the coarse matrix the preferences of users on items are computed at a cluster-level. By linearly combining the factorized cluster-based preferences with the individual user preferences, the recommendation accuracy is improved. However, both [8] and [20] use non-adaptive clustering strategies, when computing the cluster-based similarities.

Meanwhile, there are several graph-based algorithms that perform clustering on multiple domains such as the studies reported in [3, 7]. However, these techniques focus on grouping instances e.g., users from different domains, and do not generate cross-domain recommendations.

Table 1: Notation.

Symbol	Description
d	Number of domains
n_p	Number of users in the p -th domain, $p = 1, \dots, d$
m_p	Number of items in the p -th domain
$\mathbf{R}_p \in \mathbb{R}^{n_p \times m_p}$	User-item interaction (rating) matrix in the p -th domain
$\mathbf{A}_p \in \mathbb{R}^{n_p \times n_p}$	Adjacency matrix of the users' graph in the p -th domain
c_p	Number of user clusters in the p -th domain
$\mathbf{C}_p \in \mathbb{R}^{n_p \times c_p}$	Cluster assignment matrix in the p -th domain
$\mathbf{Y}_{pk} \in \mathbb{R}^{c_p \times c_k}$	Cluster-based cross domain matrix between domains p and k
$\mathbf{X}_{pk} \in \mathbb{R}^{n_p \times n_k}$	Cross domain matrix between the users in domains p and k

3 Problem Formulation

3.1 Notation

Our notation is presented in Table 1. We assume that we have d different domains, where n_p and m_p are the numbers of users and items in the p -th domain, respectively. In matrix \mathbf{R}_p , we store the user preferences on items, in the form of explicit feedback e.g., ratings or in the form of implicit feedback e.g., number of views, clicks, and so on. Based on the matrix \mathbf{R}_p , we capture the user-based similarities in the p -th domain. If users i and j have interacted with at least a common item q , then users i and j are connected. The connections/similarities are stored in an adjacency matrix \mathbf{A}_p , whose ij -th entries are calculated as follows [24]:

$$(\mathbf{A}_p)_{ij} = \begin{cases} \frac{\sum_{q=1}^{m_p} (\mathbf{R}_p)_{iq} (\mathbf{R}_p)_{jq}}{\sqrt{\sum_{q=1}^{m_p} (\mathbf{R}_p)_{iq}^2} \sqrt{\sum_{q=1}^{m_p} (\mathbf{R}_p)_{jq}^2}} & , \text{ if users } i \text{ and } j \text{ are connected} \\ 0 & , \text{ otherwise} \end{cases} \quad (1)$$

with $i, j = 1, \dots, n_p$.

3.2 Cross-Domain Similarities

In our approach, we consider two types of cross-domain similarities, that is, the *cluster-based* and the *user-based* cross-domain similarities, defined as follows:

Definition 1. (Cluster-based cross-domain similarities) For the p -th domain, we consider a cluster assignment matrix \mathbf{C}_p , with $(\mathbf{C}_p)_{ic}$ expressing the probability that user i belongs to cluster c . We define a cluster-based cross domain matrix $\mathbf{Y}_{pk} \in \mathbb{R}^{c_p \times c_k}$, where c_p and c_k are the numbers of user clusters in the p -th and k -th domains respectively. The entry $(\mathbf{Y}_{pk})_{ij}$ expresses the similarity between clusters i and j in the p -th and k -th domains, accordingly.

Definition 2. (User-based cross-domain similarities) We define a cross-domain matrix \mathbf{X}_{pk} between users in domains p and k . The entry $(\mathbf{X}_{pk})_{ij}$ expresses the similarity between users i and j in domains p and k , respectively.

3.3 Problem Definition

In the cross-domain recommendation task, we assume that we have a target domain p and $d - 1$ auxiliary domains. The goal is to predict the missing user preferences on items (recommendations) in the target domain p , while considering the user-based similarities in the rest of $d - 1$ auxiliary domains. Following the notation of matrix factorization, let $\mathbf{U}_p \in \mathbb{R}^{n_p \times l}$ and $\mathbf{V}_p \in \mathbb{R}^{m_p \times l}$ be the user and item latent factor matrices, with the factorized matrix $\hat{\mathbf{R}}_p = \mathbf{U}_p \mathbf{V}_p^T \in \mathbb{R}^{n_p \times m_p}$ containing the missing user preferences on items. As the i -th row of matrix \mathbf{U}_p , denoted by $(\mathbf{U}_p)_{i*}$, contains the l -dimensional user latent factor of user i , we can use a social regularization term $\Omega(\mathbf{U}_p)$, when learning the factorized matrix $\hat{\mathbf{R}}_p$ as follows [19]:

$$\min_{\mathbf{U}_p, \mathbf{V}_p} \|\mathbf{R}_p - \mathbf{U}_p \mathbf{V}_p^T\|_F^2 + \gamma(\|\mathbf{U}_p\|_F^2 + \|\mathbf{V}_p\|_F^2) + \Omega(\mathbf{U}_p) \quad (2)$$

where the first term is the approximation error between the factorized matrix $\hat{\mathbf{R}}_p$ and the initial data matrix \mathbf{R}_p ; the second one is the regularization term to avoid model overfitting with the parameter $\gamma > 0$; and the third term corresponds to the social regularization term based on the $d - 1$ auxiliary domains between the partial user overlaps. In the social regularization term $\Omega(\mathbf{U}_p)$, we have to weigh the influence of the user latent factors based on the user-based cross-domain similarities in \mathbf{X}_{pk} (Definition 2) as follows:

$$\Omega(\mathbf{U}_p) = \sum_{ij} \frac{1}{d-1} \sum_{k=1}^{d-1} (\mathbf{X}_{pk})_{ij} \|(\mathbf{U}_p)_{i*} - (\mathbf{U}_p)_{j*}\|^2, \quad \text{with } p \neq k \quad (3)$$

The term in the sum expresses the *approximation error between the user latent factors*, weighted by the user-based cross-domain similarities in \mathbf{X}_{pk} . The goal of the proposed approach is formally defined as follows:

Definition 3. (Problem) The goal of the proposed approach is to calculate the weights in \mathbf{X}_{pk} based on the preferences that users have in the different domains, in order to weigh the approximation error between the user latent factors $(\mathbf{U}_p)_{i*}$ and $(\mathbf{U}_p)_{j*}$ in (3).

