Domain-Sensitive Recommendation with User-Item Subgroup Analysis

Jing Liu, *Member, IEEE*, Yu Jiang, Zechao Li, *Member, IEEE*, Xi Zhang, and Hanqing Lu, *Senior Member, IEEE*

Abstract—Collaborative Filtering (CF) is one of the most successful recommendation approaches to cope with information overload in the real world. However, typical CF methods equally treat every user and item, and cannot distinguish the variation of user's interests across different domains. This violates the reality that user's interests always center on some specific domains, and the users having similar tastes on one domain may have totally different tastes on another domain. Motivated by the observation, in this paper, we propose a novel Domain-sensitive Recommendation (DsRec) algorithm, to make the rating prediction by exploring the user-item subgroup analysis simultaneously, in which a user-item subgroup is deemed as a domain consisting of a subset of items with similar attributes and a subset of users who have interests in these items. The proposed framework of DsRec includes three components: a matrix factorization model for the observed rating reconstruction, a bi-clustering model for the user-item subgroup analysis, and two regularization terms to connect the above two components into a unified formulation. Extensive experiments on Movielens-100K and two real-world product review datasets show that our method achieves the better performance in terms of prediction accuracy criterion over the state-of-the-art methods.

Index Terms—Recommender system, matrix factorization, user-item subgroup, collaborative filtering

1 Introduction

Last decades have witnessed the overwhelming supply of online information with the evolution of the Internet. Thus, recommender systems have been indispensable nowadays, which support users with possibly different judgments and opinions in their quest for information, through taking into account the diversity of preferences and the relativity of information value. Collaborative Filtering (CF) is an effective and widely adopted recommendation approach. Different from content-based recommender systems which rely on the profiles of users and items for predictions, CF approaches make predictions by only utilizing the user-item interaction information such as transaction history or item satisfaction expressed in ratings, etc. As more attention is paid on personal privacy, CF systems become increasingly popular, since they do not require users to explicitly state their personal information [1].

Nevertheless, there still exist some problems which might limit the performance of typical CF methods. On one hand, user's interests always center on some specific domains but not all the domains. However, typical CF approaches do not treat these domains distinctively. On the other hand, the fundamental assumption for typical CF approaches is that users rate similarly on partial items, and hence they will rate on all the other items similarly. However, it is observed that this assumption is not always so tenable. Usually, the collaborative effect among users varies across different domains. In other words, two users have similar tastes in one domain cannot infer that they have similar taste in other domain. Taking an intuitive example, two users who love romantic movies probably have totally different preference in action movies. Thus, it is more reasonable and necessary to automatically mine different domains and perform domain sensitive CF for recommender systems.

Numerous efforts have been paid on this direction. Generally, these efforts can be divided into two types. The first type is to discover domains with the help of external information such as social trust network [2], product category information [3], etc. In this paper we focus on the second type called clustering CF, which only exploits the user-item interaction information and detects the domains by clustering methods. Among algorithms of this type, some are one-side clustering in the sense that they only consider to cluster either items or users [4], [5], [6], [7], [8]. And others are two-side clustering, which make use of the duality between users and items to partition both dimensions simultaneously [9], [10], [11], [12], [13]. In most of clustering CF approaches, each user or item is assigned to a single cluster (domain). However, in reality, the user interests and item attributes are not always exclusive, e.g., a user likes romantic movies does not means the user does not like other genre movies, and a romantic movie could also be an war movie. Thus, it is more natural to assume that a user or an item can join multiple domains. Besides, most of these clustering CF approaches are performed in a two-stage sequential process: domain detection by clustering and rating prediction by typical CF within the clusters. One advantage of this

- *J. Liu, Y. Jiang, X. Zhang, and H. Lu are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.*
E-mail: {jliu, yjiang, xi.zhang, luhq}@nlpr.ia.ac.cn.
- *Z. Li is with the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China.*
E-mail: zechao.li@gmail.com.

Manuscript received 21 Nov. 2014; revised 29 Sept. 2015; accepted 2 Oct. 2015. Date of publication 26 Oct. 2015; date of current version 3 Mar. 2016.
Recommended for acceptance by M. Spiliopoulou.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2015.2492540

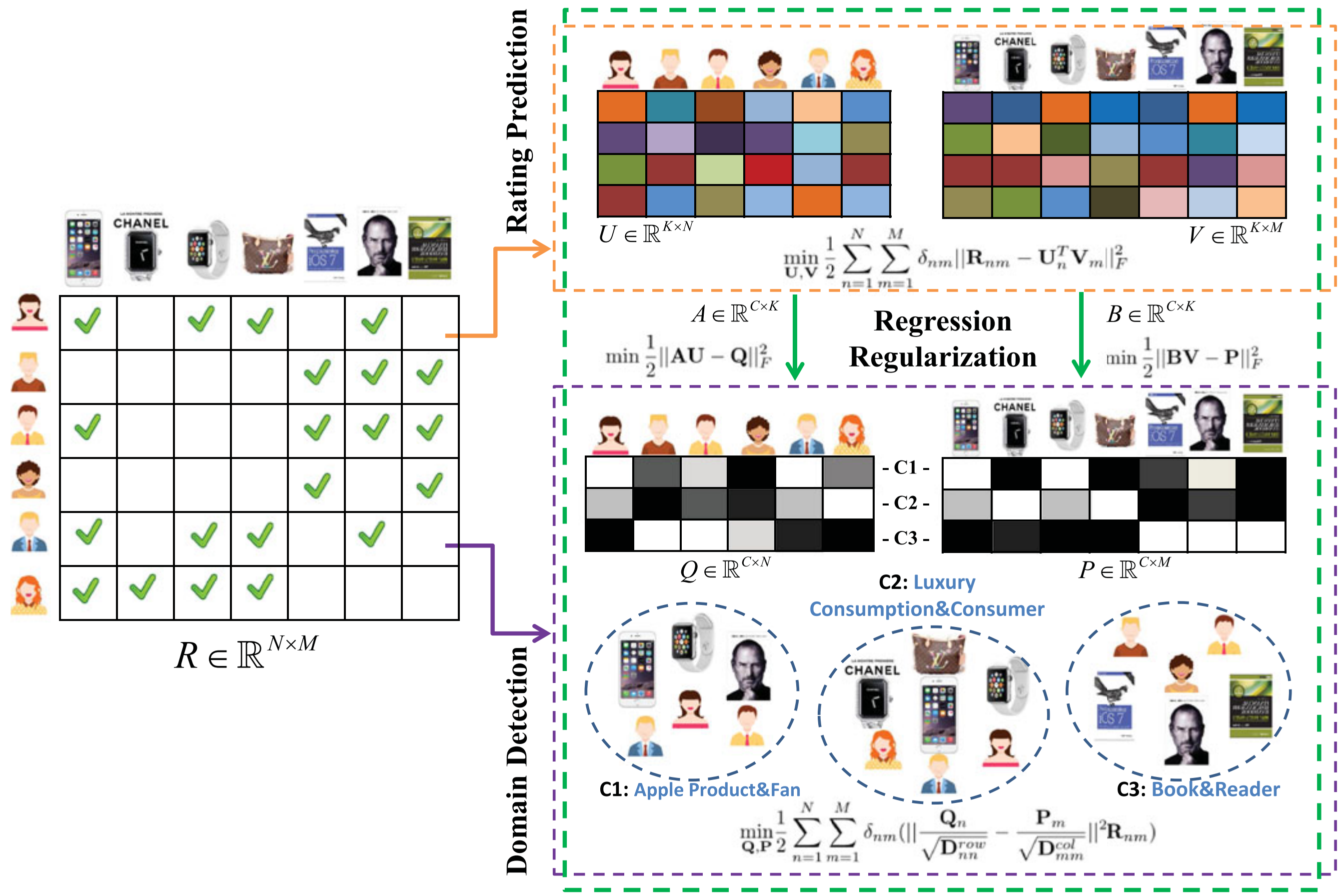


Fig. 1. Architecture of the proposed DsRec, which includes three components: rating prediction model (outlined with orange lines), domain detection model (outlined with purple lines), and regression regularization terms (outlined with green lines). $\mathbf{R}$ is the user-item rating matrix. $\mathbf{U}$ and $\mathbf{V}$ are user-specific and item-specific latent factor representations, and $\mathbf{Q}$ and $\mathbf{P}$ are confidence distribution matrices of users and items. $\mathbf{A}$ and $\mathbf{B}$ are the regression coefficient matrices.

approach is to overcome the problem of scalability brought by many memory-based CF techniques where the heavy computational burden is brought by the similarity calculations. However, such divide-and-conquer style brings a new problem, i.e., the algorithm cannot take full advantage of the observed rating data which is limited and precious.

To address above problems, in this paper, we propose a novel Domain-sensitive Recommendation (DsRec) algorithm assisted with the user-item subgroup analysis, which integrates rating prediction and domain detection into a unified framework. We call the proposed algorithm DsRec for short, and illustrate its basic architecture in Fig. 1. There are three components in the unified framework. First, we apply a matrix factorization model to best reconstruct the observed rating data with the learned latent factor representations of both users and items, with which those unobserved ratings to users and items can be predicted directly. Second, a bi-clustering model is used to learn the confidence distribution of each user and item belonging to different domains. Actually, a specific domain is a user-item subgroup, which consists of a subset of items with similar attributes and a subset of users interesting in the subset of items. In the bi-clustering formulation, we assume that a high rating score rated by a user to an item encourages the user and the item to be assigned to the same subgroups together. Additionally, two regression regularization items are imported to build a bridge between the confidence distribution of users (items) and the corresponding latent factor representations. That is, the confidence distribution over different subgroups (domains) in DsRec could be considered as soft pseudo domain labels, to guide the exploration of the latent space. Thus, connected with the regression regularizations, DsRec could learn discriminative and domain-sensitive latent spaces of users and items to perform the tasks of rating prediction and domain identification. To the best of our knowledge, our work is the first to jointly consider the both tasks by only utilizing user-item interaction information. An alternate optimization scheme is developed to solve the unified objective function, and the experimental analysis on three real-world datasets demonstrates the effectiveness of our method.

