DCT

Cold-Start Item and User Recommendation with Decoupled Completion and Transduction

D C T

Iman Barjasteh
Electrical & Computer Eng.
Michigan State University
barjaste@msu.edu

Rana Forsati
Computer Science & Eng.
Michigan State University
forsati@cse.msu.edu

Farzan Masrour
Statistics & Probability
Michigan State University
masrour@msu.edu

Abdol-Hossein Esfahanian
Computer Science & Eng.
Michigan State University
esfahanian@cse.msu.edu

Hayder Radha
Electrical & Computer Eng.
Michigan State University
radha@egr.msu.edu

ABSTRACT

A major challenge in collaborative filtering based recommender systems is how to provide recommendations when rating data is sparse or entirely missing for a subset of users or items, commonly known as the *cold-start* problem. In recent years, there has been considerable interest in developing new solutions that address the cold-start problem. These solutions are mainly based on the idea of exploiting other sources of information to compensate for the lack of rating data. In this paper, we propose a novel algorithmic framework based on matrix factorization that simultaneously exploits the similarity information among users and items to alleviate the cold-start problem. In contrast to existing methods, the proposed algorithm *decouples* the following two aspects of the cold-start problem: (a) the completion of a rating sub-matrix, which is generated by excluding cold-start users and items from the original rating matrix; and (b) the transduction of knowledge from existing ratings to cold-start items/users using side information. This crucial difference significantly boosts the performance when appropriate side information is incorporated. We provide theoretical guarantees on the estimation error of the proposed two-stage algorithm based on the richness of similarity information in capturing the rating data. To the best of our knowledge, this is the first algorithm that addresses the cold-start problem with provable guarantees. We also conduct thorough experiments on synthetic and real datasets that demonstrate the effectiveness of the proposed algorithm and highlights the usefulness of auxiliary information in dealing with both cold-start users and items.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Miscellaneous

Keywords

Matrix factorization; Sparsity; Cold-start Problem

1. INTRODUCTION

The ever-growing popularity of e-commerce sites and exponential growth of online users and huge collections of items have created a compelling demand for efficient recommender systems to guide users toward items (e.g., products, books, movies, etc). In the past decade, there has been a vast amount of research in this field, mostly focusing on designing novel algorithms for generating high quality suggestions (see e.g. [1], for a more recent survey). For instance, Collaborative filtering (CF) aims to predict preferences of a given "user" for some "objects" based on her previously revealed preferences– typically in the form of purchases or ratings– as well as the revealed preferences of other users.

Despite significant improvements, recommendation systems suffer from a few inherent limitations and weaknesses that need to be addressed and have captured the attention of researchers. Among such limitations are *data sparsity* and *cold-start* problems. For instance, for many applications the rating data is usually very sparse or there is no available rating for a subset of users or items (known as cold-start user and cold-start item, respectively)– making standard collaborative filtering algorithms infeasible. Therefore, it is of significant interest to develop algorithms that alleviate the data sparsity challenge and solve the cold-start problem.

Due in part to its importance, there has been an active line of work to address difficulties associated with cold-start users and items, and several researchers have considered a variety of techniques [16, 22, 21, 24, 27, 8, 26, 28, 14, 17], where a common theme among them is to exploit auxiliary information about users or items besides the rating data that are usually available. The main motivation behind these techniques stems from the following observation: other sources of data could potentially provide useful information about the underlying model and complement the rating data. For instance, users' decisions on item ratings are very likely to be influenced by other users with whom they have social connections (friends, families, etc). As a result, the availability of auxiliary information such as users' profile information [2], social context (trust and distrust relations) of users [4], information embedded in the review text [8], and features of items [5] would definitely benefit the recommendations. Therefore, an interesting research question, which is the main focus of this paper is: *how to effectively incorporate the side information about users and items to overcome the cold-start problem*.

The aim of this paper is to propose an efficient matrix factorization algorithm with similarity information to answer the above question. In fact, we study a somewhat more general problem where both cold-start users and cold-

RecSys'15, September 16–20, 2015, Vienna, Austria.
© 2015 ACM. ISBN 978-1-4503-3692-5/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2792838.2800196.

22: LCE

start items are present simultaneously and address these two challenges in combination. The dominant paradigm in existing methods such as shared subspace learning [22] and kernelized matrix factorization [28] is to augment the existing methods with side information and perform (a) the *completion* of rating matrix and (b) the *transduction* of knowledge from existing ratings to cold-start items/users simultaneously. While this recipe can generate good results in practice, it has a notable drawback: these methods propagate the completion and transduction errors repetitively and in an uncontrolled way. Therefore, we propose a two-stage algorithm that *decouples* completion from transduction. In particular, in the first step we exclude the cold-start users and items from the rating matrix and recover the obtained sub-matrix in a perfect way. Then, the recovered sub-matrix along with available similarity side information about users and items will be utilized to transduct the knowledge to cold-start users/items. We provide theoretical performance guarantees on the estimation error of the proposed algorithm while most of the existing methods do not provide any theoretical support. We also conduct a set of through experiments to demonstrate the merits of the proposed algorithm and complement our theoretical results. Finally, we compare our algorithm with several other specialized algorithms for cold-start user and item recommendation to demonstrates the superiority of our proposed framework over existing methods.

Outline. The remainder of the paper is organized as follows. In Section 2, we begin by establishing notation and providing background on matrix factorization, noting a few of its deficiencies for our setting. We then in Section 3 describe the proposed algorithm and state our main theorem on its estimation error. The experimental results are provided in Section 4 and concluding remarks are given in Section 5.

2. THE SETTING

Before proceeding to the proposed algorithm, in this section we establish the notations which are used throughout the paper and formally describe our problem setting.

The set of non-negative real numbers is denoted by $\mathbb{R}_+$. We use $[n]$ to denote a set on integers $\{1, 2, \cdots, n\}$. Scalars are denoted by lower case letters and vectors by bold face lower case letters such as $\mathbf{u}$. We use bold face upper case letters such as $\mathbf{M}$ to denote matrices. The Frobenius and spectral norms of a matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$ is denoted by $\|\mathbf{M}\|_F$, i.e, $\|\mathbf{M}\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} |\mathbf{M}_{ij}|^2}$ and $\|\mathbf{M}\|_2$, respectively. The transpose of a vector and a matrix denoted by $\mathbf{m}^\top$ and $\mathbf{M}^\top$, respectively. We use $(\mathbf{M})^\dagger$ to denote the Moore-Penrose pseudo inverse of matrix $\mathbf{M}$. The dot product between two vectors $\mathbf{m}$ and $\mathbf{n}$ is denoted by $\mathbf{m}^\top \mathbf{n}$.