4 Joint Cross-Domain Objective Function

User clustering. To simplify the presentation, from now on we assume that we have a target domain p and an auxiliary domain k . Given the adjacency matrix

\mathbf{A}_p (computed in (1)), first we have to define the objective function for performing user clustering on the p -th domain, that is, to calculate the cluster assignment matrix \mathbf{C}_p , which corresponds to the following minimization problem:

$$\begin{aligned} \min_{\mathbf{C}_p} \sum_{ij} \|(\mathbf{C}_p)_{i*} - (\mathbf{C}_p)_{j*}\|^2, \quad \text{with } i, j = 1, \dots, n_p \\ \text{subject to } \mathbf{C}_p^T \mathbf{C}_p = \mathbf{I}, \mathbf{C}_p \geq 0 \end{aligned} \quad (4)$$

with orthogonality constraints on the cluster matrix \mathbf{C}_p , and the user assignments to clusters being 0 or positive. According to the Laplacian method of [9], the minimization problem of (4) is equivalent to:

$$\begin{aligned} \min_{\mathbf{C}_p} \sum_{ij} \|(\mathbf{C}_p)_{i*} - (\mathbf{C}_p)_{j*}\|^2 = \min_{\mathbf{C}_p} Tr(\mathbf{C}_p^T \mathbf{L}_p \mathbf{C}_p) \\ \text{subject to } \mathbf{C}_p^T \mathbf{C}_p = \mathbf{I}, \mathbf{C}_p \geq 0 \end{aligned} \quad (5)$$

where $Tr(\cdot)$ is the trace operator. Matrix $\mathbf{L}_p \in \mathfrak{R}^{n_p \times n_p}$ is the Laplacian of the adjacency matrix \mathbf{A}_p , which is computed as follows: $\mathbf{L}_p = \mathbf{D}_p - \mathbf{A}_p$, where $\mathbf{D} \in \mathfrak{R}^{n_p \times n_p}$ is a diagonal matrix, whose entries are calculated as $(\mathbf{D}_p)_{ii} = \sum_{ij} (\mathbf{A}_p)_{ij}$.

Similarly, we define the respective objective function in (5), for performing user clustering on the auxiliary domain k , denoted by matrix $\mathbf{C}_k \in \mathfrak{R}^{n_k \times c_k}$.

Cluster-based and User-based Similarities. To compute the cluster-based and user-based similarities between domains p and k , we follow a co-clustering strategy [7], where we have to minimize the following objective function:

$$\begin{aligned} \min_{\mathbf{Y}_{pk}, \mathbf{X}_{pk}} \|\mathbf{X}_{pk} - \mathbf{C}_p \mathbf{Y}_{pk} \mathbf{C}_k^T\|_F^2 + \lambda_x \|\mathbf{X}_{pk}\|_{2,1} + \lambda_y \|\mathbf{Y}_{pk}\|_{2,1} \\ \text{subject to } \mathbf{Y}_{pk}^T \mathbf{Y}_{pk} = \mathbf{I}, \mathbf{Y}_{pk} \geq 0, \mathbf{X}_{pk} \geq 0 \end{aligned} \quad (6)$$

with orthogonality constraints on the cluster-based matrix \mathbf{Y}_{pk} , and the user-based and cluster-based (cross-domain) similarities being 0 or positive. Symbol $\|\cdot\|_{2,1}$ denotes the $L_{2,1}$ norm of a matrix which is calculated as follows [21]:

$$\|\mathbf{X}_{pk}\|_{2,1} = \sum_{i=1}^{n_p} \sqrt{\sum_{j=1}^{n_k} (\mathbf{X}_{pk})_{ij}^2} = \sum_{i=1}^{n_p} \|(\mathbf{X}_{pk})_{i*}\|_2 \quad (7)$$

The $L_{2,1}$ regularization terms in (6) force the solutions of matrices \mathbf{X}_{pk} and \mathbf{Y}_{pk} to be sparse, reflecting on the real-world scenario, where the user-based and cluster-based cross-domain similarities are usually sparse [5]. Parameters $\lambda_x, \lambda_y > 0$ control the respective $L_{2,1}$ regularization terms in (6).

Joint Objective Function. By combining (i) the objective function in (6) with (ii) the two clustering objective functions in (5) for domains p and k , we have to

minimize the following *joint objective function*:

$$\min_{\mathbf{C}_k, \mathbf{C}_p, \mathbf{Y}_{pk}, \mathbf{X}_{pk}} \mathcal{F} = \|\mathbf{X}_{pk} - \mathbf{C}_p \mathbf{Y}_{pk} \mathbf{C}_k^T\|_F^2 + \lambda_x \|\mathbf{X}_{pk}\|_{2,1} + \lambda_y \|\mathbf{Y}_{pk}\|_{2,1} + \beta_p \text{Tr}(\mathbf{C}_p^T \mathbf{L}_p \mathbf{C}_p) + \beta_k \text{Tr}(\mathbf{C}_k^T \mathbf{L}_k \mathbf{C}_k) \quad (8)$$

$$\text{subject to } \mathbf{C}_k^T \mathbf{C}_k = \mathbf{I}, \mathbf{C}_p^T \mathbf{C}_p = \mathbf{I}, \mathbf{Y}_{pk}^T \mathbf{Y}_{pk} = \mathbf{I}, \mathbf{C}_k, \mathbf{C}_p, \mathbf{Y}_{pk}, \mathbf{X}_{pk} \geq 0$$

where $\beta_p, \beta_k > 0$ are the regularization parameters for the clusterings in domains p and k , respectively.