The remainder of this paper is organized as follows: In Section 2, we overview some related work. Sections 3 and 4 elaborate the proposed model of DsRec and an efficient iterative algorithm in details, respectively. The experimental evaluations and discussions are presented in Section 5. Section 6 concludes this paper with future research directions.

2 Related Work

2.1 Collaborative Filtering

According to [14], different CF techniques can be classified into two classes: memory-based methods and model based methods.

Memory-based methods focus on finding similar users [15] or items [16] for recommendation. These algorithms generally comply with the following steps: 1) calculate the similarity which reflects correlation between two users or two items. Popular similarity measures include Pearson correlation, vector similarity and various extensions of them [17]; 2) produce a prediction for the active user based on the ratings of similar users found, or based on the computed information of items similar to those chosen by the active user. Such recommender systems are easy-to-implement and highly effective. However, there are several limitations for the memory-based CF techniques, such as sparsity and scalability [17].

To overcome the shortcomings of memory-based methods, model-based approaches have been proposed and actively studied, which establish a model using the observed ratings that can interpret the given data and predict the unknown ratings. Many learning models have been used for modeling the rating process, such as Bayesian model [18], regression-based model [19], latent semantic model [20], clustering model and matrix factorization model [21], [22], [23], [24]. Among all these models, matrix factorization methods, such as Singular Value Decomposition (SVD) [21], Weighted Nonnegative Matrix Factorization (WNMF) [22], Maximum Margin Matrix Factorization (MMMF) [23] and Probabilistic Matrix Factorization (PMF) [24], are perhaps the most popular ones in recent years due to their efficiency in handling very huge datasets and the good performance in other applications [25], [26]. These methods assume that the features of both user and item lie in some low dimensional latent space and then make predictions based on the latent features.

2.2 Clustering Collaborative Filtering

A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. Thus, in essence, the task of clustering approach in clustering CF is to discover domains. Recently, with the development of internet, various contextual information as well as the rating matrix are integrated to discover some meaningful domains [27], [28], [29], [30], [31], where the typical contexts include item attributes, user trust, social network, and so on. That is, the above methods explore more broad knowledge to learn their recommendation models. In this paper, our so-called clustering CF methods are on a narrow scale, which only utilizes the user-item interaction information. Generally, such clustering CF methods can be categorized into three classes: user-based clustering CF, item-based clustering CF and two-side clustering CF.

User-based clustering CF is to partition users into distinct subgroups. In order to addresses the scalability problem, Han et al. [4] proposed a divide-and-conquer approach, which recursively splits the large-scale original rating data into two sub-groups, and makes the prediction within the sub-group that the active user belongs to. Sarwar et al. [5] clustered the complete user set based on user similarity and considered users in the same cluster as the neighbors. Xue et al. [6] generated clusters from the training data to provide the basis for data smoothing and neighborhood selection and then predicted the rating of a particular item for the active user by the behaviors of the neighbors. Bellogin and Parapar [7] utilized Normalized Cut in the recommendation process as a tool for user neighbor selection. In contrast, item-based clustering CF approaches cluster items into distinct subgroups. The representative work is proposed by O'Connor and Herlocker [8], which utilizes existing data partitioning and clustering algorithms to partition the set of items based on user rating data. Predictions are then conducted independently within each partition.

There are also many algorithms considering both sides simultaneously which are more similar to our methods. Ungar et al. [9] clustered users and items separately using variations of k-means and Gibbs sampling. Their CF performance on synthetic data is good, but not good on real data. Si and Jin [10] extended existing clustering algorithms for CF by clustering both users and items at the same time, allowing each user and item to be in multiple clusters and modeling the clusters of users and items separately. Banerjee et al. [11] presented a substantially generalized co-clustering framework wherein any Bregman divergence can be used in the objective function and applied this method to the CF task. George and Merugu [12] obtained user and item neighborhoods via co-clustering and generated predictions based on the average ratings of the co-clusters while taking the biases of the users and items into account. Xu et al. [13] proposed an unified framework to extend the traditional CF algorithms by utilizing the subgroups information to improve the top-N recommendation performance.

Our proposed approach is different from the above clustering CF methods. First, in most of the above methods, each user or item can be clustered into one single cluster only, whereas some recommender systems may benefit from the ability of clustering users and items into several clusters at the same time. Our user-item subgroup analysis allows a user or an item to appear in multiple subgroups. Second, for all above clustering CF methods, clustering is only an intermediate step. In other words, these methods are divided into two independent stages: clustering users or items into some sub-groups first, and then making rating prediction within each sub-group. Such sequential processing may not utilize the observed data fully. Thus, we integrate the rating prediction model and domain detection model into a unified framework, to best use of the available rating data and make the both stages boost each other. In addition, DsRec could handle the data sparsity problem better. Usually, data sparsity may lead that there are no common rated items for some users, who actually share similar interests. Thus, for the typical CF methods, the exact similarities among these users cannot be obtained (for memory-based CF), or the latent representations of these users may be different completely (for matrix factorization style model-based CF). In DsRec, these users may still be grouped into the same subgroup if their rated items have similar attributes and they would be closer in the latent space due to the same pseudo domain labels.

3 Our Approach

In this section, we first introduce an overview of our recommendation framework for DsRec, in which the motivation of our proposed method are also given. Then we discuss the main components consisted in the framework and their implementations.

3.1 Framework Overview

The goal of DsRec is to perform domain sensitive recommendation by jointly discovering user-item subgroups and predicting domain-specific user-item correlation, where only the user-item ratings are explored. It is motivated by the assumption that the collaborative effect among users varies across different domains. To achieve such a goal, we design a unified framework with three components: the factorization model for rating prediction, the bi-clustering model for domain detection, and the regression regularization items as the bridge between the above two models.

The basic ideas behind the three components are given as follows. First, the typical matrix factorization model is adopted to find user-specific and item-specific latent factors to reconstruct the observable user-item ratings, and we can utilize the learned factors to predict the rating of any user-item pair. Second, a bi-clustering model is formulated to make full use of the duality between users and items to cluster them into subgroups. The underlying assumption is that the labels of a user and an item for their subgroup identification should be the same if they are strongly associated, i.e., a high rated user-item pair should be grouped together. Third, the regression regularization attempts to learn the mappings from the latent factor representations of users (and items) to their confidence distribution belonging to different subgroups, where the former is learned from the factorization model, and the latter is explored in the bi-clustering model. Simply, the rating prediction model and the domain detection model are both estimated based on the observable user-item ratings. The regression terms are considered as a bridge between the both above models, in order to learn more discriminative latent spaces of users and items for recommendation and domain identification. From this view, the unified model is tightly integrated with the three models, and they enhance each other.

3.2 Rating Prediction Model

As a typical solution, matrix factorization is adopted for rating prediction in our work. Suppose we have a user-item rating matrix $\mathbf{R} \in \mathbb{R}^{N\times M}$ describing N users' numerical ratings on M items. Since in the real-world, each user usually rates a very small portion of items, the matrix $\mathbf{R}$ is extremely sparse. A matrix factorization approach seeks to approximate the rating matrix $\mathbf{R}$ by a multiplication of K-rank factors,

$$\mathbf{R} \approx \mathbf{U}^T\mathbf{V}, \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{K\times N}$ and $\mathbf{V} \in \mathbb{R}^{K\times M}$ with $K < \min(N, M)$. Column vectors $\mathbf{U}_n$ and $\mathbf{V}_m$ represent user-specific and item-specific latent feature vectors, respectively.

Traditionally, the SVD method is utilized to approximate a rating matrix $\mathbf{R}$ by

$$\min \frac{1}{2}||\mathbf{R} - \mathbf{U}^T\mathbf{V}||_F^2, \tag{2}$$

where $||.||_F^2$ denotes the Frobenius norm. However, due to the reason that $\mathbf{R}$ contains a large number of missing values, we only need to factorize the observed ratings in $\mathbf{R}$. Hence, we rewrite the Eq. (2) as

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\delta_{nm}||\mathbf{R}_{nm} - \mathbf{U}_n^T\mathbf{V}_m||_F^2, \tag{3}$$

where δ_{nm} is an indicator function that is equal to 1 if the n-th user rates the m-th item and equal to 0 otherwise.

In order to avoid overfitting, two regularization terms are added into Eq. (3). Hence we have

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\delta_{nm}||\mathbf{R}_{nm} - \mathbf{U}_n^T\mathbf{V}_m||_F^2 + \frac{\lambda_u}{2}||\mathbf{U}||_F^2 + \frac{\lambda_v}{2}||\mathbf{V}||_F^2, \tag{4}$$

where $\lambda_u, \lambda_v > 0$. The optimization problem in Eq. (4) also contains a nice probabilistic interpretation with Gaussian observation noise, which is detailed in [24].