In collaborative filtering we assume that there is a set of n users $\mathcal{U} = \{u_1, \cdots, u_n\}$ and a set of m items $\mathcal{I} = \{i_1, \cdots, i_m\}$ where each user u_i expresses opinions about a subset of items. In this paper, we assume opinions are expressed through an explicit numeric rating (e.g., scale from one to five), but other rating methods such as hyperlink clicks are possible as well. We are mainly interested in recommending a set of items for an active user such that the user has not rated these items before. The rating information is summarized in an $n \times m$ matrix $\mathbf{R} \in \mathbb{R}^{n \times m}, 1 \leq i \leq n, 1 \leq j \leq m$ where the rows correspond to the users and the columns correspond to the items and (p, q)th entry is the rate given by user u_p to the item i_q. We note that the rating matrix is partially observed and it is sparse in most cases.

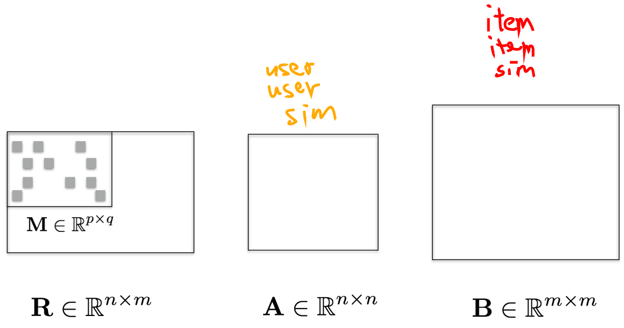
user
user
sim
item
item
sim
$\mathbf{M} \in \mathbb{R}^{p \times q}$
$\mathbf{R} \in \mathbb{R}^{n \times m}$
$\mathbf{A} \in \mathbb{R}^{n \times n}$
$\mathbf{B} \in \mathbb{R}^{m \times m}$

Figure 1: In our setting we assume that there are many fully unobserved rows and columns in the input rating matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$ and only a small submatrix $\mathbf{M} \in \mathbb{R}^{p \times q}$ has enough information to be recovered by the off-the-shelf methods such as matrix factorization or completion. Besides, the similarity matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{m \times m}$ about users and items are provided, respectively, that could potentially benefit the recommendation.

An efficient and effective approach for recommender systems is to factorize the user-item rating matrix $\mathbf{R}$ by a multiplicative of k-rank matrices $\mathbf{R} \approx \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n \times k}$ and $\mathbf{V} \in \mathbb{R}^{m \times k}$ utilize the factorized user-specific and item-specific matrices, respectively, to make further missing data prediction. There are two basic formulations to solve this problem: these are optimization approaches(see e.g., [19, 9, 11, 7]) and probabilistic approach [15]. Let $\Omega_{\mathbf{R}}$ be the set of observed ratings in the user-item matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$, i.e., $\Omega_{\mathbf{R}} = \{(i, j) \in [n] \times [m] : \mathbf{R}_{ij} \text{ has been observed}\}$, where n is the number of users and m is the number of items to be rated. In optimization based matrix factorization, the goal is to learn the latent matrices $\mathbf{U}$ and $\mathbf{V}$ by solving the following optimization problem:

$$\mathcal{L}(\mathbf{U}, \mathbf{V}) = \min_{\mathbf{U}, \mathbf{V}} \frac{1}{2} \sum_{(i,j) \in \Omega_{\mathbf{R}}} \left(\mathbf{R}_{i,j} - \mathbf{u}_i^\top \mathbf{v}_j \right)^2 + \lambda_{\mathbf{U}} \|\mathbf{U}\|_F^2 + \lambda_{\mathbf{V}} \|\mathbf{V}\|_F^2 \qquad (1)$$

The optimization problem in Eq. (1) has three main terms: the first term aims to minimize the inconsistency between the observed entries and their corresponding values obtained by the factorized matrices. The last two terms regularize the latent matrices for users and items, respectively. The parameters $\lambda_{\mathbf{U}}$ and $\lambda_{\mathbf{V}}$ are regularization parameters that are introduced to control the regularization of latent matrices $\mathbf{U}$ and $\mathbf{V}$, respectively. After learning the latent features for users and items, the prediction of each missing entry can be computed by the inner product of latent vectors of the corresponding row and the corresponding column.

For new users or items in the system, since there is no observed rating data in $\mathbf{R}$, the corresponding rows and columns in the rating matrix is fully unobserved which results in an inability to draw inferences to either recommend existing items to new users or new items to existing users. In particular, it is not feasible for standard Matrix Factorization formulation to fill in a row or a column that is entirely missing in the original rating matrix. In this paper, we assume that there is a sub-matrix $\mathbf{M} \in \mathbb{R}^{p \times q}, 1 \leq p \leq n, 1 \leq q \leq m$ which includes enough rating data to be fully recovered via standard methods such as matrix factorization or matrix completion. We call the rest of items and users for which the rating data is entirely missing and are not present in $\mathbf{M}$ as cold-start. To make recommendations to cold-start users/items we assume that besides the observed entries in the matrix $\mathbf{M}$, there exist two auxiliary similarity matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{m \times m}$ that capture the pairwise similarity between users and items, respectively as shown in Figure 1. The similarity matrices can be computed from the available information such as users' profile or social context or items' features. The main focus of this paper is on exploiting available side information to improve the accuracy of recommendations and resolve the cold-start items and users problems.

3. MATRIX FACTORIZATION WITH SIMILARITY INFORMATION

We now focus on describing our algorithm and the assumptions underlying it. We assume that the rating matrix and the side information matrices are correlated and to some extent. This will be formalized later to share the same latent information; that is, the row and column vectors in $\mathbf{R}$ share an underlying subspace spanned by the leading eigen-vectors in the similarity matrices $\mathbf{A}$ and $\mathbf{B}$, respectively. This assumption follows from the fact that the similarity matrices provide auxiliary information about the users/items, otherwise there would not be any chance to benefit from these information in addressing the cold-start problems.

Before delving into the algorithm, we discuss two alternative methods to exploit the similarity information in matrix factorization and motivate our own algorithm. A straightforward approach to exploit and transfer knowledge from similarity matrices to the rating data is to cast the problem as a shared subspace learning framework based on a joint matrix factorization to jointly learn a common subset of basis vectors for the rating matrix $\mathbf{R}$ and the corresponding similarity matrices $\mathbf{A}$ and $\mathbf{B}$ for users and items. In particular, the goal is is to factorize the matrices $\mathbf{R}$ and $\mathbf{A}$ with equal number of rows and the matrices $\mathbf{R}$ and $\mathbf{B}$ with equal number of columns into four subspaces: two are shared between rating matrix and each of two similarity matrices, and two are specific to the matrices as formulated in the following optimization problems:

≈ LCE [22]

$$\min_{\substack{\mathbf{U}\in\mathbb{R}^{n\times r},\mathbf{V}\in\mathbb{R}^{m\times r},\\ \mathbf{W}\in\mathbb{R}^{n\times r},\mathbf{Z}\in\mathbb{R}^{m\times r}}} \frac{1}{2}\|\mathbf{R}-\mathbf{U}\mathbf{V}^\top\|_{\mathrm{F}}^2+\lambda\left(\|\mathbf{U}\|_{\mathrm{F}}^2+\|\mathbf{V}\|_{\mathrm{F}}^2\right) + \frac{1}{2}\|\mathbf{A}-\mathbf{U}\mathbf{W}^\top\|_{\mathrm{F}}^2+\frac{1}{2}\|\mathbf{B}-\mathbf{Z}\mathbf{V}^\top\|_{\mathrm{F}}^2 + \lambda\left(\|\mathbf{W}\|_{\mathrm{F}}^2+\|\mathbf{Z}\|_{\mathrm{F}}^2\right) \tag{2}$$

with λ as the regularization parameter for the norms of the solution matrices and common latent space representation is achieved by using the same matrices $\mathbf{W}\in\mathbb{R}^{n\times r}$ and $\mathbf{Z}$.