5 Alternating Optimization

As the joint objective function $\mathcal{F}(\mathbf{C}_k, \mathbf{C}_p, \mathbf{Y}_{pk}, \mathbf{X}_{pk})$ in (8) is not convex with respect to the four variables/matrices, we propose an alternating optimization algorithm, where we update one variable, while keeping the remaining three fixed. The cluster assignment matrices $\mathbf{C}_k, \mathbf{C}_p$ are initialized by performing k -means to the respective adjacency matrices \mathbf{A}_k and \mathbf{A}_p , while \mathbf{Y}_{pk} and \mathbf{X}_{pk} are initialized by random (sparse) matrices. Next, we present the updating steps for each variable.

Step 1, fix $\mathbf{C}_p, \mathbf{Y}_{pk}, \mathbf{X}_{pk}$ and update \mathbf{C}_k . By considering the optimality condition $\partial \mathcal{F} / \partial \mathbf{C}_k = 0$, we calculate the partial derivative of \mathcal{F} with respect to \mathbf{C}_k :

$$\frac{\partial \mathcal{F}}{\partial \mathbf{C}_k} = -2\mathbf{X}_{pk}^T \mathbf{C}_p \mathbf{Y}_{pk} + 2\mathbf{C}_k \mathbf{Y}_{pk}^T \mathbf{C}_p^T \mathbf{C}_p \mathbf{Y}_{pk} + 2\beta_k \mathbf{L}_k \mathbf{C}_k \quad (9)$$

As the joint objective function \mathcal{F} in (8) is subject to the orthogonality constraints $\mathbf{C}_p^T \mathbf{C}_p = \mathbf{I}, \mathbf{Y}_{pk}^T \mathbf{Y}_{pk} = \mathbf{I}$, the second term of (9) equals $2\mathbf{C}_k$. By setting the partial derivative equal to zero we have to solve the following equation with respect to \mathbf{C}_k :

$$-\mathbf{X}_{pk}^T \mathbf{C}_p \mathbf{Y}_{pk} + \mathbf{C}_k + \beta_k \mathbf{L}_k \mathbf{C}_k = 0 \quad (10)$$

As $(\mathbf{I} + \beta_k \mathbf{L}_k)$ is a positive definite matrix, we can obtain the following closed-form solution (*updating rule*) of \mathbf{C}_k :

$$\mathbf{C}_k = (\mathbf{I} + \beta_k \mathbf{L}_k)^{-1} \mathbf{X}_{pk}^T \mathbf{C}_p \mathbf{Y}_{pk} \quad (11)$$

Step 2, fix $\mathbf{C}_k, \mathbf{Y}_{pk}, \mathbf{X}_{pk}$ and update \mathbf{C}_p . The partial derivative of \mathcal{F} with respect to \mathbf{C}_p is equivalent to:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{C}_p} = -2\mathbf{X}_{pk} \mathbf{C}_k \mathbf{Y}_{pk}^T + 2\mathbf{C}_p \mathbf{Y}_{pk} \mathbf{C}_k^T \mathbf{C}_k \mathbf{Y}_{pk}^T + 2\beta_p \mathbf{L}_p \mathbf{C}_p \quad (12)$$

Similarly, provided that \mathcal{F} is subject to $\mathbf{C}_k^T \mathbf{C}_k = \mathbf{I}, \mathbf{Y}_{pk}^T \mathbf{Y}_{pk} = \mathbf{I}$, we have the optimality condition by setting the partial derivative equal to zero:

$$-\mathbf{X}_{pk} \mathbf{C}_k \mathbf{Y}_{pk}^T + \mathbf{C}_p + \beta_p \mathbf{L}_p \mathbf{C}_p = 0 \quad (13)$$

Given that $(\mathbf{I} + \beta_p \mathbf{L}_p)$ is positive definite, we have the following *updating rule* for \mathbf{C}_p :

$$\mathbf{C}_p = (\mathbf{I} + \beta_p \mathbf{L}_p)^{-1} \mathbf{X}_{pk} \mathbf{C}_k \mathbf{Y}_{pk}^T \quad (14)$$

Step 3, fix \mathbf{C}_p , \mathbf{C}_k , \mathbf{X}_{pk} and update \mathbf{Y}_{pk} . The presence of the $L_{2,1}$ -norm regularization in the objective function \mathcal{F} of (8) makes the model difficult to optimize, as the algorithm cannot be guaranteed to convergence based on the analysis at [21]. To overcome this issue, we define a diagonal matrix $\mathbf{Q}_y \in \mathbb{R}^{c_p \times c_k}$ (with the same size as \mathbf{Y}_{pk}), whose entries are calculated as follows:

$$(\mathbf{Q}_y)_{ii} = \frac{1}{2\|(\mathbf{Y}_{pk})_{i*}\|_2} \quad (15)$$

thus, we can calculate the partial derivative of \mathcal{F} with respect to \mathbf{Y}_{pk} as follows:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{Y}_{pk}} = -2\mathbf{C}_p^T \mathbf{X}_{pk} \mathbf{C}_k + 2\mathbf{C}_p^T \mathbf{C}_p \mathbf{Y}_{pk} \mathbf{C}_k^T \mathbf{C}_k + 2\lambda_y \mathbf{Q}_y \mathbf{Y}_{pk} \quad (16)$$