3.3 Domain Detection Model

In this section, we will systematically interpret how to detect user-item subgroups (domains) with a bi-clustering model, which is also a two-sided clustering solution. It has been shown that the two-sided clustering often yields impressive performance over traditional one-sided clustering algorithms. More importantly, the resulting co-clustered subgroups may reveal valuable insights from the item attributes. For example, John likes both iPhone6Plus and a Louis Vuitton bag. If our bi-clustering model groups John, iPhone6Plus, and Louis Vuitton together, we can explain the domain is related to luxury consumption, since both iPhone6Plus and Louis Vuitton are luxury products. Different from [2], [3], the only information we have is the user-item rating matrix $\mathbf{R}$. Considering that user tastes and item attributes are not exclusive but diverse (e.g., John prefers to consume both electronic devices and luxury products), the goal of user-item subgroup analysis is to simultaneously assign the N users and M items with the confidence distribution over C subgroups. Different from the traditional Co-Clustering algorithem [32], [33], our user-item subgroup analysis allows a user or an item to appear in multiple subgroups and confidence distributions are required instead of partition matrices.

In our framework, the domain detection model works on the assumption that a high rating score rated by a user to an item encourages the user and the item to be assigned to the same subgroups together. As a transitive result, the items highly rated by the same user are probably grouped together, and the same to the case that the users who prefers to the same item. Accordingly, we can obtain some user-item subgroups each consisting of a subset of items and a group of users who interested in those items. The above simple example about John's consumption can also be used to explain the idea of demain detection. Since John gives high rating scores to the both products of iPhone6Plus and a Louis Vuitton bag, we think John, iPhone6Plus, and the Louis Vuitton bag should belong to the same subgroup.

Specifically, we model the relationship between users and items using a bipartite graph. In the bipartite graph model, N users and M items are two sets of vertices. An edge between the n-th user and the m-th item exists if and only if the user has rated the item, and the edge weight is set to be $\mathbf{R}_{nm}$, which represents the association between the user-item pair. Let $\mathbf{Q} \in \mathbb{R}^{C\times N}$ and $\mathbf{P} \in \mathbb{R}^{C\times M}$ be confidence distribution matrices for users and items, and the column vectors $\mathbf{Q}_n$ and $\mathbf{R}_m$ indicate the confidence distribution of the n-th user and the m-th item over C subgroups respectively. In order to make those strongly associated users and items (i.e., the items are highly rated by the users) be grouped into the same subgroups, we minimize the following loss function:

$$\min_{\mathbf{Q},\mathbf{P}} \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\delta_{nm}\left(\left\|\frac{\mathbf{Q}_n}{\sqrt{\mathbf{D}^{row}_{nn}}}-\frac{\mathbf{P}_m}{\sqrt{\mathbf{D}^{col}_{mm}}}\right\|^2\mathbf{R}_{nm}\right), \quad \text{s.t.}\quad \mathbf{Q},\mathbf{P}\geq 0,\quad \mathbf{Q}^T\mathbf{1}_C=\mathbf{1}_N,\quad \mathbf{P}^T\mathbf{1}_C=\mathbf{1}_M, \tag{5}$$

where $\mathbf{D}^{row} \in \mathbb{R}^{N\times N}$ is the diagonal degree matrix of users with $\mathbf{D}^{row}_{nn} = \sum_{m=1}^{M}\delta_{nm}\mathbf{R}_{nm}$ and $\mathbf{D}^{col} \in \mathbb{R}^{M\times M}$ is the diagonal degree matrix of items with $\mathbf{D}^{col}_{mm} = \sum_{n=1}^{N}\delta_{nm}\mathbf{R}_{nm}$. And the combined constraints $\mathbf{Q},\mathbf{P}\geq 0$ and $\mathbf{Q}^T\mathbf{1}_C=\mathbf{1}_N$, $\mathbf{P}^T\mathbf{1}_C=\mathbf{1}_M$ force each element of $\mathbf{Q}$, $\mathbf{P}$ to stay in the range of $[0,1]$.

Though user tastes and item attributes are not exclusive, they should not be dispersive over too many domains. Thus, two 1-norm regularization items, encouraging confidence distributions sparse, are added to Eq. (5). Hence, we have

$$\begin{aligned}\min_{\mathbf{Q},\mathbf{P}} &\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\delta_{nm}\left(\left\|\frac{\mathbf{Q}_n}{\sqrt{\mathbf{D}^{row}_{nn}}}-\frac{\mathbf{P}_m}{\sqrt{\mathbf{D}^{col}_{mm}}}\right\|^2\mathbf{R}_{nm}\right)\\ &+\frac{\eta_q}{2}\sum_{n=1}^{N}||\mathbf{Q}_n||_1+\frac{\eta_r}{2}\sum_{m=1}^{M}||\mathbf{P}_m||_1,\\ \text{s.t.}\quad &\mathbf{Q},\mathbf{P}\geq 0,\quad \mathbf{Q}^T\mathbf{1}_C=\mathbf{1}_N,\quad \mathbf{P}^T\mathbf{1}_C=\mathbf{1}_M,\end{aligned} \tag{6}$$

where $\eta_q, \eta_r > 0$.

3.4 Regression Regularization Terms

The idea behind the regression regularization is that the latent factor representations of users (and items) are required to reflect the preferences of users (and the attributes of items) across different domains. In other words, the latent factor representations should be discriminative enough to find the subgroup distributions of users and items, and in turn some domain information should be embedded into the latent factor representations for rating prediction.

In Eq. (6) above, $\mathbf{Q}$ and $\mathbf{P}$, denote the confidence distribution matrices for users and items over different domains. The confidence distributions can be considered as soft pseudo domain labels to guide the construction of the user-specific latent space $\mathbf{U}$ and the item-specific latent space $\mathbf{V}$. That is, the goal of the regression regularization is to find the mapping matrices from the user-specific latent space $\mathbf{U}$ and the item-specific latent space $\mathbf{V}$ to the subgroup identification vectors of $\mathbf{Q}$ and $\mathbf{P}$ respectively.

For user-specific latent factor representation $\mathbf{U}$ and the users' confidence distribution matrix $\mathbf{Q}$, we attempt to learn a mapping matrix $\mathbf{A} \in \mathbb{R}^{C\times K}$ based on the following regression loss function:

$$\min \frac{1}{2}||\mathbf{AU}-\mathbf{Q}||_F^2+\frac{\gamma_a}{2}||\mathbf{A}||_F^2, \tag{7}$$

where $\gamma_a > 0$, and the penalty term $\frac{\gamma_a}{2}||\mathbf{A}||_F^2$ is introduced to avoid the overfitting problem.

Similarly, we define the loss function for item-specific latent factor representation $\mathbf{V}$ and the items' confidence distribution matrix $\mathbf{R}$ as follows:

$$\min \frac{1}{2}||\mathbf{BV}-\mathbf{P}||_F^2+\frac{\gamma_b}{2}||\mathbf{B}||_F^2, \tag{8}$$

where $\gamma_b > 0$, and $\mathbf{B} \in \mathbb{R}^{C\times K}$ is the learned mapping matrix similar to $\mathbf{A}$ in Eq. (7) .

3.5 The Unified Model

To integrate the above three components into a unified framework, we get the final objective function as follows:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V}} &\mathcal{O}(\mathbf{U},\mathbf{V},\mathbf{Q},\mathbf{P},\mathbf{A},\mathbf{B})\\ &=\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\delta_{nm}||\mathbf{R}_{nm}-\mathbf{U}_n^T\mathbf{V}_m||_F^2\\ &\quad+\frac{\lambda_u}{2}||\mathbf{U}||_F^2+\frac{\lambda_v}{2}||\mathbf{V}||_F^2\\ &\quad+\frac{\alpha}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\delta_{nm}\left(\left\|\frac{\mathbf{Q}_n}{\sqrt{\mathbf{D}^{row}_{nn}}}-\frac{\mathbf{P}_m}{\sqrt{\mathbf{D}^{col}_{mm}}}\right\|^2\mathbf{R}_{nm}\right)\\ &\quad+\frac{\alpha\eta_q}{2}\sum_{n=1}^{N}||\mathbf{Q}_n||_1+\frac{\alpha\eta_r}{2}\sum_{m=1}^{M}||\mathbf{P}_m||_1\\ &\quad+\frac{\beta_a}{2}||\mathbf{AU}-\mathbf{Q}||_F^2+\frac{\beta_a\gamma_a}{2}||\mathbf{A}||_F^2\\ &\quad+\frac{\beta_b}{2}||\mathbf{BV}-\mathbf{P}||_F^2+\frac{\beta_b\gamma_b}{2}||\mathbf{B}||_F^2,\\ \text{s.t.}\quad &\mathbf{Q},\mathbf{P}\geq 0,\quad \mathbf{Q}^T\mathbf{1}_C=\mathbf{1}_N,\quad \mathbf{P}^T\mathbf{1}_C=\mathbf{1}_M,\end{aligned} \tag{9}$$

where $\alpha, \beta_a, \beta_b > 0$ are weighting parameters to balance different parts.

Through the unified model, we tightly connect the rating prediction model and the domain detection model together by the two regression regularized terms. On one hand, such a unified model ensures the collaborative effect among users varies across different domains. On the other hand, the domain-sensitive recommendation reduces the distances among the users within similar domains even they have no common rated items under the data sparsity condition.

4 Optimization

The joint optimization function in Eq. (9) is convex *w.r.t.* $\mathbf{U}$, $\mathbf{V}$, $\mathbf{Q}$, $\mathbf{P}$, $\mathbf{A}$ and $\mathbf{B}$ respectively, but not simultaneously. Thus, we iteratively optimize the above objective function by the following six alternate steps:

1) Keep $\mathbf{Q}$, $\mathbf{U}$ fixed, update $\mathbf{A}$;
2) Keep $\mathbf{P}$, $\mathbf{V}$ fixed, update $\mathbf{B}$;

3) Keep **V**, **Q** and **A** fixed, update **U**;
4) Keep **U**, **P** and **B** fixed, update **V**;
5) Keep **P**, **U** and **A** fixed, update **Q**;
6) Keep **Q**, **V** and **B** fixed, update **P**;

We summarize the updating algorithm in Algorithm 1.