An alternative approach is to kernelize the matrix factorization by assessing the similarity of latent features using the available similarity matrices [28]. In particular, instead of directly solving the optimization problem in Eq. (1), we optimize the following objective

$$\mathcal{L}(\mathbf{U},\mathbf{V}) = \min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\sum_{(i,j)\in\Omega_{\mathbf{R}}}\left(\mathbf{R}_{i,j}-\mathbf{u}_i^\top\mathbf{v}_j\right)^2 + \lambda_{\mathbf{U}}\sum_{i=1}^{n}\mathbf{u}_i^\top\mathbf{A}\mathbf{u}_i+\lambda_{\mathbf{V}}\sum_{i=1}^{m}\mathbf{v}_i^\top\mathbf{B}\mathbf{v}_i, \tag{3}$$

where the similarity matrices are explicitly used to regularize the latent features.

However, in our setting there are many rows and columns in the rating matrix $\mathbf{R}$ that are fully unobserved, which makes the subspace learning and kernelized factorization methods inapplicable. The main issue with these approaches is that the *completion* of the unobserved entries in rating matrix $\mathbf{R}$ and *transduction* of knowledge from these entries to cold-start users/items via similarity matrices is carried out simultaneously. Therefore, the completion and transduction errors are propagated repetitively and in a uncontrolled way. The issue with error propagation becomes even worse due to the non-convexity of optimization problems in Eq. (2) and Eq. (3)– jointly in parameters $\mathbf{U}$ and $\mathbf{V}$.

Algorithm 1 Matrix Factorization with Decoupled Completion and Transduction (DCT)

```
Input:
    • $\mathbf{R}\in\mathbb{R}^{n\times m}, r$: the partially observed rating matrix and its rank
    • $\mathbf{A}\in\mathbb{R}^{n\times n}$: the auxiliary users' similarity matrix
    • $\mathbf{B}\in\mathbb{R}^{m\times m}$: the auxiliary items' similarity matrix
    • $\mathbf{M}\in\mathbb{R}^{p\times q}$: the maximal recoverable rating sub-matrix
Complete the sub-matrix $\mathbf{M}$ to get $\widehat{\mathbf{M}}$    [Completion]
Decompose $\widehat{\mathbf{M}}$ as $\widehat{\mathbf{M}}=\sum_{i=1}^{r}\hat{\mathbf{u}}_i\hat{\mathbf{v}}_i^\top$
Extract subspaces $\mathbf{U}_{\mathbf{A}}$ and $\mathbf{U}_{\mathbf{B}}$ by spectral clustering from similarity matrices $\mathbf{A}$ and $\mathbf{B}$, respectively
Compute $\hat{\mathbf{a}}_i=\left(\widehat{\mathbf{U}}_{\mathbf{A}}^\top\widehat{\mathbf{U}}_{\mathbf{A}}\right)^\dagger\widehat{\mathbf{U}}_{\mathbf{A}}^\top\hat{\mathbf{u}}_i, i=1,2,\cdots,r$
Compute $\hat{\mathbf{b}}_i=\left(\widehat{\mathbf{U}}_{\mathbf{B}}^\top\widehat{\mathbf{U}}_{\mathbf{B}}\right)^\dagger\widehat{\mathbf{U}}_{\mathbf{B}}^\top\hat{\mathbf{v}}_i, i=1,2,\cdots,r$
Compute $\widehat{\mathbf{R}}=\mathbf{U}_{\mathbf{A}}\left(\sum_{i=1}^{r}\hat{\mathbf{a}}_i\hat{\mathbf{b}}_i^\top\right)\mathbf{U}_{\mathbf{B}}^\top$    [Transduction]
Output: $\widehat{\mathbf{R}}$
```

In effort to alleviate this difficulty, in this section we propose an alternative approach that diverges from these algorithms and transfers information from similarity matrices to rating matrix via the fully recoverable sub-matrix $\mathbf{M}$. In particular, the proposed algorithm **decouples** the *completion* from *transduction* and constitutes of two stages: (i) completion of the sub-matrix $\mathbf{M}$ which can be done perfectly with zero completion error, and (ii) transduction of rating data from the recovered sub-matrix to cold-start items and users using similarity matrices. This crucial difference greatly boosts the performance of the proposed algorithm when appropriate side information is incorporated.

With our assumption concerning the correlation of rating and similarity matrices in place, we can now describe our algorithm. To this end, we construct an orthogonal matrix $\mathbf{U}_{\mathbf{A}}=[\mathbf{u}_1^{\mathbf{A}},\cdots,\mathbf{u}_s^{\mathbf{A}}]\in\mathbf{R}^{n\times s}$ whose column space subsumes the row space of the rating matrix. We also construct another orthogonal matrix $\mathbf{U}_{\mathbf{B}}=[\mathbf{u}_1^{\mathbf{B}},\cdots,\mathbf{u}_s^{\mathbf{B}}]\in\mathbf{R}^{m\times s}$ whose column space subsumes the column space of the rating matrix. To construct subspaces $\mathbf{U}_{\mathbf{A}}$ and $\mathbf{U}_{\mathbf{B}}$ we use the first s eigen-vectors corresponding to the s largest eigen-values of the provided similarity matrices $\mathbf{A}$ and $\mathbf{B}$, respectively.

We note that the extent to which the extracted subspaces $\mathbf{U}_{\mathbf{A}}$ and $\mathbf{U}_{\mathbf{B}}$ from similarity matrices subsume the corresponding row and column spaces of the rating matrix, depends on the richness of the similarity information. To formalize this, first from the low-rank assumption of the rating matrix $\mathbf{R}$, it follows that it can be decomposed as $\mathbf{R}=\sum_{i=1}^{r}\mathbf{u}_i\mathbf{v}_i^\top$ where r is the rank of the matrix. Now we note that the ith latent features vector $\mathbf{u}_i$ can be decomposed in a unique way into two parallel and orthogonal parts as $\mathbf{u}_i=\mathbf{u}_i^{\parallel}+\mathbf{u}_i^{\perp}$, where $\mathbf{u}_i^{\parallel}$ is the part that is spanned by the subspace $\mathbf{U}_{\mathbf{A}}$ extracted from the similarity information about users and $\mathbf{u}_i^{\perp}$ is the part orthogonal to $\mathbf{U}_{\mathbf{A}}$.