where the last term corresponds to the partial derivative of the $L_{2,1}$ regularization term of \mathbf{Y}_{pk} in (8), with convergence guarantees [21]. As the joint objective function \mathcal{F} is subject to $\mathbf{C}_k^T \mathbf{C}_k = \mathbf{I}$, $\mathbf{C}_p^T \mathbf{C}_p = \mathbf{I}$, by setting the partial derivative of (16) equal to zero, we have:

$$-\mathbf{C}_p^T \mathbf{X}_{pk} \mathbf{C}_k + \mathbf{Y}_{pk} + \lambda_y \mathbf{Q}_y \mathbf{Y}_{pk} = 0 \quad (17)$$

which results in the following *update rule* for \mathbf{Y}_{pk} :

$$\mathbf{Y}_{pk} = (\mathbf{I} + \lambda_y \mathbf{Q}_y)^{-1} \mathbf{C}_p^T \mathbf{X}_{pk} \mathbf{C}_k \quad (18)$$

where $(\mathbf{I} + \lambda_y \mathbf{Q}_y)$ is a positive definite matrix.

Step 4, fix \mathbf{C}_k , \mathbf{C}_p , \mathbf{Y}_{pk} and update \mathbf{X}_{pk} . Similarly, given the $L_{2,1}$ regularization term of \mathbf{X}_{pk} in the joint objective function \mathcal{F} , we define the diagonal matrix $\mathbf{Q}_x \in \mathbb{R}^{n_p \times n_k}$, whose entries are computed as follows:

$$(\mathbf{Q}_x)_{ii} = \frac{1}{2\|(\mathbf{X}_{pk})_{i*}\|_2} \quad (19)$$

Then, we take the partial derivative of \mathcal{F} with respect to \mathbf{X}_{pk} :

$$\frac{\partial \mathcal{F}}{\partial \mathbf{X}_{pk}} = 2\mathbf{X}_{pk} - 2\mathbf{C}_p \mathbf{Y}_{pk} \mathbf{C}_k^T + \beta_x \mathbf{Q}_x \mathbf{X}_{pk} \quad (20)$$

By setting the partial derivative of (20) equal to zero, we obtain the following *update rule* for \mathbf{X}_{pk} :

$$\mathbf{X}_{pk} = (\mathbf{I} + \beta_x \mathbf{Q}_x)^{-1} \mathbf{C}_p \mathbf{Y}_{pk} \mathbf{C}_k^T \quad (21)$$

Analysis. The alternating optimization is performed iteratively, where in each iteration matrices \mathbf{C}_k , \mathbf{C}_p , \mathbf{Y}_{pk} and \mathbf{X}_{pk} are updated based on (11), (14), (18) and (21), respectively. More precisely, at each iteration each variable/matrix is recalculated based on the rest three matrices, which means that *each matrix*

is adapted to the values that the rest matrices have taken at the previous iteration, in order to reach a consensus solution for all four matrices over the iterations. The alternating optimization algorithm is repeated, until the algorithm converges. The optimization algorithm converges on condition that the joint objection function \mathcal{F} in (8) monotonically decreases after each iteration. Based on [21] the $L_{2,1}$ -norm regularization terms of \mathcal{F} are differentiable at zero, by using the diagonal matrices \mathbf{Q}_y and \mathbf{Q}_x in (15) and (19) when updating \mathbf{Y}_{pk} and \mathbf{X}_{pk} in (18) and (21), respectively³. By considering the optimality condition in each step, that is, setting the partial derivative of \mathcal{F} with respect to each variable equal to zero when updating the four variables, the proof that the algorithm converges is similar to the convergence analysis of [11].

6 Generating Top-N Recommendations

The joint objective function \mathcal{F} for $k = 1, \dots, d - 1$ auxiliary domains can be extended to:

$$\begin{aligned} \min_{\mathbf{C}_k, \mathbf{C}_p, \mathbf{Y}_{pk}, \mathbf{X}_{pk}} \mathcal{F} = & \sum_{k=1}^{d-1} \left[\|\mathbf{X}_{pk} - \mathbf{C}_p \mathbf{Y}_{pk} \mathbf{C}_k^T\|_F^2 + \lambda_x \|\mathbf{X}_{pk}\|_{2,1} + \lambda_y \|\mathbf{Y}_{pk}\|_{2,1} \right] \\ & + \beta_p \text{Tr}(\mathbf{C}_p^T \mathbf{L}_p \mathbf{C}_p) + \sum_{k=1}^{d-1} \beta_k \text{Tr}(\mathbf{C}_k^T \mathbf{L}_k \mathbf{C}_k) \end{aligned}$$

$$\text{subject to } \mathbf{C}_k^T \mathbf{C}_k = \mathbf{I}, \mathbf{C}_p^T \mathbf{C}_p = \mathbf{I}, \mathbf{Y}_{pk}^T \mathbf{Y}_{pk} = \mathbf{I}, \mathbf{C}_k, \mathbf{C}_p, \mathbf{Y}_{pk}, \mathbf{X}_{pk} \geq 0 \quad (22)$$

where the variables of \mathcal{F} are (i) the $d - 1$ clustering matrices \mathbf{C}_k of the auxiliary domains; (ii) the cluster matrix \mathbf{C}_p of the target domain p ; (iii) the $d - 1$ matrices \mathbf{Y}_{pk} , and \mathbf{X}_{pk} . An overview of our approach is presented in Algorithm 1.