Algorithm 1. Algorithm of DsRec

Input:
user-item rating matrix $\mathbf{R}$,
rank K, number of subgroup C
parameters λ_u, λ_v, α, β_a, β_b, η_q, η_r, γ_a and γ_b
Output:
latent factor representation $\mathbf{U}$, $\mathbf{V}$
1: Randomly initialize $\mathbf{U}$, $\mathbf{V}$;
2: Randomly initialize $\mathbf{Q}$, $\mathbf{P}$ with nonnegative;
3: Normalize $\mathbf{Q}$, $\mathbf{P}$;
4: **loop**
5: Fix $\mathbf{Q}$ and $\mathbf{U}$, update $\mathbf{A}$ as in Eq. (12);
6: Fix $\mathbf{P}$ and $\mathbf{V}$, update $\mathbf{B}$ as in Eq. (13);
7: Fix $\mathbf{V}$, $\mathbf{Q}$ and $\mathbf{A}$, compute derivative of $\mathbf{U}$ as in Eq. (14) and update $\mathbf{U}$ using gradient descent method;
8: Fix $\mathbf{U}$, $\mathbf{P}$ and $\mathbf{B}$, compute derivative of $\mathbf{V}$ as in Eq. (15) and update $\mathbf{V}$ using gradient descent method;
9: Fix $\mathbf{P}$, $\mathbf{U}$ and $\mathbf{A}$, update $\mathbf{Q}$ as in Eq. (20) and normalize $\mathbf{Q}$ so that $\mathbf{Q}^T\mathbf{1}_C = \mathbf{1}_N$;
10: Fix $\mathbf{Q}$, $\mathbf{V}$ and $\mathbf{B}$, update $\mathbf{P}$ as in Eq. (21) and normalize $\mathbf{P}$ so that $\mathbf{P}^T\mathbf{1}_C = \mathbf{1}_M$;
11: **end loop** until convergence

4.1 The Updating Rules for A and B

The derivatives of objective function *w.r.t.* **A** and **B** can be calculated as

$$\frac{\partial \mathcal{O}}{\partial \mathbf{A}} = \beta_a(\mathbf{AU} - \mathbf{Q})\mathbf{U}^T + \beta_a\gamma_a\mathbf{A}, \tag{10}$$

$$\frac{\partial \mathcal{O}}{\partial \mathbf{B}} = \beta_b(\mathbf{BV} - \mathbf{P})\mathbf{V}^T + \beta_b\gamma_b\mathbf{B}. \tag{11}$$

We set the derivatives to zero and obtain the analytical solutions as

$$\mathbf{A} = \mathbf{QU}^T(\mathbf{UU}^T + \gamma_a\mathbf{I})^{-1}, \tag{12}$$

$$\mathbf{B} = \mathbf{PV}^T(\mathbf{VV}^T + \gamma_b\mathbf{I})^{-1}. \tag{13}$$

4.2 The Updating Rules for U and V

Similar to [24], the local minimum of the objective function of **U** and **V** can be found with the gradient descent method. Thus, **U** and **V** are updated with the derivatives of objective function as follows:

$$\frac{\partial \mathcal{O}}{\partial \mathbf{U}_n} = \sum_{m=1}^{M} \delta_{nm}\big(\mathbf{U}_n^T\mathbf{V}_m - \mathbf{R}_{nm}\big)\mathbf{V}_m + \lambda_u\mathbf{U}_n + \beta_a(\mathbf{A}^T\mathbf{AU} - \mathbf{A}^T\mathbf{Q})_n, \tag{14}$$

$$\frac{\partial \mathcal{O}}{\partial \mathbf{V}_m} = \sum_{n=1}^{N} \delta_{nm}\big(\mathbf{U}_n^T\mathbf{V}_m - \mathbf{R}_{nm}\big)\mathbf{U}_n + \lambda_v\mathbf{V}_m + \beta_b(\mathbf{B}^T\mathbf{BV} - \mathbf{B}^T\mathbf{R})_m. \tag{15}$$

4.3 The Updating Rules for Q and P

Let Φ and Ψ be the Lagrange multiplier matrices for the nonnegative constraints of $\mathbf{Q} \geq 0$ and $\mathbf{P} \geq 0$, respectively. Keeping the parts of $\mathcal{O}$ which are related to corresponding variables, the Lagrange functions of **Q** and **P** are defined as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{Q}) = {} & \frac{\alpha}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\delta_{nm}\left(\left\|\frac{\mathbf{Q}_n}{\sqrt{\mathbf{D}_{nn}^{row}}} - \frac{\mathbf{P}_m}{\sqrt{\mathbf{D}_{mm}^{col}}}\right\|^2 \mathbf{R}_{nm}\right) \\ & + \frac{\alpha\eta_q}{2}\sum_{n=1}^{N}||\mathbf{Q}_n||_1 + \frac{\beta_a}{2}||\mathbf{AU} - \mathbf{Q}||_F^2 + Tr(\Phi^T\mathbf{Q}),\end{aligned} \tag{16}$$

$$\begin{aligned}\mathcal{L}(\mathbf{P}) = {} & \frac{\alpha}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\delta_{nm}\left(\left\|\frac{\mathbf{Q}_n}{\sqrt{\mathbf{D}_{nn}^{row}}} - \frac{\mathbf{P}_m}{\sqrt{\mathbf{D}_{mm}^{col}}}\right\|^2 \mathbf{R}_{nm}\right) \\ & + \frac{\alpha\eta_r}{2}\sum_{m=1}^{M}||\mathbf{P}_m||_1 + \frac{\beta_b}{2}||\mathbf{BV} - \mathbf{P}||_F^2 + Tr(\Psi^T\mathbf{P}),\end{aligned} \tag{17}$$

where $Tr(.)$ denotes the trace of matrix.

Generally, for the 1-norm sparsity function, the objective is not continuously differentiable and the most straightforward gradient-based methods are difficult to apply. However, in Eqs. (16) and (17), the nonnegative constraints ease the problem of 1-norm regularization. Requiring derivatives of $\mathcal{L}(\mathbf{Q})$ *w.r.t.* $\mathbf{Q}_{cn}$ and $\mathcal{L}(\mathbf{P})$ *w.r.t.* $\mathbf{P}_{cm}$, we have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{Q}_{cn}} = {} & \alpha\sum_{m=1}^{M}\delta_{nm}\frac{\mathbf{R}_{nm}}{\sqrt{\mathbf{D}_{nn}^{row}}}\left(\frac{\mathbf{Q}_{cn}}{\sqrt{\mathbf{D}_{nn}^{row}}} - \frac{\mathbf{P}_{cm}}{\sqrt{\mathbf{D}_{mm}^{col}}}\right) \\ & + \alpha\eta_q\mathbf{Q}_{cn} + \beta_a(\mathbf{Q} - \mathbf{AU})_{cn} + \Phi_{cn},\end{aligned} \tag{18}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{P}_{cm}} = {} & \alpha\sum_{n=1}^{N}\delta_{nm}\frac{\mathbf{R}_{nm}}{\sqrt{\mathbf{D}_{mm}^{col}}}\left(\frac{\mathbf{P}_{cm}}{\sqrt{\mathbf{D}_{mm}^{col}}} - \frac{\mathbf{Q}_{cn}}{\sqrt{\mathbf{D}_{nn}^{row}}}\right) \\ & + \alpha\eta_r\mathbf{P}_{cm} + \beta_b(\mathbf{P} - \mathbf{BV})_{cm} + \Psi_{cm}.\end{aligned} \tag{19}$$

Using the Karush-Kuhn-Tucker (KKT) conditions $\Phi_{cn}\mathbf{Q}_{cn} = 0$ and $\Psi_{cm}\mathbf{R}_{cm} = 0$, we have the following updating rules

$$\mathbf{Q}_{cn} \leftarrow \mathbf{Q}_{cn}\frac{\alpha\sum_{m=1}^{M}\delta_{nm}\frac{\mathbf{R}_{nm}}{\sqrt{\mathbf{D}_{nn}^{row}}}\frac{\mathbf{P}_{cm}}{\sqrt{\mathbf{D}_{mm}^{col}}} + \beta_a(\mathbf{AU})_{cn}^{+}}{(\alpha(1+\eta_q) + \beta_a)\mathbf{Q}_{cn} + \beta_a(\mathbf{AU})_{cn}^{-}}, \tag{20}$$

$$\mathbf{R}_{cm} \leftarrow \mathbf{R}_{cm}\frac{\alpha\sum_{n=1}^{N}\delta_{nm}\frac{\mathbf{R}_{nm}}{\sqrt{\mathbf{D}_{mm}^{col}}}\frac{\mathbf{Q}_{cn}}{\sqrt{\mathbf{D}_{nn}^{row}}} + \beta_b(\mathbf{BV})_{cm}^{+}}{(\alpha(1+\eta_r) + \beta_b)\mathbf{R}_{cm} + \beta_b(\mathbf{BV})_{cm}^{-}}, \tag{21}$$

where $\mathbf{Y}^{+} = \frac{\mathbf{Y}+|\mathbf{Y}|}{2} \geq 0$ and $\mathbf{Y}^{-} = \frac{\mathbf{Y}-|\mathbf{Y}|}{2} \geq 0$.

5 Experimental Analysis

In this section, we use real-world user-item rating data to empirically validate the effectiveness of our proposed model for rating prediction.