In a similar way, for the latent features vector of jth item, i.e., $\mathbf{v}_j$, we have the following similar decomposition as $\mathbf{v}_i=\mathbf{v}_i^{\parallel}+\mathbf{v}_i^{\perp}$, where $\mathbf{v}_i^{\parallel}$ is the part that is spanned by the subspace $\mathbf{U}_{\mathbf{B}}$ extracted from the similarity information about users and $\mathbf{v}_i^{\perp}$ is the part orthogonal to $\mathbf{U}_{\mathbf{B}}$. We note that the orthogonal components of left and singular vectors $\mathbf{u}_i^{\perp}$ and $\mathbf{v}_i^{\perp}$ captures the extent to which the similarity matrices do not provide information about rating data and can not be recovered using these auxiliary information.

To build intuition for the algorithm we propose, we first relate the rating matrix to the similarity matrices. Having decomposed the latent features as above into two parallel and orthogonal components, we can rewrite the rating matrix $\mathbf{R}$ as:

$$\begin{aligned}\mathbf{R} &= \sum_{i=1}^{r} \mathbf{u}_i \mathbf{v}_i^\top = \sum_{i=1}^{r} (\mathbf{u}_i^{\|} + \mathbf{u}_i^{\perp})(\mathbf{v}_i^{\|} + \mathbf{v}_i^{\perp})^\top \\ &= \sum_{i=1}^{r} \mathbf{u}_i^{\|} \mathbf{v}_i^{\|\top} + \sum_{i=1}^{r} \mathbf{u}_i^{\|} \mathbf{v}_i^{\perp\top} + \sum_{i=1}^{r} \mathbf{u}_i^{\perp} \mathbf{v}_i^{\|\top} + \sum_{i=1}^{r} \mathbf{u}_i^{\perp} \mathbf{v}_i^{\perp\top} \\ &= \mathbf{R}_* + \mathbf{R}_{\mathrm{L}} + \mathbf{R}_{\mathrm{R}} + \mathbf{R}_{\mathrm{E}},\end{aligned} \tag{4}$$

where $\mathbf{R}_*$ is the part of the rating matrix that is fully spanned by the subspaces $\mathbf{U}_{\mathbf{A}}$ and $\mathbf{U}_{\mathbf{B}}$, the matrix $\mathbf{R}_{\mathrm{L}}$ is the part where only the left singular vectors are spanned by $\mathbf{U}_{\mathbf{A}}$ and the right singular vectors are orthogonal to the subspace spanned by $\mathbf{U}_{\mathbf{B}}$, and the matrix $\mathbf{R}_{\mathrm{R}}$ is the part that the left singular vectors are orthogonal to the subspace spanned by $\mathbf{U}_{\mathbf{A}}$ and the right singular vectors are spanned by $\mathbf{U}_{\mathbf{B}}$. Finally, the matrix $\mathbf{R}_{\mathrm{E}}$ is the error matrix where both left and singular vectors are orthogonal to the subspaces spanned by $\mathbf{U}_{\mathbf{A}}$ and $\mathbf{U}_{\mathbf{B}}$, respectively, which does not benefit form the side information at all. In particular, the error matrix $\mathbf{R}_{\mathrm{E}}$ can not be recovered from the side information as the extracted subspaces provide no information about the orthogonal parts $\mathbf{u}_i^{\perp}$ and $\mathbf{v}_i^{\perp}$ of the singular vectors. Therefore, the error contributed by this matrix into the estimation error of final recovered rating matrix is unavoidable.

In the following subsections, we first devise an effective method to recover the rating matrix $\mathbf{R}$ from the sub-matrix $\mathbf{M}$ and subspaces $\mathbf{U}_{\mathbf{A}}$ and $\mathbf{U}_{\mathbf{B}}$, and then provide theoretical guarantees on the estimation error in terms of the magnitude of the error matrix $\|\mathbf{R}_{\mathrm{E}}\|_{\mathrm{F}}$.

The Completion Stage. The first step in Algorithm 1 is to complete the sub-matrix $\mathbf{M}$ to create the fully recovered matrix $\widehat{\mathbf{M}}$. To do so, we use the matrix factorization formulation in Eq. (1) which has achieved great success and popularity among the existing matrix completion techniques [25, 15, 23]. We note that based on matrix completion theory, it is guaranteed that the recovery of the low-rank matrix $\mathbf{M}$ is perfect provided the number of observed entries is sufficiently high.

The Transduction Stage. We now turn to recovering the matrix $\mathbf{R} = \sum_{i=1}^{r} \mathbf{u}_i \mathbf{v}_i^\top$ from the submatrix $\widehat{\mathbf{M}}$ and the subspaces $\mathbf{U}_{\mathbf{A}}$ and $\mathbf{U}_{\mathbf{B}}$ extracted from the similarity matrices $\mathbf{A}$ and $\mathbf{B}$ about users and items, respectively. The detailed steps of the proposed completion algorithm are shown in Algorithm 1.

In the second step, the rating information in the recovered matrix $\widehat{\mathbf{M}}$ is transducted to the cold-start users and items. To motivate the transduction step, let us focus on the $\mathbf{R}_*$ matrix as defined in Eq. (4). Since $\mathbf{u}_i^{\|}$ and $\mathbf{v}_i^{\|}$ are fully spanned by the subspaces $\mathbf{U}_{\mathbf{A}}$ and $\mathbf{U}_{\mathbf{B}}$ following our construction above, we can write them as:

$$\mathbf{u}_i^{\|} = \mathbf{U}_{\mathbf{A}} \mathbf{a}_i, \mathbf{v}_i^{\|} = \mathbf{U}_{\mathbf{B}} \mathbf{b}_i, i = 1, 2, \cdots, r \tag{5}$$

where $\mathbf{a}_i \in \mathbb{R}^s, i = 1, 2, \cdots, r$ and $\mathbf{b}_i \in \mathbb{R}^s, i = 1, 2, \cdots, r$ are the orthogonal projection of the singular vectors to the corresponding subspaces. By substituting the equations in Eq. (5) into the decomposition of $\mathbf{R}_*$ we get:

$$\mathbf{R}_* = \sum_{i=1}^{r} \mathbf{u}_i^{\|} \mathbf{v}_i^{\|\top} = \sum_{i=1}^{r} \mathbf{U}_{\mathbf{A}} \mathbf{a}_i \mathbf{b}_i^\top \mathbf{U}_{\mathbf{B}}^\top = \mathbf{U}_{\mathbf{A}} \left(\sum_{i=1}^{r} \mathbf{a}_i \mathbf{b}_i^\top \right) \mathbf{U}_{\mathbf{B}}^\top \tag{6}$$