7 Experimental Evaluation

7.1 Settings

Cross-domain recommendation tasks. Our experiments were performed on ten cross-domain tasks from the Rich Epinions Dataset (RED)⁴, which contains 131,228 users, 317,775 items and 1,127,673 user preferences, in the form of ratings. The items are grouped in categories/domains, and we evaluate the performance of cross-domain recommendation on the 10 largest domains. The evaluation data were provided by the authors of [20]. The main characteristics of the ten cross-domain recommendation tasks are presented in Table 2. The evaluation tasks are challenging, as the domains do not have item overlaps, but only user overlaps. In each cross-domain recommendation task, we consider one target domain and the remaining nine serve as auxiliary domains. For each task

³ At the first iteration matrices \mathbf{Q}_y and \mathbf{Q}_x are initialized by using the identity matrix [21].

⁴ <http://liris.cnrs.fr/red/>

ALGORITHM 1: Joint Cross-Domain User Clustering and Similarity Learning

Input: $\mathbf{R}_p, \mathbf{A}_p, \mathbf{R}_k, \mathbf{A}_k$, with $k = 1, \dots, d - 1$
Output: factorized matrix $\hat{\mathbf{R}}_p$

- 1 Initialize $\mathbf{C}_p, \mathbf{C}_k$ by performing k -means on \mathbf{A}_p and $\mathbf{A}_k, \forall k = 1, \dots, d - 1$
- 2 Initialize \mathbf{Y}_{pk} and $\mathbf{X}_{pk}, \forall k = 1, \dots, d - 1$
- 3 $convergence = False$
- 4 **while** $convergence = False$ **do**
- 5 Update \mathbf{C}_k based on (11), $\forall k = 1, \dots, d - 1$
- 6 Update \mathbf{C}_p based on (14)
- 7 Update \mathbf{Y}_{pk} based on (18), $\forall k = 1, \dots, d - 1$
- 8 Update \mathbf{X}_{pk} based on (21), $\forall k = 1, \dots, d - 1$
- 9 Calculate \mathcal{F} in (22) using the updated $\mathbf{C}_k, \mathbf{C}_p, \mathbf{Y}_{pk}, \mathbf{X}_{pk}, \forall k = 1, \dots, d - 1$
- 10 **if** \mathcal{F} converges
- 11 $convergence = true$
- 12 **end if**
- 13 **end while**
- 14 Calculate $\Omega(\mathbf{U}_p)$ based on the updated \mathbf{X}_{pk} and (3), $\forall k = 1, \dots, d - 1$
- 15 Calculate \mathbf{U}_p and \mathbf{V}_p based on $\Omega(\mathbf{U}_p)$ and (2)
- 16 $\hat{\mathbf{R}}_p = \mathbf{U}_p \mathbf{V}_p^T$

we preserve all the ratings of the auxiliary domains, and we randomly select 25%, 50% and 75% of the target domain as training set [20]. For each split, the remaining ratings of the target domain are considered as test set. We repeated our experiment five times, and we report mean values and standard deviations over the runs.

Table 2: The ten cross-domain recommendation tasks.

Domain	Users	Items	Ratings	Density (%)
Books	15,507	59,346	108,887	0.011
Baby Care	5,422	3,165	21,340	0.124
Destinations	9,290	3,615	31,418	0.093
Music	16,002	35,807	96,226	0.016
Online Stores & Services	28,643	5,518	54,734	0.034
Personal Care	6,214	10,786	28,945	0.043
Sport & Outdoor	6,750	9,597	19,181	0.029
Toys	9,040	18,681	51,152	0.030
Used Cars	17,041	4,174	28,598	0.040
Video & DVD	25,218	28,972	175,665	0.024

Compared methods. In our experiments, we evaluate the performance of the following methods:

- *NMF* [14]: a *single-domain* baseline Nonnegative Matrix Factorization method, which generates recommendations based only on the ratings of the target domain, ignoring the ratings in the auxiliary domains.

- *CLFM* [8]: a *cross-domain* Cluster-based Latent Factor Model which uses joint nonnegative tri-factorization to construct a latent space to represent the rating patterns of user clusters on the item clusters from each domain, and then generates the cross-domain recommendations based on a subspace learning strategy.
- *CBMF* [20]: a *cross-domain* Cluster-based Matrix Factorization model, which defines a coarse cross-domain matrix to capture the shared preferences between user and item clusters in the multiple domains, and then reveals the latent associations at a cluster level by factorizing the coarse cross-domain matrix. The final recommendations are generated by linearly combining the cluster-based latent associations and the user-based latent associations in the target domain. CBMF controls the influence of the cluster-based relationships on the personalized recommendations based on a parameter α .
- *JCSL*: the proposed Joint *cross-domain* user Clustering and Similarity Learning model.

In all models we varied the number of latent factors from [10,100] by a step of 10. In the cross-domain methods of CLMF, CBMF and JCSL we fixed the number of clusters to 100, as suggested in [20]. The predefined clusters in both the CLMF and CBMF methods are computed by performing k -means, also used in [8, 20]. Similarly, in the proposed JCSL method, the clusters are initialized by the k -means algorithm (Section 5). Following [8], in CLFM we tuned the dimensionality of the subspace up to the minimum number of latent factors of the multiple domains. In CBMF, we varied the α parameter in [0,1], where lower values of α consider to a fewer extent the cluster-based relationships, when computing the top- N recommendations. In JCSL the maximum number of iterations⁵ of the alternating optimization algorithm is fixed to 50, and the regularization parameters of the objective function in (22) were varied in [0.0001,0.1]. In all examined models, the parameters were determined via cross validation and in our experiments we report the best results.