TABLE 1
Basis Statistics of the Datasets Movielens-100 K, Epinions, and Ciao

	Movielens-100 K	Epinions	Ciao
# of users	943	21,109	5,636
# of items	1,682	13,957	4,147
# of ratings	100,000	433,507	98,529
# of ratings per user	106.04	20.54	17.48
# of ratings per item	59.45	31.06	23.76
Rating Sparsity	93.70%	99.85%	99.58%

5.1 Datasets

We examine how DsRec behaves on the classic movie rating dataset Movielens-100 K[1] and two product review datasets: Epinions and Ciao.[2]

Movielens-100 K is collected through the MovieLens web site. It consists of 100,000 ratings (1-5) from 943 users on 1,682 movies and each user has rated at least 20 movies. Epinions and Ciao are well-known consumer opinion websites where users can assign their familiar products integer ratings from 1 to 5. The two datasets used in this study are published by the authors of [34] including data records until May 2011. Note that the original Epinions dataset consists of 22,166 users who have rated on 296,277 different items, while the original Ciao dataset consists of 7,375 users who have rated on 106,797 different items. To build a compact and informative dataset for model learning, we expect to maintain those active users and popular items in original dataset. Specifically, we first remove the users who rate less than 10 items and then remove the items which has less than 10 ratings by the users. Thus we obtain a Epinions subset and a Ciao subset whose detailed statistics are shown in Table 1. Compared to Movielens-100 K, these two datasets are much sparser.

5.2 Evaluation Measures

In the experimental evaluations, we use the Mean Absolute Error (MAE) to measure the prediction quality of our proposed model in comparison with other CF methods. The measure of MAE is defined as:

$$MAE = \frac{1}{T}\sum_{n,m} |\mathbf{R}_{nm} - \hat{\mathbf{R}}_{n,m}|, \tag{22}$$

where $\mathbf{R}_{nm}$ denotes the rating score which is given by the n-th user to the m-th item, $\hat{\mathbf{R}}_{n,m}$ denotes the predicted rating score corresponding to the n-th user and the m-th item, and T denotes the number of tested ratings. From the definition, we can see a smaller MAE value means a better performance.

5.3 Compared Methods

In order to show the effectiveness of our proposed recommendation approach, we compare the recommendation results of DsRec with the following methods:

- *WNMF* [22]: Weighted Nonnegative Matrix Factorization. We set the weight to 1 if the rating is observed, otherwise, it is set to 0. Thus, only observed ratings are used to train the model, and the missing ratings are predicted by the reconstructed matrix.
- *SVD* [21]: Singular Value Decomposition. We first remove sparsity by filling the missing data in the user-item rating matrix with the average rating of the corresponding item. And then we normalize the matrix by subtraction of the corresponding customer's average rating. The rating prediction is made by the matrix reconstruction.
- *PMF* [24]: Probabilistic Matrix Factorization, a widely used matrix factorization model for recommendation. The regularization parameters in the model is set by the grid {0.01, 0.05, 0.1, 0.5, 1, 5, 10}. DsRec ignoring the user-item subgroup analysis degrades into PMF.
- *UserClu*: this method groups users with k-means and performs basis CF method within the clusters. We adopt PMF as the basis CF method, and also try the regularization parameter as that in PMF experiments.
- *ItemClu*: this method groups items with k-means and performs basis CF method within the clusters just like UserClu.
- *CoClust* [12]: a collaborative filtering approach based on a weighted co-clustering algorithm. this method generates predictions based on the average ratings of the co-clusters (user-item neighborhoods) while taking into account the individual biases of the users and items.
- *MCoC* [13]: CF using Multiclass Co-Clustering. A staged framework to combine subgroups analysis with basis CF models. We also adopt PMF as the basis CF method just like UserClu.

In the comparison, the first three methods are traditional and popular collaborative recommendation techniques without considering the domain influence, and the other four methods are clustering based collaborative recommendation techniques, in which UserClu and ItemClu are one-sided clustering based solution for domain detection, and CoClust [12] and MCoC [13] are two-sided clustering based ones. As stated above, our proposed DsRec is a two-sided clustering method, and the main difference from the compared work is that the domain detection and rating prediction are unified into the same framework, instead of performing the both tasks in a divide-and-conquer style. We will present their performance and comprehensively analyze the results to demonstrate the effectiveness of our method as follows.

5.4 Experimental Results

For each dataset, we use different observed data divisions (20, 50 and 80 percent) in our experiments. Training data 80 percent, for example, means we randomly select 80 percent of observed ratings from user-item rating matrix as the training data to predict the remaining 20 percent ratings. We also set different latent factor dimension (K) to test the matrix factorization methods. 10 random divisions of

1. http://grouplens.org/datasets/movielens/
2. http://www.public.asu.edu/ jtang20/datasetcode/truststudy.htm

TABLE 2
MAE Comparison on Movielens-100 K ($\alpha = 100$, $\beta = 100$, and $C = 10$)

		WNMF	SVD	PMF	UserClu	ItemClu	CoClust	MCoC	DsRec
20%	K=5	0.8366 ± 0.0022	0.8057 ± 0.0015	0.8261 ± 0.0028	0.8525 ± 0.0025	0.8624 ± 0.0021	0.8020 ± 0.0035	0.8029 ± 0.0033	**0.7592 ± 0.0013**
	K=10	0.8348 ± 0.0009	0.8058 ± 0.0012	0.8257 ± 0.0031	0.8742 ± 0.0027	0.8620 ± 0.0022		0.8159 ± 0.0015	**0.7670 ± 0.0011**
	K=15	0.8368 ± 0.0022	0.8063 ± 0.0021	0.8234 ± 0.0024	0.8671 ± 0.0017	0.8580 ± 0.0013		0.8180 ± 0.0022	**0.7721 ± 0.0006**
	K=20	0.8337 ± 0.0015	0.8069 ± 0.0012	0.8238 ± 0.0008	0.8694 ± 0.0032	0.8545 ± 0.0011		0.8160 ± 0.0019	**0.7714 ± 0.0013**
50%	K=5	0.7955 ± 0.0010	0.7691 ± 0.0023	0.7748 ± 0.0018	0.8075 ± 0.0011	0.7906 ± 0.0021	0.7670 ± 0.0016	0.7560 ± 0.0031	**0.7277 ± 0.0014**
	K=10	0.7933 ± 0.0012	0.7663 ± 0.0011	0.7703 ± 0.0022	0.8335 ± 0.0023	0.8216 ± 0.0018		0.7563 ± 0.0022	**0.7349 ± 0.0017**
	K=15	0.7947 ± 0.0026	0.7667 ± 0.0012	0.7721 ± 0.0023	0.8249 ± 0.0029	0.8147 ± 0.0032		0.7585 ± 0.0012	**0.7394 ± 0.0015**
	K=20	0.7955 ± 0.0020	0.7686 ± 0.0017	0.7720 ± 0.0029	0.8146 ± 0.0021	0.8126 ± 0.0010		0.7513 ± 0.0032	**0.7409 ± 0.0015**
80%	K=5	0.7790 ± 0.0012	0.7485 ± 0.0010	0.7348 ± 0.0023	0.7682 ± 0.0017	0.7548 ± 0.0011	0.7560 ± 0.0023	0.7213 ± 0.0011	**0.7060 ± 0.0014**
	K=10	0.7784 ± 0.0013	0.7407 ± 0.0023	0.7343 ± 0.0019	0.7834 ± 0.0027	0.7777 ± 0.0020		0.7337 ± 0.0033	**0.7108 ± 0.0021**
	K=15	0.7772 ± 0.0016	0.7408 ± 0.0031	0.7358 ± 0.0015	0.7847 ± 0.0012	0.7694 ± 0.0028		0.7229 ± 0.0019	**0.7134 ± 0.0023**
	K=20	0.7747 ± 0.0022	0.7430 ± 0.0022	0.7370 ± 0.0022	0.7869 ± 0.0022	0.7734 ± 0.0022		0.7243 ± 0.0022	**0.7162 ± 0.0018**

observed ratings are carried independently, and the average results are reported.

In all the experiments of DsRec conducted in this paper, we set the regularization parameters $\lambda_u = \lambda_v = 0.1$, $\eta_q = \eta_p = 1$ and $\gamma_a = \gamma_b = 10$ empirically. We also set the weighting coefficient $\beta_a = \beta_b = \beta$. For those clustering CF methods (UserClu, ItemClu, CoClust, MCoC and DsRec), the number of subgroups (C) is set to 10 in all the experiments. The performance sensitivity in terms of the key parameters, including α, β, and C, will be discussed in Section 5.5.