From above derivation, we observe that the key to recover the matrix $\mathbf{R}_*$ is to estimate the vectors $\mathbf{a}_i, \mathbf{b}_i, i = 1, 2, \cdots, r$. In the following we show how the recovered rating sub-matrix $\widehat{\mathbf{M}}$ along with the subspaces extracted from the similarity matrices can be utilized to estimate these vectors perfectly under some mild condition on the number of cold-start users and items. To this end, first consider the decomposition of the recovered matrix as $\widehat{\mathbf{M}} = \sum_{i=1}^{r} \widehat{\mathbf{u}}_i \widehat{\mathbf{v}}_i^\top$. The estimation of vectors $\mathbf{a}_i, \mathbf{b}_i, i = 1, 2, \cdots, r$ in Eq. (6) and equivalently the matrix $\mathbf{R}_*$ is as follows. First, let $\widehat{\mathbf{U}}_{\mathbf{A}} \in \mathbf{R}^{p \times s}$ be a random submatrix of $\mathbf{U}_{\mathbf{A}}$ where the sampled rows correspond to the subset of rows in the matrix $\widehat{\mathbf{M}}$. Similarly, we construct a submatrix of $\mathbf{U}_{\mathbf{B}}$ denoted by $\widehat{\mathbf{U}}_{\mathbf{B}} \in \mathbf{R}^{q \times s}$ by sampling the rows of $\mathbf{U}_{\mathbf{B}}$ corresponding to the columns in $\widehat{\mathbf{M}}$. An estimation of $\mathbf{a}_i, \mathbf{b}_i, i \in [r]$ vectors is obtained by orthogonal projection of left and right singular vectors of $\widehat{\mathbf{M}}$ onto the sampled subspaces $\widehat{\mathbf{U}}_{\mathbf{A}}$ and $\widehat{\mathbf{U}}_{\mathbf{B}}$ by solving following optimization problems:

$$\begin{aligned}\widehat{\mathbf{a}}_i &= \arg\min_{\mathbf{a} \in \mathbb{R}^s} \left\| \widehat{\mathbf{u}}_i - \widehat{\mathbf{U}}_{\mathbf{A}} \mathbf{a} \right\|_2^2, \\ \widehat{\mathbf{b}}_i &= \arg\min_{\mathbf{b} \in \mathbb{R}^s} \left\| \widehat{\mathbf{v}}_i - \widehat{\mathbf{U}}_{\mathbf{B}} \mathbf{a} \right\|_2^2.\end{aligned} \tag{7}$$

Then, we estimate the $\mathbf{R}_*$ by:

$$\begin{aligned}\widehat{\mathbf{R}}_* &= \mathbf{U}_{\mathbf{A}} \left(\sum_{i=1}^{r} \widehat{\mathbf{a}}_i \widehat{\mathbf{b}}_i^\top \right) \mathbf{U}_{\mathbf{B}}^\top \\ &= \mathbf{U}_{\mathbf{A}} \left(\widehat{\mathbf{U}}_{\mathbf{A}}^\top \widehat{\mathbf{U}}_{\mathbf{A}} \right)^\dagger \widehat{\mathbf{U}}_{\mathbf{A}}^\top \left(\sum_{i=1}^{r} \widehat{\mathbf{u}}_i \widehat{\mathbf{v}}_i^\top \right) \widehat{\mathbf{U}}_{\mathbf{B}} \left(\widehat{\mathbf{U}}_{\mathbf{B}}^\top \widehat{\mathbf{U}}_{\mathbf{B}} \right)^\dagger \mathbf{U}_{\mathbf{B}},\end{aligned}$$

where in the last equality we used the fact that $\left(\widehat{\mathbf{U}}_{\mathbf{A}}^\top \widehat{\mathbf{U}}_{\mathbf{A}} \right)^\dagger \widehat{\mathbf{U}}_{\mathbf{A}}^\top \widehat{\mathbf{u}}_i$ and $\left(\widehat{\mathbf{U}}_{\mathbf{B}}^\top \widehat{\mathbf{U}}_{\mathbf{B}} \right)^\dagger \widehat{\mathbf{U}}_{\mathbf{B}}^\top \widehat{\mathbf{v}}_i$ are the optimal solutions to the ordinary least squares regression problems in Eq. (7). Here $(\cdot)^\dagger$ denotes the Moore-Penrose pseudo inverse of a matrix. The final estimated rating matrix $\widehat{\mathbf{R}}$ is simply set to be $\widehat{\mathbf{R}} = \widehat{\mathbf{R}}_*$.

An upper bound on estimation error. In order to see the impact of similarity information on recovering the rating matrix, we theoretically analyze the estimation error. In particular, the performance of proposed algorithm on estimating the rating matrix is stated in the following theorem.

Theorem 3.1. *Let $\mathbf{R} \in \mathbb{R}^{n \times m}$ be a low-rank matrix with coherence parameter μ. Let $\mathbf{M} \in \mathbb{R}^{p \times q}$ be a sub-matrix of $\mathbf{R}$ where the rows and columns are uniformly sampled with*

$$p \geq 8\mu r \log\left(\frac{r}{\delta}\right) \text{ and } q \geq 8\mu r \log\left(\frac{r}{\delta}\right)$$

where r is the rank of the original matrix $\mathbf{R}$. Let $\widehat{\mathbf{R}}$ be the recovered matrix by Algorithm 1 using similarity matrices $\mathbf{A}$ and $\mathbf{B}$ about users and items, respectively. Then with probability $1 - \delta$, it holds:

$$\|\mathbf{R} - \widehat{\mathbf{R}}\|_{\mathrm{F}} \leq \left(1 + \frac{4nm}{pq} + \frac{2n}{p} + \frac{2m}{q}\right) \|\mathbf{R}_{\mathrm{E}}\|_{\mathrm{F}}.$$

We note that in above inequality the parameter μ is known as incoherence [3, 18] which is the prevailing assumption in analysis of matrix completion algorithms. This parameter states that the singular vectors of $\mathbf{R}$ should be de-localized so that they have a small inner product with the standard basis making a full recovery possible.

Remark 3.2. *Let us pause to make some remarks concerning the results given above. First, one can observe that* *the recovery error is stated in terms of the norm of error matrix $\mathbf{R}_{\mathrm{E}}$ which captures the extent to which the similarity matrices $\mathbf{A}$ and $\mathbf{B}$ fail to capture the rating data. This error is unavoidable even if there is no cold-start item or users, $p = n$ and $q = m$ which yields $O(1)\left\|\mathbf{R}_{\mathrm{E}}\right\|_{\mathrm{F}}$ error bound. Also, the error decreases by reducing the number of cold-start items and users as expected.*

4. EXPERIMENTS

In this section, we conduct exhaustive experiments to demonstrate the merits and advantages of the proposed algorithm. We conduct the experiments on synthetic and two well-known NIPS [1] and MovieLens [2] datasets, aiming to accomplish and answer the following fundamental questions:

- **Prediction accuracy:** How does the proposed algorithm perform in comparison to the state-of-the-art algorithms with incorporating side information of users/items. And to what degree the available side information could help in making more accurate recommendations?
- **Dealing with cold-start users:** How does exploiting similarity relationships between users affect the performance of recommending existing items to cold-start users?
- **Dealing with cold-start items:** How does exploiting similarity information between items affect the performance of recommending cold-start items to existing users?
- **Dealing with cold-start users and items simultaneously:** How does exploiting similarity information between users and items affect the performance of recommending cold-start items to cold-start users?