Evaluation protocol. Popular commercial systems make top- N recommendations to users, and relevant studies showed that rating error metrics, such as RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) do not necessarily reflect on the top- N recommendation performance [5]. Therefore, in our experiments we used the ranking-based metrics Recall and Normalized Discounted Cumulative Gain to evaluate the top- N performance of the examined models directly [20]. Recall ($R@N$) is defined as the ratio of the relevant items in the top- N ranked list over all the relevant items for each user. The Normalized Discounted Cumulative Gain ($NDCG@N$) metric considers the ranking of the relevant items in the top- N list. For each user the Discounted Cumulative Gain

⁵ The algorithm terminates (converges) in less iterations, if $(\mathcal{F}^{(t+1)} - \mathcal{F}^{(t)})/\mathcal{F}^{(t)} \leq 1e - 03$, where $\mathcal{F}^{(t)}$ is the value of the objective function \mathcal{F} after the t -th iteration.

($DCG@N$) is defined as:

$$DCG@N = \sum_{j=1}^N \frac{2^{rel_j} - 1}{\log_2 j + 1} \quad (23)$$

where rel_j represents the relevance score of the item j to the user. $NDCG@N$ is the ratio of $DCG@N$ over the ideal $iDCG@N$ value for each user, that is, the $DCG@N$ value given the ratings in the test set. In our experiments we averaged $R@N$ and $NDCG@N$ over all users.

7.2 Results

In the first set of experiments, we use 75% of the target domain as training set, while the remaining is considered as test set. Table 3 presents the experimental results in terms of $NDCG@10$. The cross-domain methods of CLFM, CBMF and JCSL significantly outperform the single-domain NMF method, by exploiting the auxiliary domains when generating the recommendations. This happens because the cross-domain methods incorporate the additional information of user preferences on items from the auxiliary domains, thus reducing the data sparsity in the target domain. The proposed JCSL method achieves an 8.95% improvement on average when comparing with the second best method. Using the paired t-test we found that JCSL is superior over the rest approaches for $p < 0.05$. JCSL beats the competitive strategies, as it exploits the cluster-based similarities more efficiently than the competitive cluster-based models. The joint learning strategy of the adaptive user clustering while computing the user-based and cluster-based similarities, makes JCSL to efficiently incorporate the additional information of user preferences in the auxiliary domains. On the other hand, CLFM uses a subspace learning strategy on non-adaptive clusters in a common latent space. Finally, CBMF linearly combines the cluster-based and the individual latent associations by capturing the user preferences in the auxiliary domains based on predefined clusters. In this set of experiments there is the exceptional case of the ‘‘Baby Care’’ cross-domain task, where the proposed method has similar performance with CBMF. This happens because ‘‘Baby Care’’ is the less sparse domain, as presented in Table 2. Figure 1 compares the examined models in terms of recall ($R@N$), by varying the number of the top- N recommendations. Similarly, JCSL achieves a 12.17% improvement on average, for all the cross-domain recommendation tasks.

To evaluate the performance of the examined methods when sparsity increases, the training set is reduced to 25% of the target domain, while keeping all the ratings of the auxiliary domains. Table 4 reports the experimental results in terms of recall ($R@10$) based on the reduced training sets. In relation to the experimental results of Figure 1, recall drops for all methods, due to the increased sparsity. As we can observe, the proposed JCSL method achieves relatively high recall. In all cross-domain recommendation tasks, JCSL is superior to the competitive cross-domain strategies (for $p < 0.05$), by achieving on average relative improvement of 14.49%, when comparing with the second best method.

Table 3: Effect on $NDCG@10$ for the ten cross-domain recommendation tasks, using 75% of the target domain as training set. Bold values denote the best scores, for $*p < 0.05$ in paired t-test. The last column denotes the relative improvement (%), when comparing JCSL with the second best method (CBMF).

Target domain	NMF	CLFM	CBMF	JCSL	Improv. (%)
Books	.0997 \pm .0119	.1502 \pm .0308	.1780 \pm .0619	.1919 \pm .0472*	7.80*
Baby Care	.2054 \pm .0506	.3144 \pm .0345	.3875 \pm .0153	.3749 \pm .0460	-3.25
Destinations	.2991 \pm .0810	.3648 \pm .0961	.4271 \pm .0903	.4587 \pm .0805*	7.39*
Music	.1235 \pm .0263	.1631 \pm .0417	.1991 \pm .0399	.2109 \pm .0420*	7.43*
Online Stores	.1718 \pm .0909	.2550 \pm .0183	.3222 \pm .0320	.3665 \pm .0509*	13.74*
Personal Care	.1385 \pm .0173	.1677 \pm .0687	.2155 \pm .0311	.2491 \pm .0460*	15.59*
Sport & Outdoor	.1167 \pm .0290	.1469 \pm .0308	.1493 \pm .0917	.1665 \pm .0922*	11.52*
Toys	.1718 \pm .0649	.2269 \pm .0406	.2757 \pm .0453	.3017 \pm .0434*	9.43*
Used Cars	.0981 \pm .0841	.1301 \pm .0167	.1750 \pm .0471	.1897 \pm .0209*	8.15*
Video & DVD	.2413 \pm .0381	.3225 \pm .0350	.3984 \pm .0428	.4451 \pm .0428*	11.72*