Table 2 summarizes the performance of different methods on Movielens-100 K, where the parameters α and β are set to 100 and 100, respectively. The experimental results on Epinions are shown in Table 3, where the parameters α and β are sets to 10 and 1,000, respectively. Table 4 shows the experimental results on Ciao, where the parameters α and β are sets to 100 and 1,000 respectively. These experiments reveal a number of interesting points:

- Our method DsRec consistently outperforms other approaches in all the settings on these three datasets. This verifies the effectiveness of our method.
- The performance of CoClust, MCoC and DsRec outperform that of WNMF, SVD, and PMF. This illustrates that domain detection can improve the rating prediction accuracy. The performance of UserClu and ItemClu are not satisfactory. The reason may be that the user-item rating matrix is too sparse, which results in clustering by simple k-means cannot discovers meaningful domains.
- As a kind of two-sided clustering solution, MCoC and DsRec achieve better performance than those one-sided clustering solutions, e.g., UserClu, ItemClu, and CoClust. The reason is that the two-sided clustering can make full use of the intra- and inter-correlations of users and items in the rating matrix.
- DsRec consistently achieves better results than MCoC. Different from the staged style of MCoC, DsRec integrates domain detection and rating prediction into a unified framework. Therefore, on one hand, the train data could be fruitfully exploited. On the other hand, as mentioned above, DsRec could alleviate the data sparsity problem better than the staged style methods.
- Different from other matrix factorization methods, there is a significant character for WNMF which is more sensitive with the change of the latent factor dimension. In certain range, with the increasing of parameter K, the performance of WNMF is improved markedly.
- A surprising finding is that when the user-item rating matrix is very sparse (20 percent), the performance of rating prediction by average score is not so poor as we might expect, except that the performance of WNMF is disappointing. The reason may be that WNMF could not capture the meaningful basis components when the matrix is too sparse,

TABLE 3
MAE Comparison on Epinions ($\alpha = 10$, $\beta = 1{,}000$, and $C = 10$)

		WNMF	SVD	PMF	UserClu	ItemClu	CoClust	MCoC	DsRec
20%	K=5	1.1007 ± 0.0031	0.9310 ± 0.0033	0.9405 ± 0.0021	0.9773 ± 0.0026	0.9864 ± 0.0014	1.0754 ± 0.0026	0.9396 ± 0.0017	**0.8707 ± 0.0024**
	K=10	1.0702 ± 0.0024	0.9312 ± 0.0025	0.9464 ± 0.0010	0.9723 ± 0.0019	0.9741 ± 0.0017		0.9389 ± 0.0037	**0.8675 ± 0.0021**
	K=15	1.0538 ± 0.0015	0.9314 ± 0.0023	0.9465 ± 0.0018	0.9701 ± 0.0033	0.9740 ± 0.0019		0.9345 ± 0.0017	**0.8636 ± 0.0017**
	K=20	1.0447 ± 0.0015	0.9313 ± 0.0019	0.9414 ± 0.0009	0.9660 ± 0.0034	0.9692 ± 0.0024		0.9356 ± 0.0022	**0.8587 ± 0.0023**
50%	K=5	0.9258 ± 0.0017	0.8647 ± 0.0027	0.8796 ± 0.0031	0.9140 ± 0.0026	0.9135 ± 0.0023	0.8924 ± 0.0015	0.8622 ± 0.0013	**0.8184 ± 0.0025**
	K=10	0.9141 ± 0.0020	0.8648 ± 0.0015	0.8794 ± 0.0016	0.9083 ± 0.0025	0.9073 ± 0.0031		0.8651 ± 0.0021	**0.8234 ± 0.0011**
	K=15	0.8947 ± 0.0021	0.8649 ± 0.0017	0.8748 ± 0.0032	0.8973 ± 0.0031	0.9175 ± 0.0018		0.8680 ± 0.0022	**0.8248 ± 0.0017**
	K=20	0.8835 ± 0.0020	0.8651 ± 0.0016	0.8715 ± 0.0027	0.8769 ± 0.0028	0.8946 ± 0.0023		0.8643 ± 0.0019	**0.8283 ± 0.0026**
80%	K=5	0.8795 ± 0.0018	0.8571 ± 0.0021	0.8562 ± 0.0022	0.8561 ± 0.0016	0.8743 ± 0.0034	0.8450 ± 0.0019	0.8420 ± 0.0009	**0.7979 ± 0.0017**
	K=10	0.8727 ± 0.0021	0.8572 ± 0.0024	0.8598 ± 0.0017	0.8862 ± 0.0022	0.8556 ± 0.0020		0.8465 ± 0.0026	**0.8052 ± 0.0013**
	K=15	0.8696 ± 0.0016	0.8580 ± 0.0021	0.8546 ± 0.0019	0.8520 ± 0.0013	0.8674 ± 0.0023		0.8467 ± 0.0031	**0.8103 ± 0.0028**
	K=20	0.8525 ± 0.0022	0.8582 ± 0.0020	0.8556 ± 0.0014	0.8430 ± 0.0024	0.8703 ± 0.0018		0.8452 ± 0.0035	**0.8120 ± 0.0024**

TABLE 4
MAE Comparison on Ciao ($\alpha = 100$, $\beta = 1{,}000$, and $C = 10$)

		WNMF	SVD	PMF	UserClu	ItemClu	CoClust	MCoC	DsRec
20%	K=5	1.1241 ± 0.0033	0.9050 ± 0.0030	0.8780 ± 0.0021	0.9120 ± 0.0022	0.9118 ± 0.0035	1.0283 ± 0.0032	0.8637 ± 0.0029	$\mathbf{0.8423 \pm 0.0027}$
	K=10	1.0970 ± 0.0028	0.9070 ± 0.0036	0.8782 ± 0.0031	0.8927 ± 0.0025	0.8912 ± 0.0020		0.8644 ± 0.0031	$\mathbf{0.8193 \pm 0.0028}$
	K=15	1.0813 ± 0.0024	0.9084 ± 0.0031	0.8777 ± 0.0033	0.8871 ± 0.0025	0.8851 ± 0.0029		0.8659 ± 0.0030	$\mathbf{0.8108 \pm 0.0027}$
	K=20	1.0667 ± 0.0028	0.9092 ± 0.0021	0.8774 ± 0.0015	0.8830 ± 0.0036	0.8850 ± 0.0032		0.8638 ± 0.0031	$\mathbf{0.8142 \pm 0.0023}$
50%	K=5	0.8991 ± 0.0033	0.8034 ± 0.0025	0.8420 ± 0.0029	0.8730 ± 0.0030	0.8548 ± 0.0027	0.7928 ± 0.0025	0.8350 ± 0.0024	$\mathbf{0.7738 \pm 0.0026}$
	K=10	0.8731 ± 0.0026	0.8071 ± 0.0032	0.8402 ± 0.0031	0.8744 ± 0.0019	0.8341 ± 0.0031		0.8348 ± 0.0037	$\mathbf{0.7692 \pm 0.0029}$
	K=15	0.8573 ± 0.0028	0.8097 ± 0.0027	0.8377 ± 0.0031	0.8452 ± 0.0025	0.8372 ± 0.0025		0.8235 ± 0.0029	$\mathbf{0.7716 \pm 0.0033}$
	K=20	0.8466 ± 0.0029	0.8112 ± 0.0033	0.8355 ± 0.0036	0.8537 ± 0.0032	0.8351 ± 0.0027		0.8238 ± 0.0023	$\mathbf{0.7650 \pm 0.0020}$
80%	K=5	0.8289 ± 0.0027	0.7895 ± 0.0026	0.8054 ± 0.0023	0.8249 ± 0.0028	0.8107 ± 0.0020	0.7556 ± 0.0031	0.7823 ± 0.0033	$\mathbf{0.7467 \pm 0.0021}$
	K=10	0.8187 ± 0.0021	0.7935 ± 0.0031	0.8067 ± 0.0028	0.8383 ± 0.0027	0.8204 ± 0.0025		0.7851 ± 0.0030	$\mathbf{0.7416 \pm 0.0025}$
	K=15	0.8026 ± 0.0031	0.7950 ± 0.0033	0.8077 ± 0.0020	0.8409 ± 0.0037	0.8168 ± 0.0031		0.7789 ± 0.0035	$\mathbf{0.7484 \pm 0.0032}$
	K=20	0.7974 ± 0.0033	0.7940 ± 0.0028	0.8048 ± 0.0032	0.8365 ± 0.0034	0.8157 ± 0.0028		0.7834 ± 0.0031	$\mathbf{0.7482 \pm 0.0029}$

while other methods have a strong ability to mine useful information from the observed ratings.

To further evaluate the effectiveness of various approaches, we perform the two-sample t-test to analyze their statistical significance, for which only the best three methods (i.e., CoCluster, MCoC, and our proposed DsRec) are considered in the evaluation. Specifically, 20 random divisions of observed ratings are carried independently for the above three approaches, and we perform the two-sample t-tests for each pair of approaches, in which the null hypothesis is stated as that the two samples have the same means. Table 5 gives the p-value comparison in t-test. From the results, we can clearly see the superiority of our solution, since the p-values are much less than the significance level (0.05).

Besides, we compute the average rating scores within each subgroup (*i.e.*, the average rating scores of the rating submatrix which corresponds to the subgroup), to show the effectiveness of our domain detection model. If the confidence value $\mathbf{Q}_{cn}(\mathbf{R}_{cm}) > \frac{1}{C}$, the user (item) is considered to belong to the c-th subgroup. The average rating scores, indicated by the blue points, are illustrated in Fig. 2. Level lines indicate the average rating scores of the total dataset (i.e., the average rating scores of the rating matrix). All these blue points are above the level line over all these three datasets, which is in line with what we expected. Because DsRes just attempts to group the users and items into the same subgroup if the users rate the items with high scores. Thus, the domain detection model is effective.