In the following subsections, we intend to answer these questions. First we introduce the datasets that we use in our experiments and then the metrics that we employ to evaluate the results, followed by our detailed experimental results on both synthetic and real datasets.

4.1 Metrics

We adopt the widely used the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) [6] metrics for prediction accuracy [6, 13]. Let $\mathcal{T}$ denote the set of ratings that we want to predict, i.e., $\mathcal{T} = \{(i,j) \in [n] \times [m], \mathbf{R}_{ij}$ needs to be predicted$\}$ and let $\widehat{\mathbf{R}}$ denote the prediction matrix obtained by a recommendation algorithm. Then,

$$\text{MAE} = \frac{\sum_{(i,j)\in\mathcal{T}} |\mathbf{R}_{ij} - \widehat{\mathbf{R}}_{ij}|}{|\mathcal{T}|},$$

The RMSE metric is defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{(i,j)\in\mathcal{T}} \left(\mathbf{R}_{ij} - \widehat{\mathbf{R}}_{ij}\right)^2}{|\mathcal{T}|}}.$$

We would like to emphasize that even *small improvements* in RMSE are considered valuable in the context of recommender systems. For example, the Netflix prize competition offered a $1,000,000 reward for only 10% reduction of RMSE.

Table 1: Statistics of MovieLens dataset

Statistic	Quantity
Number of users	6,040
Number of items	3,706
Number of ratings	1,000,209
Range of ratings	1 to 5
Maximum number of ratings by users	2314
Maximum number of ratings for items	3428
Average number of ratings by users	165.59
Average number of ratings for items	257.58

To measure the effectiveness of the order of the recommended users/items based on their predicted rating, we employ Mean Average Precision (MAP) [12] and Normalized Discount Cumulative Gain (NDCG) measures. Given an item i, let r_i be the relevance score of the item ranked at position i, where $r_i = 1$ if the item is relevant to the i and $r_i = 0$ otherwise. Then we can compute the Average Precision (AP) as

$$\text{AP} = \frac{\sum_i r_i \times \text{Precision@}i}{\#\text{ of relevant items}}.$$

where Precision is defined as the number of relevant items (users) divided by the number of items (users) in the dataset. MAP is the average of AP over all the users in the network.

The NDCG measure is a normalization of the Discounted Cumulative Gain (DCG) measure. DCG is a weighted sum of the degree of relevancy of the ranked users. The value of NDCG is between $[0,1]$ and at position k is defined as:

$$\text{NDCG@}k = Z_k \sum_{i=1}^{k} \frac{2^{r_i} - 1}{\log(i+1)}$$

where k is also called the scope, which means the number of top-ranked items presented to the user and Z_k is chosen such that the perfect ranking has a NDCG value of 1.

4.2 Datasets and configuration

MovieLens dataset. As our first dataset, we use the well known MovieLens dataset, which has side information about both users and items that makes it ideal for our setting. This dataset consists of 1000K ratings made by 6040 users for 3706 movies. In addition to rating data, this dataset also contains features for both users and movies. For each movie we used features such as title, year, genre, etc. For each user we extracted features such as gender, age, occupation, and location. The statistics of the dataset is given in Table 4.1.

NIPS dataset. We have also applied our algorithm to paper-author and paper-word matrices extracted from the co-author network at the NIPS conference [20]. This dataset has only side information about items. The number of users (i.e., authors) and items (i.e., papers) are 2037 and 1740, respectively. The length of feature vector of each paper is 13649, which is the size of the vocabulary in NIPS (i.e., unique tokens in all articles). The content of the papers is preprocessed such that all words are converted to lower case and stemmed and stop-words are removed. We compute the cosine similarity of the vector representation weighted with TD-IDF of papers with the ones of all other papers.

Experiments setup. We separated the dataset into two subsets, a training set that used for training the model and a test set that used for evaluating it. We partition the data into 5 equal subsets and then use one for testing and the rest for training, giving a ratio of testing to training data 1:4, and we repeat the experiment for each of the partitions.

To better evaluate the effect of utilizing the features of users and items in dealing with cold-start problem, we employ different cold-start scenarios.

- I [Existing items/Cold-start users]: Often, in a recommender system a vast majority of the users are comprised of new users. To simulate the cold-start user

[1] http://www.cs.nyu.edu/~roweis/data.html
[2] http://www.grouplens.org/node/73/

scenario, we divide the users into a disjoint training set and a test set. The representative items are selected only based on the ratings by the users in the training set. Then for the users in the test set, we make predictions only based on their ratings on the representative items.

- II [Existing users/Cold-start items]: Similarly, for cold-start item scenario, we divide the items into a training and a test set. The representative users are determined based on the training set only. Then for the test set of items, we try to predict the interests level of existing users on new items
- III [Cold-start items/Cold-start users]: In this scenario we divide both users and items into two subsets of existing and cold-start users and items, respectively. We aim to recommend the cold-start items to the cold-start users, which is the most challenging form of cold-start problems.

  The first scenario happens in systems when new users request for recommendations while the second and third scenarios usually happen in newly-launched systems, new item advertisements, news or where a recommender is always forced to recommend new items.

4.3 The baseline algorithms

To evaluate the performance of our proposed DCT algorithm, we consider a variety baseline approaches. The baseline algorithms are chosen from three types of categories: (i) those that can only deal with cold-start users, (ii) ones that can deal with cold-start users, and (iii) algorithms that are capable of dealing with both cold-start items and users. In particular, we consider the following basic algorithms:

- **RS** (Random Strategy) [10]: A simple baseline that selects at random a subset of users or items. The recommendation for cold-start users and items is a challenging case, where RS is one of the baseline methods.
- **MP** (Most Popular) [16]: Another typical cold-start recommendation algorithm is to select the most popular item. For cold-start users without historical ratings, MP recommends the highly-rated items. MP provides the same recommendation to all users based on global popularity of items. To compute the global popularity of an item we utilized the scoring method introduced in [16].
- **CBF** (Content Based Filtering) [22]: Can effectively recommend new items to existing users based on the users' historical ratings and features of items. This algorithm builds a profile for each user based on the properties of the past users's preferred items.
- **LCE** (Local Collective Embeddings) [22]: Is a matrix factorization method that exploits items' properties and past user preferences while enforcing the manifold structure exhibited by the collective embeddings. We use LCE-I and LCE-U to refer the algorithm when LCE is fed with items' and users' features, respectively. We also conduct experiments with a version of LCE without laplacian regularization which will be referred by LCE-NL [22].
- **KMF** (Kernelized Matrix Factorization) [28]: Is a matrix completion based algorithm, which incorporates external side information of the users or items into the matrix factorization process. In comparison with Probabilistic Matrix Factorization (PMF), it imposes a Gaussian Process (GP) prior over all rows of the matrix, hence the learned model explicitly captures the underlying correlation among the rows.