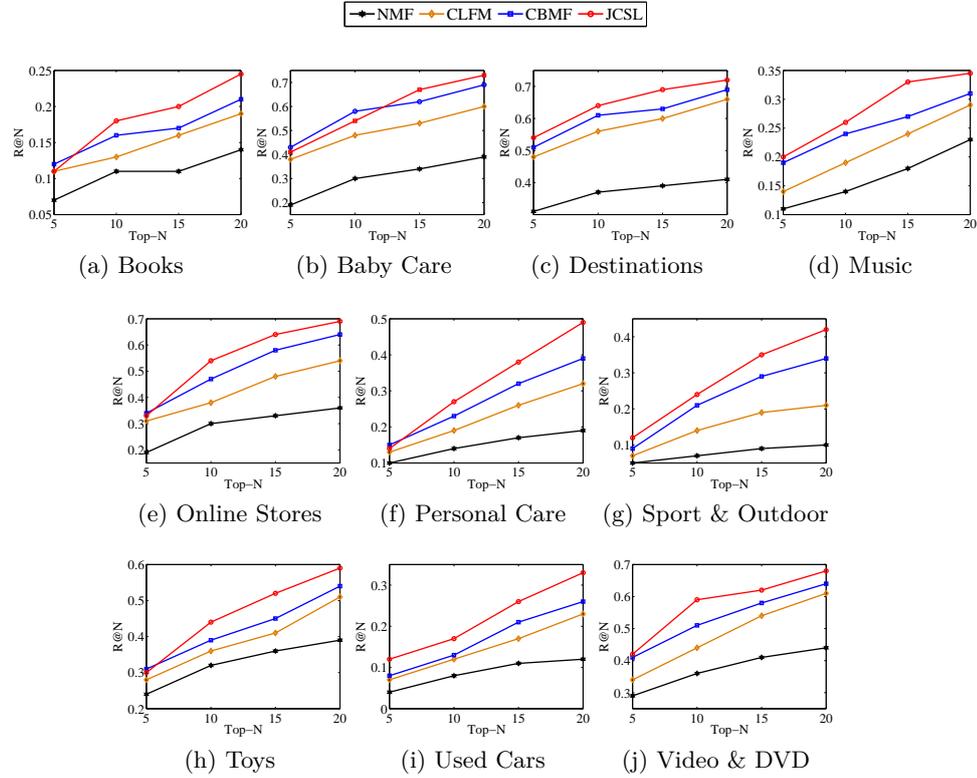


Fig. 1: Effect on $Recall (R@N)$ by varying the number of the top- N recommendations. In the ten cross-domain tasks, 75% of the target domain is considered as training set.

Table 4: Effect on *Recall* ($R@10$) for the ten cross-domain recommendation tasks, using 25% of the target domain as training set. Bold values denote the best scores, for $*p < 0.05$ in paired t-test. The last column denotes the relative improvement (%), when comparing JCSL with the second best method (CBMF).

Target domain	NMF	CLFM	CBMF	JCSL	Improv. (%)
Books	.0856 \pm .0348	.1108 \pm .0744	.1468 \pm .0390	.1779 \pm .0173*	21.18*
Baby Care	.1796 \pm .0947	.3411 \pm .0872	.4474 \pm .0752	.4968 \pm .1212*	11.04*
Destinations	.2604 \pm .0520	.3748 \pm .0869	.4365 \pm .0803	.5013 \pm .0720*	14.84*
Music	.1485 \pm .0431	.1805 \pm .0450	.2073 \pm .0869	.2304 \pm .0330*	11.14*
Online Stores	.2099 \pm .0112	.3151 \pm .0860	.3689 \pm .0401	.4230 \pm .0622*	14.74*
Personal Care	.1345 \pm .0152	.1690 \pm .0622	.1961 \pm .0560	.2232 \pm .0648*	13.81*
Sport & Outdoor	.0800 \pm .0853	.1239 \pm .0.751	.1622 \pm .0306	.1820 \pm .0294*	12.63*
Toys	.2448 \pm .0679	.3137 \pm .0585	.3497 \pm .0751	.3921 \pm .0655*	12.12*
Used Cars	.0730 \pm .0250	.1098 \pm .0959	.1160 \pm .0699	.1439 \pm .0427*	24.05*
Video & DVD	.2164 \pm .0162	.3685 \pm .0285	.4421 \pm .0892	.4834 \pm .0733*	9.34*

Figure 2 shows the effect on $NDCG@10$ of the cross-domain recommendation models, by varying the training sizes of three representative target domains, which are at different scale (Table 2). For presentation purposes, in this set of experiments the baseline NMF method is omitted, due to its low performance. We observe that all cross-domain methods increase the $NDCG$ metric, when a larger training set is used. Figure 2 shows that the proposed JCSL method keeps $NDCG$ relatively high in all settings, while outperforming CLFM and CBMF. The three cross-domain models differ in how the knowledge of user preferences is transferred between the domains when generating the recommendations, which explains their different performance when decreasing the training set size. JCSL adapts the clustering in each domain separately, while computing the cross-domain cluster-based similarities; whereas CLFM and CBMF compute the similarities between predefined/non-adaptive clusters, when generating the recommendations.

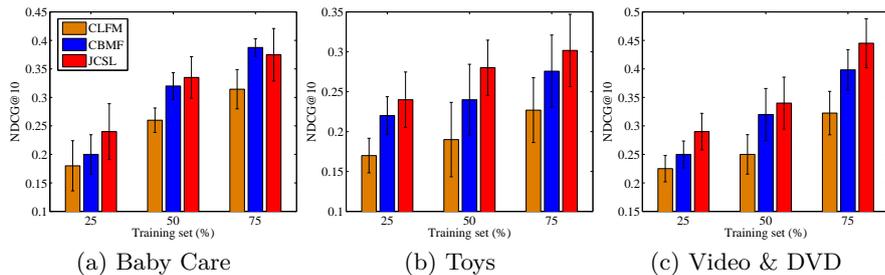


Fig. 2: Effect on $NDCG@10$ by varying the size of the training set.