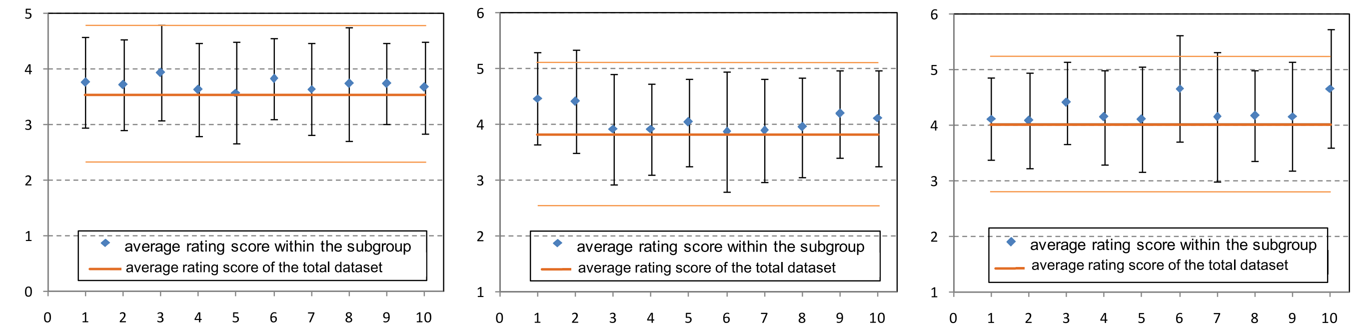
5.5 Impact of Parameters

In this section, we mainly discuss the impact of parameters in the unified model. As stated above, we set the regularization parameters $\lambda_u = \lambda_v = 0.1$, $\eta_q = \eta_p = 1$ and $\gamma_a = \gamma_b = 10$ as constants empirically during our experiments. Observed from Table 2, 3, 4, DsRec is not so sensitive to the variation of K, i.e., the latent factor dimension. In the following experiments, we set $K = 10$, and use 80 percent observed data division as training data. Beyond all that, the main parameters, i.e., α, β and C, remain for discussion in the following. The three parameters are closely related to the roles of rating prediction and domain detection in the unified framework. Specifically, α is the weighting coefficient of user-item subgroup detection model, β $(\beta = \beta_a = \beta_b)$ is the weighting coefficient of regression regularization items, and C is the number of subgroups. Figs. 3, 4 and 5 illustrate the impact of these parameters on three data sets, respectively.

Fig. 3 shows the variation of performance with respect to α. The larger α means the role of domain detection is more important. From the results, we find the performance variation is not very remarkable with respect to the parameter, but we can still find the best choice of α are 10 for Epinions and 100 for both Movielens-100 K and Ciao. From Fig. 4, it can be observed that DsRec is very sensitive to the variation

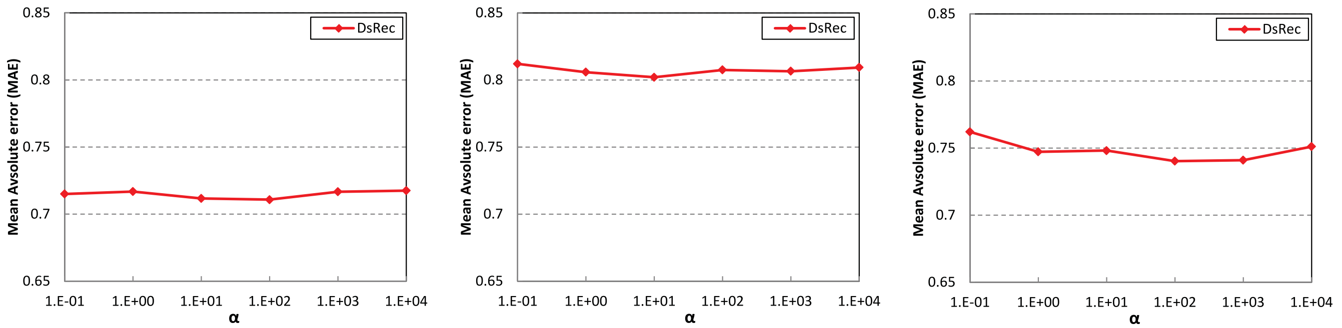
TABLE 5
Statistical Significance Analysis with P-Values Comparison in t-Test

		Movielens-100 K ($\alpha = 100, \beta = 100, C = 10$)		Epinions ($\alpha = 10, \beta = 1{,}000, C = 10$)		Ciao ($\alpha = 100, \beta = 1{,}000, C = 10$)	
		DsRec versus MCoC	DsRec versus CoClust	DsRec versus MCoC	DsRec versus CoClust	DsRec versus MCoC	DsRec versus CoClust
20%	K=5	1.37E-39	3.41E-40	2.19E-48	1.88E-63	9.75E-24	2.58E-58
	K=10	1.35E-48	7.10E-28	1.78E-45	1.59E-68	4.92E-35	3.73E-63
	K=15	2.77E-45	1.38E-37	1.11E-51	8.50E-67	7.09E-39	3.03E-63
	K=20	2.46E-46	5.04E-30	1.74E-51	4.08E-63	5.13E-38	4.29E-62
50%	K=5	2.14E-31	1.40E-42	1.52E-40	1.36E-49	7.59E-42	2.19E-20
	K=10	9.42E-28	2.42E-36	2.05E-46	7.19E-58	7.49E-39	3.10E-26
	K=15	1.48E-36	4.42E-38	3.25E-42	9.06E-58	1.17E-38	1.75E-22
	K=20	2.31E-15	1.46E-37	3.41E-34	7.16E-45	1.76E-47	3.99E-31
80%	K=5	1.61E-32	5.18E-46	2.87E-44	2.45E-42	4.71E-32	2.01E-12
	K=10	7.99E-27	6.82E-40	2.54E-42	2.63E-41	2.69E-37	3.11E-18
	K=15	3.44E-18	2.82E-38	4.97E-32	4.35E-37	9.99E-28	6.43E-09
	K=20	5.78E-15	1.31E-39	1.44E-34	1.24E-38	2.99E-32	1.95E-08



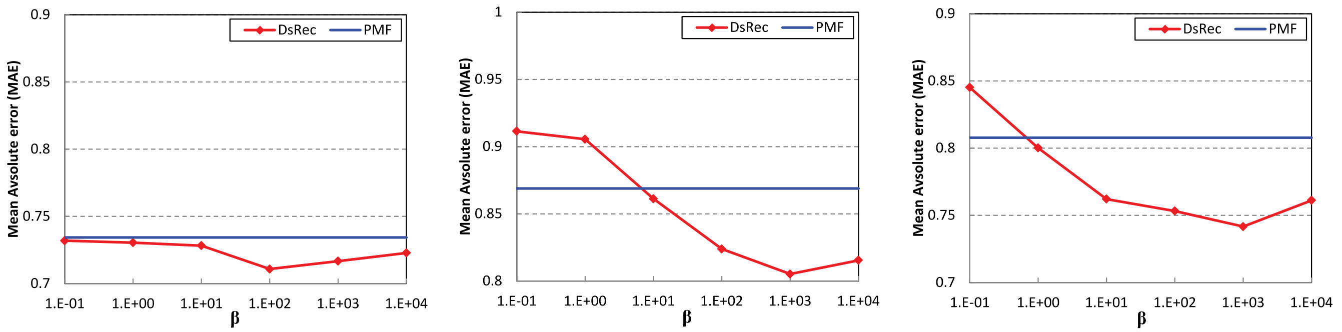
(a) Movielens-100K (b) Epinions (c) Ciao

Fig. 2. The average rating scores and the corresponding standard variances within different subgroups when $C = 10$. If the confidence value $\mathbf{Q}_{cn}(\mathbf{R}_{cm}) > \frac{1}{C}$, we think the user (item) belongs to the cth subgroup.



(a) Movielens-100K (b) Epinions (c) Ciao

Fig. 3. Impact of the parameter α. It is a weighting coefficient of user-item subgroup analysis model. For all these experiments, we set $C = 10$, and $\beta = 100, 1{,}000, 1{,}000$ for Movielens-100 K, Epinions, and Ciao, respectively.

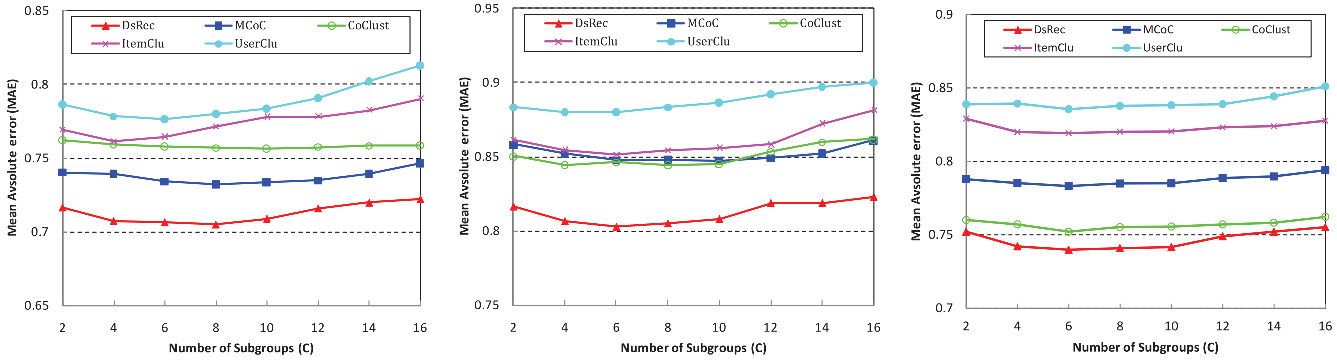


(a) Movielens-100K (b) Epinions (c) Ciao

Fig. 4. Impact of the parameter $\beta = \beta_a = \beta_b$. It is a weighting coefficient of regression regularization items. For all these experiments, we set $C = 10$, and $\alpha = 100, 10, 100$ for Movielens-100 K, Epinions, and Ciao, respectively. The performance of PMF [24] is used as a baseline, since DsRec will be degenerated to be an approximate PMF model, when a too small value of β is set.

of β, and a too small or too large value is not a good selection for the parameter. The parameter controls how much impact is functioned between the rating predication model and the domain detection model. Extremely, only importing the regression regularization terms, the rating prediction model and the domain detection model can be truly connected to perform the domain-sensitive recommendation. That is, when a too small value of β, DsRec will be degenarated to be an approximate PMF model, and the performance will become comparable with or even worse than the typical PMF [24], whose results are also presented in Fig. 4. A too larger value of β maybe make the learned latent factors focus more on the discriminative ability on different domains, while the ability of observed rating reconstruction is neglected. From this view, the insensitivity of the parameter α is understandable, since it cannot directly work on the rating prediction. For Movielens-100 K, the best performance is achieved when $\beta = 100$, while for Epinions and Ciao, β is set to 1,000.