4.4 Experiments on synthetic dataset

In this section we present the results of our proposed algorithm on one synthetic dataset. We generated a synthetic dataset to evaluate our approach before moving forward to real datasets. First we generate two matrices $\mathbf{U} \in [0,1]^{4000\times r}$ and $\mathbf{V} \in [0,1]^{2000\times r}$. Then by using $\mathbf{U}$ and $\mathbf{V}$ we generated a rating matrix $\mathbf{R}^{4000\times 2000} = \mathbf{U}\mathbf{V}^\top$ that includes 4000 users and 2000 items. Then we generated a similarity matrix $\mathbf{A}^{4000\times 4000} = \mathbf{U}\mathbf{U}^\top$ for users and a similarity matrix $\mathbf{B}^{2000\times 2000} = \mathbf{V}\mathbf{V}^\top$ for items. Then we added random noise to the all elements of $\mathbf{A}$ and $\mathbf{B}$ where the noise follows a Gaussian distribution $\mathcal{N}(0, 0.5)$. We consider $\mathbf{A}$ and $\mathbf{B}$ as two similarity matrices between users and items, respectably.

synthetic ↙

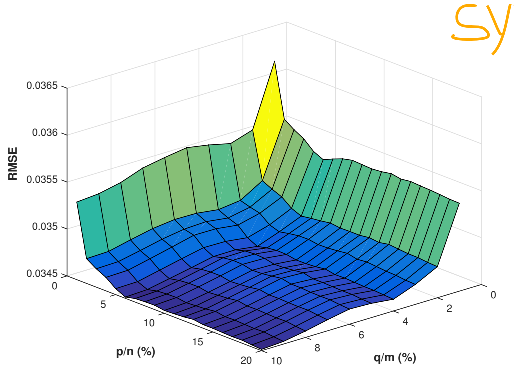

Figure 2: The RMSE of test data for different p and q values

In this study we tried different values for p within the range of 100 to 2000 with step-size of 100 and different values for q within the range of 100 to 1000 with same step-size. To generate cold-start users and items in our dataset, we divided the dataset into 4 partitions. We divided users into two groups of p existing users and $n-p$ cold-start users. We also selected q items as existing items and $m-q$ as cold-start items. Hence Partitions 1 and 2 have p users and partitions 3 and 4 have $n-p$ cold-start users; partitions 1 and 3 have q items and partitions 2 and 4 have $m-q$ cold-start items.

The statistics of the dataset is given in Table 2. The RMSE of test data for different combinations of p and q values is shown in Figure 2. It shows that by increasing p and q, the RMSE decreases. Hence, the fewer unobserved elements we have, the lower the error is. Considering the fact that our theoretical upper bound of error decreases by increasing $\frac{p}{m}$ or $\frac{q}{n}$, this observation is completely consistent with our theoretical upper bound.

For our synthetic dataset, we added noise to similarity matrices that follows a zero mean Gaussian distribution with variance of 0.5. To observe the effects of noise variance, we changed the variance from 0.1 to 1 with step size of 0.1and calculated the accuracy of our algorithm. As Figure 3 shows, by increasing the noise variance, both RMSE and MAE of the results on test dataset increase.

User cold @?

Table 3: Cold-start users on MovieLens dataset

How get rating? →

	MAE	RMSE	MAP	NDCG
RS	0.7423	0.6734	0.1234	0.4240
MP	0.7213	0.5800	0.107	0.5890
LCE-NL-U	0.7014	0.5409	0.1746	0.6047
LCE-U	0.7004	0.5400	0.1798	0.6147
KMF-U	0.6620	0.4217	0.1587	0.6214
DCT	0.5862	0.4018	0.1749	0.6223

0.58	0.40	0.18	0.62
↕	↕	↕	↑
0-74	0.67	0.11	0-42

In all our experiments, we set the value of k to the number of rated items of users for calculating NDCG.

Table 2: Synthetic DataSet

Setting	Algorithm	Training (10%)		Training (20%)		Training (30%)		Training (40%)		Training (50%)	
		MAE	RSME	MAE	RSME	MAE	RSME	MAE	RSME	MAE	RSME
Scenario I	DCT	0.0353	0.0020	0.0345	0.0018	0.0345	0.0018	0.0341	0.0018	0.0340	0.0018
Scenario II	DCT	0.0345	0.0010	0.0342	0.0019	0.0342	0.0019	0.0343	0.0019	0.0342	0.0019
Scenario III	DCT	0.0349	0.0019	0.0340	0.0018	0.0346	0.0019	0.0343	0.0019	0.0344	0.0018

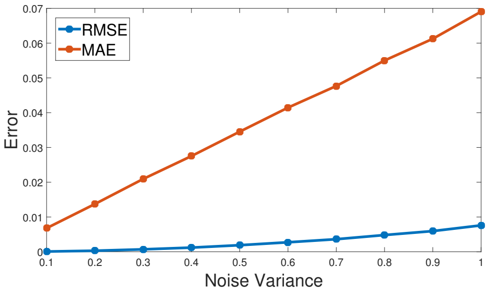

Figure 3: RMSE and MAE of test dataset for different noise variances

Both cold

Table 4: Cold-start items and cold-start users on MovieLens

How get rating?

	MAE	RMSE	MAP	NDCG @?
RS	0.9326	1.3820	0.0070	0.0652
KMF	0.8442	0.9730	0.0171	0.1834
DCT	0.7162	0.8524	0.1943	0.2783
	0.72	0.85	0.19	0.28
	0.93	1.38	0.007	0.07

4.5 Experiments of cold-start users

To simulate cold-start user scenario, we divide the users into two training and test disjoint subsets. We use 80% of the users for training and the remaining 20% of users for testing. Then we apply baseline algorithms to predict ratings of cold-start users for different items and evaluate the results using the ground truth, which is the complete rating matrix.

We compared our proposed approach to five alternative recommendation methods: RS, MP, KMF-U and two variations of the LCE algorithm. Since NIPS dataset does not have any information (features) about users, for testing this scenario we only apply these methods on MovieLens dataset. Table 3 shows the average performance of each method across the MovieLens dataset for the first scenario.

Considering the RMSE measure, we observed that DCT and KMF-U behave similarly and outperform others algorithms. DCT has also the highest MAP and outperforms others on MovieLens dataset. The LCE-U performs better than MP and the differences between two version of LCE are not significant, but always LCE performs better than LCE-NL. As expected, RS is the least accurate recommender.

4.6 Experiments of cold-start items

To simulate cold-start item scenario, we divide the items into two training and test disjoint subsets. We use 80% of the items for training and the remaining 20% of items for testing. We compared our algorithm to four alternative recommendation methods: CBF, KMF-I, LCE and LCE-NL on NIPS and MovieLens datasets. In the scenarios that we have cold-start items, since there is no historical ratings, MP cannot work. Table 5 shows the average performance of each method across the MovieLens and NIPS datasets for cold-start item scenario.