8 Conclusion

In this paper, we presented an efficient cross-domain recommendation algorithm based on a joint strategy to adapt the user clusters, while calculating the user-based and cluster-based similarities across multiple domains. The joint optimization problem is solved via an efficient alternating optimization algorithm. Our experiments on ten cross-domain tasks confirmed the superiority of the proposed approach over competitive cross-domain strategies at different levels of sparsity. The main advantages of our approach are that JCSL adapts the clusters in each domain separately, while computing the cross-domain cluster-based similarities, whereas the competitors compute the similarities between predefined/non-adaptive clusters when generating the recommendations. Instead of linearly combining the cluster-based and user-based similarities, as for example CBMF does, JCSL jointly learns both types of similarities. In this study we considered partial user overlaps, with the mapping of users being known between the different domains. An interesting future direction is to extend our proposed approach for unknown user-matching across multiple domains [17]. In addition, an interesting future direction is to evaluate the performance of different clustering algorithms, such as spherical k -means [25] or power iteration [18], to initialize the clusters in the different domains.

Acknowledgments. We would like to thank Nima Mirbakhsh and Charles Ling for providing us with the evaluation data of the ten cross-domain tasks.

References

1. Abel, F., Araújo, S., Gao, Q., Houben, G.: [Analyzing cross-system user modeling on the social web](#). In: [Proceedings of the International Conference on Web Engineering, ICWE](#). pp. 28–43 (2011)
2. Berkovsky, S., Kuflik, T., Ricci, F.: [Distributed collaborative filtering with domain specialization](#). In: [Proceedings of the ACM Conference on Recommender Systems, RecSys](#). pp. 33–40 (2007)
3. Cheng, W., Zhang, X., Guo, Z., Wu, Y., Sullivan, P.F., Wang, W.: [Flexible and robust co-regularized multi-domain graph clustering](#). In: [Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD](#). pp. 320–328 (2013)
4. Chung, R., Sundaram, D., Srinivasan, A.: [Integrated personal recommender systems](#). In: [Proceedings of the International Conference on Electronic Commerce, ICEC](#). pp. 65–74 (2007)
5. Cremonesi, P., Koren, Y., Turrin, R.: [Performance of recommender algorithms on top-n recommendation tasks](#). In: [Proceedings of the ACM Conference on Recommender Systems, RecSys](#). pp. 39–46 (2010)
6. Cremonesi, P., Tripodi, A., Turrin, R.: [Cross-domain recommender systems](#). In: [Proceedings of the IEEE International Conference on Data Mining Workshops, ICDMW](#). pp. 496–503 (2011)
7. Du, L., Shen, Y.: [Towards robust co-clustering](#). In: [Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI](#) (2013)

8. Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., Guo, J.: Cross-domain recommendation via cluster-level latent factor model. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD, Part II. pp. 161–176 (2013)
9. [Gao, S., Tsang, I.W., Chia, L.: Laplacian sparse coding, hypergraph laplacian sparse coding, and applications. IEEE Trans. Pattern Anal. Mach. Intell. 35\(1\), 92–104 \(2013\)](#)
10. [Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., Zhu, C.: Personalized recommendation via cross-domain triadic factorization. In: Proceedings of the International World Wide Web Conference, WWW. pp. 595–606 \(2013\)](#)
11. [Jain, P., Netrapalli, P., Sanghavi, S.: Low-rank matrix completion using alternating minimization. In: Symposium on Theory of Computing Conference, STOC. pp. 665–674 \(2013\)](#)
12. [Kaminskas, M., Ricci, F., Schedl, M.: Location-aware music recommendation using auto-tagging and hybrid matching. In: Proceedings of the ACM Conference on Recommender Systems, RecSys. pp. 17–24 \(2013\)](#)
13. [Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer 42\(8\), 30–37 \(2009\)](#)
14. [Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems, NIPS. pp. 556–562 \(2000\)](#)
15. [Li, B.: Cross-domain collaborative filtering: A brief survey. In: Proceedings of the IEEE International Conference on Tools with Artificial Intelligence, ICTAI. pp. 1085–1086 \(2011\)](#)
16. [Li, B., Yang, Q., Xue, X.: Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI. pp. 2052–2057 \(2009\)](#)
17. [Li, C., Lin, S.: Matching users and items across domains to improve the recommendation quality. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD. pp. 801–810 \(2014\)](#)
18. [Lin, F., Cohen, W.W.: Power iteration clustering. In: Proceedings of the 27th International Conference on Machine Learning. pp. 655–662 \(2010\)](#)
19. [Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the International Conference on Web Search and Web Data Mining, WSDM. pp. 287–296 \(2011\)](#)
20. [Mirbakhsh, N., Ling, C.X.: Improving top-n recommendation for cold-start users via cross-domain information. TKDD 9\(4\), 33 \(2015\)](#)
21. [Nie, F., Huang, H., Cai, X., Ding, C.H.Q.: Efficient and robust feature selection via joint \$l_2, l_1\$ -norms minimization. In: Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems, NIPS. pp. 1813–1821 \(2010\)](#)
22. [Pan, W., Xiang, E.W., Liu, N.N., Yang, Q.: Transfer learning in collaborative filtering for sparsity reduction. In: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI \(2010\)](#)
23. [Shi, Y., Larson, M., Hanjalic, A.: Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. ACM Comput. Surv. 47\(1\), 3:1–3:45 \(2014\)](#)
24. [Tang, J., Hu, X., Liu, H.: Social recommendation: a review. Social Netw. Analys. Mining 3\(4\), 1113–1133 \(2013\)](#)
25. [Zhong, S.: Efficient online spherical k-means clustering. In: Proceedings of the IEEE Joint Conference on Neural Networks. pp. 3180–3185 \(2005\)](#)