As for C, the number of subgroups, we explore the other four related methods (i.e., MCoC [13], CoClust [12], ItemClu, and UserClu) to compare their sensitivity to the parameter. From the comparison in Fig. 5, we observe that dividing users or items, or both into some subgroups is necessary, but too small or too large number of subgroups brings the poor performance. It is easy to understand: a user usually tastes on a few domains, but not dispersive over all the item set. Clustering users and items into subgroups actually do the work of collecting correlated items and users into different clusters. Thus, a too small number



(a) Movielens-100K (b) Epinions (c) Ciao

Fig. 5. Impact of the parameter C. It is the number of subgroups. We set α to 100, 10, and 100, and β to 100, 1,000, and 1,000 for Movielens-100 K, Epinions, and Ciao, respectively. The performance of the other clustering based methods (i.e., MCoC [13], CoClust [12], ItemClu, and UserClu), are also explored to present an overall comparison on the variation of cluster numbers for domain-sensitive recommendation.

of subgroups cannot clearly partition different user-item interest domains, while a too larger number of subgroups make the preferences of the obtained clusters be shared together, and the latent factors of users and items cannot be discriminative enough for effective rating prediction. We set $C = 10$ for all the three dataset, since the value can ensure a relatively stable and better performance in the comparison.

6 Conclusions

In this paper, we develop a novel Domain-sensitive Recommendation algorithm, which makes rating prediction assisted with the user-item subgroup analysis. DsRec is a unified formulation integrating a matrix factorization model for rating prediction and a bi-clustering model for domain detection. Additionally, information between these two components are exchanged through two regression regularization items, so that the domain information guides the exploration of the latent space. Systematic experiments conducted on three real-world datasets demonstrate the effectiveness of our methods. It is worth noting that our method is totally based on the user-item rating matrix. In the future, we will attempt to explore both user-item interaction information and some external information simultaneously for domain detection.

Acknowledgments

This work was supported by 973 Program (2012CB316304), and National Natural Science Foundation of China (61332016, 61272329, and 61472422). Zechao Li is the corresponding author.

References

[1] Y. Zhang, B. Cao, and D.-Y. Yeung, "Multi-domain collaborative filtering," in *Proc. 26th Conf. Annu. Conf. Uncertainty Artif. Intell.*, 2010, pp. 725–732.
[2] X. Zhang, J. Cheng, T. Yuan, B. Niu, and H. Lu, "TopRec: Domain-specific recommendation through community topic mining in social network," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 1501–1510.
[3] Y. Jiang, J. Liu, X. Zhang, Z. Li, and H. Lu, "TCRec: Product recommendation via exploiting social-trust network and product category information," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 233–234.
[4] S. Han, S. Chee, J. Han, and K. Wang, "RecTree: An efficient collaborative filtering method," in *Proc. 3rd Int. Conf. Data Warehousing Knowl. Discovery*, 2001, pp. 141–151.
[5] B. M. Sarwar, J. Konstan, and J. Riedl, "Recommender systems for large-scale e-commerce: scalable neighborhood formation using clustering," in *Proc. 5th Int. Conf. Comput. Inf. Technol.*, 2002, pp. 1–6.
[6] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing," in *Proc. Res. Develop. Inf. Retrieval*, 2005, pp. 114–121.
[7] A. Bellogin and J. Parapar, "Using graph partitioning techniques for neighbour selection in user-based collaborative filtering," in *Proc. 6th ACM Conf. Recommender Syst.*, 2012, pp. 213–216.
[8] M. O'Connor and J. Herlocker, "Clustering items for collaborative filtering," in *Proc. ACM SIGIR Workshop Recommender Syst.*, 1999, pp. 1–4.
[9] L. Ungar, D. Foster, E. Andre, S. Wars, F. S. Wars, D. S. Wars, and J. H. Whispers, "Clustering methods for collaborative filtering," in *Proc. Workshop Recommendation Syst.*, 1998, pp. 114–129.
[10] L. Si and R. Jin, "Flexible mixture model for collaborative filtering," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 704–711.
[11] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha, "A generalized maximum entropy approach to Bregman co-clustering and matrix approximation," in *Proc. 10th ACM KDD*, 2004, pp. 509–514.
[12] T. George and S. Merugu, "A scalable collaborative filtering framework based on co-clustering," in *Proc. 5th IEEE Int. Conf. Data Mining*, 2005, pp. 625–628.
[13] B. Xu, J. Bu, C. Chen, and D. Cai, "An exploration of improving collaborative recommender systems via user-item subgroups," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 21–30.
[14] J. S. Breese, D. Heckerman, and C. M. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. 14th Conf. Uncertainty Artif. Intell.*, 1998, pp. 43–52.
[15] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1999, pp. 230–237.
[16] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. World Wide Web Conf.*, 2001, pp. 285–295.
[17] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–20, 2009.
[18] K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple Bayesian classifier," in *Proc. 6th Pacific Rim Int. Conf. Artif. Intell.*, 2000, pp. 679–689.
[19] S. Vucetic and Z. Obradovic, "Collaborative filtering using a regression-based approach," *Knowl. Inf. Syst.*, vol. 7, no. 1, pp. 1–22, 2005.
[20] T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in *Proc. 6th Int. Joint Conf. Artif. Intell.*, 1999, pp. 688–693.
[21] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender system – a case study," in *Proc. ACM WebKDD Workshop*, 2000, pp. 1–12.
[22] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *Proc. 6th SIAM Int. Conf. Data Mining*, 2006, pp. 549–553.

[23] N. Srebro, J. D. M. Rennie, and T. S. Jaakola, "Maximum-margin matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1329–1336.

[24] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.

[25] J. Liu, Y. Jiang, Z. Li, Z. Zhou, and H. Lu, "Partially shared latent factor learning with multiview data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1233–1246, Jun. 2015.

[26] Z. Li, J. Liu, J. Tang, and H. Lu, "Robust structured subspace learning for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2085–2098, Oct. 2015.

[27] M. C. Pham, Y. Cao, R. Klamma, and M. Jarke, "A clustering approach for collaborative filtering recommendation using social network analysis," *J. Universal Comput. Sci.*, vol. 17, no. 4, pp. 583–604, 2011.

[28] I. Esslimani, A. Brun, and A. Boyer, "A collaborative filtering approach combining clustering and navigational based correlations," in *Proc. 5th Int. Conf. Web Inf. Syst. Technol.*, 2009, pp. 364–369.

[29] C. Chen, X. Zheng, Y. wang, F. Hong, and Z. Lin, "Context-aware collaborative topic regression with social matrix factorization for recommender systems," in *Proc Conf. Artif. Intell.*, 2014, pp. 9–15.

[30] M. Huang, L. Sun, and W. Du, "Collaborative filtering recommendation algorithm based on item attributes," in *Proc. IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput.*, 2014, pp. 1–6.

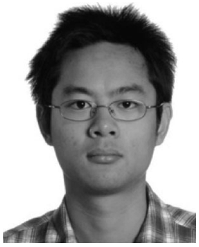
[31] M. Nilashi, D. Jannach, O. bin Ibrahim, and N. Ithnin, *Inf. Sci.*, vol. 293, no. 2, pp. 235–250, 2015.

[32] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2001, pp. 269–274.

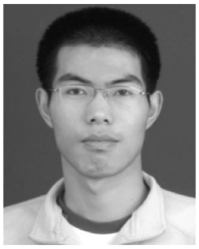
[33] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 1, no. 1, pp. 24–45, Jan. 2004.

[34] J. Tang, H. Gao, H. Liu, and A. D. Sarma, "eTrust: Understanding trust evolution in an online world," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 253–261.

Jing Liu received the BE degree in 2001 and the ME degree in 2004 from the Shandong University, and the PhD degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, in 2008. She is currently an associate professor in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. Her research interests include machine learning, image content analysis and classification, multimedia information indexing and retrieval, etc. She is a member of the IEEE.



Yu Jiang received the BE degree in electrical engineering from the Beihang University, in 2009 and the PhD degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, in 2014. His research interests includes subspace learning, multiview learning and recommender system, etc.



Zechao Li received the BE degree from the University of Science and Technology of China, in 2008 and the PhD degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, in 2013. He is currently an assistant professor in the School of Computer Science, Nanjing University of Science and Technology. His research interests include machine learning, subspace learning, multimedia understanding, etc. He is a member of the IEEE.

Xi Zhang received the BE degree from the Sichuan University, in 2008 and the ME degree from the National Space Science Center, Chinese Academy of Sciences, in 2011, and the PhD degree from the National Laboratory of Pattern Recognition, Institute of Automation, in 2015. She is currently a researcher at the Noahs Ark Lab of Huawei. Her research interests include recommender systems, text mining, and statistical machine learning.

Hanqing Lu received the BE degree in 1982 and the ME degree in 1985 from the Harbin Institute of Technology, and received the PhD degree from the Huazhong University of Sciences and Technology, in 1992. He is currently a professor at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, object tracking, recognition and image retrieval, etc. He is a senior member of the IEEE.