Item cold

Table 5: Experiments with cold-start items on MovieLens and NIPS datasets.

Dataset	Algorithm	Measure			
		MAE	RMSE	MAP	NDCG@2
MovieLens	CBF	0.7008	0.5418	0.0016	0.3800
	LCE-I	0.7088	0.5535	0.0091	0.4287
	LCE-NL-I	0.7087	0.5535	0.0085	0.4281
	KMF-I	0.5067	0.3176	0.0708	0.5087
	DCT	0.5143	0.3070	0.0800	0.5133
NIPS	CBF	0.8810	0.7944	0.1291	0.3862
	LCE-I	0.8576	0.7593	0.1656	0.4240
	LCE-NL-I	0.8663	0.7633	0.1623	0.4187
	KMF-I	0.8804	0.7657	0.1466	0.4219
	DCT	0.6805	0.5111	0.1954	0.4626
		0.51	0.31	0.08	0.51
		0.71	0.55	0.001	0.38

In terms of NDCG, MAP and RMSE measures, DCT outperforms all other approaches on MovieLens and NIPS datasets. Only the MAE of KMF-I is lower than DCT on MovieLensIn. Based on the evaluation measure one can conclude that DCT generally outperforms other approaches. We also observed that DCT works better for cold-start items than cold-start users on MovieLens; and that is because features of movies are much more richer than those of users in MovieLens dataset.

4.7 Experiments of cold-start items and cold-start users

In this scenario, since there is not any historical information for nor users neither items, we can only compare our algorithm with RS and KMF on MovieLens. Table 4 shows the results of applying these three algorithms on MovieLens. It is shown that DCT outperforms both RS and KMF in all measures. Hence, DCT is a novel algorithm that can deal with all three types of cold-start problems. We would like to note that DCT is able to predict an initial guess of an item popularity for cold-start items in online recommenders.

The results of Tables 3, 4 and 5 reveal the following findings:

1) RS generally performs significantly worse than the other algorithms in all scenarios for all datasets, which confirms that it is necessary to carefully use similarity information of users and items to have a more accurate recommendations.

2) The DCT outperformed all other methods by a significant margin in almost all scenarios and datasets except for the cold-start users on the NIPS dataset, in which it attained similar performance as KMF-U.

4) The results for MovieLens are generally better than the results for NIPS for all methods, possibly because of the fact that the items in MovieLens have more and richer features, while the items in NIPS are all papers with just keywords. Another explanation for the better results on MovieLens could be that MovieLens is a richer dataset than NIPS since there are more features and ratings per user/item.

5. CONCLUSIONS

In this paper, we proposed a novel factorization model to explicitly exploit similarity information about users and items to alleviate corresponding cold-start problems. In contrast to exiting methods such as subspace sharing and kernelized factorization methods, in the proposed method the completion of unobserved ratings and transduction of knowledge to cold-start items/users is decoupled. In particular, we first perform a full recovery of the sub-matrix obtained by excluding cold-start items and users, and then exploit the similarity matrices to transduct recovered ratings to cold-start users/items. The performance of the proposed algorithm is theoretically analyzed and empirically verified on synthetic and real datasets. Our results demonstrated that the proposed decoupling idea significantly improves the quality of the recommendations and alleviates the cold-start problem when rich side information about users and items is provided.

6. ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Awards IIS-096849, CCF-1117709, and 1331852. The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. We also acknowledge valuable technical discussions with Mehrdad Mahdavi and Dennis Ross during this work.

7. REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6), 2005.

[3] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[4] Rana Forsati, Mehrdad Mahdavi, Mehrnoush Shamsfard, Mohamed Sarwat. Matrix factorization with explicit trust and distrust side information for improved social recommendation. *ACM Transactions on Information Systems*, 32(4):17, 2014.

[5] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on.* IEEE, 2010.

[6] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[7] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[8] Guang Ling, Michael R Lyu, and Irwin King. Ratings meet reviews, a combined approach to recommend. In *ACM Conference on RecSys*, pages 105–112. ACM, 2014.

[9] Juntao Liu, Caihua Wu, and Wenyu Liu. Bayesian probabilistic matrix factorization with social relations and item contents for recommendation. *DSS*, 2013.

[10] Nathan N Liu, Xiangrui Meng, Chao Liu, and Qiang Yang. Wisdom of the better few: cold start recommendation via representative based rating elicitation. In *ACM RecSys*, pages 37–44. ACM, 2011.

[11] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[12] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university, 2008.

[13] Paolo Massa and Paolo Avesani. Trust-aware collaborative filtering for recommender systems. In *On Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 492–508. Springer, 2004.

[14] Aditya Krishna Menon, Krishna-Prasad Chitrapura, Sachin Garg, Deepak Agarwal, Nagaraj Kota. Response prediction using collaborative filtering with hierarchies and side-information. In *Proceedings of the 17th ACM SIGKDD, KDTM,* pages 141–149, 2011.

[15] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[16] Seung-Taek Park and Wei Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 21–28. ACM, 2009.

[17] Ian Porteous, Arthur U Asuncion, and Max Welling. Bayesian matrix factorization with side information and dirichlet process mixtures. In *AAAI*, 2010.

[18] Benjamin Recht. A simpler approach to matrix completion. *The Journal of Machine Learning Research*, 12:3413–3430, 2011.

[19] Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd ICML*, pages 713–719. ACM, 2005.

[20] S Roweis. Nips dataset (2002). *URL http://www. cs. nyu. edu/˜ roweis.*

[21] Laila Safoury and Akram Salah. Exploiting user demographic attributes for solving cold-start problem in recommender system. *Lecture Notes on Software Engineering*, 1(3):303–307, 2013.

[22] Martin Saveski and Amin Mantrach. Item cold-start recommendations: learning local collective embeddings. In *Proceedings of the 8th ACM Conference RecSys*, pages 89–96. ACM, 2014.

[23] Hanhuai Shan and Arindam Banerjee. Generalized probabilistic matrix factorizations for collaborative filtering. In *ICDM*, pages 1025–1030. IEEE, 2010.

[24] Le Hoang Son. Dealing with the new user cold-start problem in recommender systems: A comparative review. *Information Systems*, (0):–, 2014.

[25] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *NIPS*, pages 1329–1336, 2004.

[26] Michele Trevisiol, Luca Maria Aiello, Rossano Schifanella, Alejandro Jaimes. Cold-start news recommendation with domain-dependent browse graph. In *ACM RecSys*, volume 14, 2014.

[27] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. Functional matrix factorizations for cold-start recommendation. In *ACM SIGIR*, pages 315–324. ACM, 2011.

[28] Tinghui Zhou, Hanhuai Shan, Arindam Banerjee, and Guillermo Sapiro. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *SDM*, volume 12, pages 403–414. SIAM, 2012.