Cold-Start Recommendation with Provable Guarantees: A Decoupled Approach

Iman Barjasteh, Rana Forsati, Dennis Ross, Abdol-Hossein Esfahanian,
and Hayder Radha, *Fellow, IEEE*

Abstract—Although the matrix completion paradigm provides an appealing solution to the collaborative filtering problem in recommendation systems, some major issues, such as data sparsity and cold-start problems, still remain open. In particular, when the rating data for a subset of users or items is entirely missing, commonly known as the *cold-start* problem, the standard matrix completion methods are inapplicable due the non-uniform sampling of available ratings. In recent years, there has been considerable interest in dealing with cold-start users or items that are principally based on the idea of exploiting other sources of information to compensate for this lack of rating data. In this paper, we propose a novel and general algorithmic framework based on matrix completion that simultaneously exploits the similarity information among users and items to alleviate the cold-start problem. In contrast to existing methods, our proposed recommender algorithm, dubbed DecRec, *decouples* the following two aspects of the cold-start problem to effectively exploit the side information: (i) the completion of a rating sub-matrix, which is generated by excluding cold-start users/items from the original rating matrix; and (ii) the transduction of knowledge from existing ratings to cold-start items/users using side information. This crucial difference prevents the error propagation of completion and transduction, and also significantly boosts the performance when appropriate side information is incorporated. The recovery error of the proposed algorithm is analyzed theoretically and, to the best of our knowledge, this is the first algorithm that addresses the cold-start problem with provable guarantees on performance. Additionally, we also address the problem where both cold-start user and item challenges are present simultaneously. We conduct thorough experiments on real datasets that complement our theoretical results. These experiments demonstrate the effectiveness of the proposed algorithm in handling the cold-start users/items problem and mitigating data sparsity issue.

Index Terms—Recommender systems, cold-start problem, matrix completion, transduction

1 Introduction

Due to the popularity and exponential growth of e-commerce websites (e.g., Amazon, eBay) and online streaming websites (e.g., Netflix, Hulu), a compelling demand has been created for efficient recommender systems to guide users toward items of their interests (e.g., products, books, movies). Recommender systems seek to predict the rating that a user would give to an item and further try to suggest items that are most appealing to the users. Recently, recommender systems have received a considerable amount of attention and have been the main focus of many research studies [2].

Content-based filtering (CB) and collaborative filtering (CF) are well-known examples of recommendation approaches. As demonstrated by its performance in the KDD Cup [12] and Netflix competition [5], the most successful recommendation technique used is collaborative filtering. This technique exploits the users' opinions (e.g., movie ratings) and/or purchasing (e.g., watching, reading) history in order to extract a set of interesting items for each user. In factorization based CF methods, both users and items are mapped into a latent feature space based on observed ratings that are later used to make predictions.

Despite significant improvements in recommendation systems, and in particular factorization based methods, these systems suffer from a few inherent limitations and weaknesses such as *data sparsity* and *cold-start* problems. Specifically, in many real world applications, the rating data are very sparse (i.e., most users do not rate most items typically resulting in a very sparse rating matrix) or for a subset of users or items the rating data is entirely missing (knows as cold-start user and cold-start item problem, respectively [56]). To address the difficulties associated with the latter issues, there has been an active line of research during the past decade and a variety of techniques have been proposed [14], [16], [31], [33], [40], [44], [49], [61], [63], [64].

The studies in the literature have approached the cold-start problem from many different angles, but they commonly exploit the auxiliary information about the users and items in addition to the rating data that are usually available (see e.g., [58], [59] for a more recent survey). By leveraging multiple sources of information one can potentially bridge the gap between existing items (or users) and new (cold start) items (or users) to mitigate the cold-start problem.

The main motivation behind these techniques stems from the observation that other sources of data can be used to reveal more information about the underlying patterns between users and items and thus complement the rating data. For instance, knowing the underlying social connections (friends, family, etc.) between users can give us a better understanding of the sources of influence on a user's

- *I. Barjasteh and H. Radha are with the Department of Electrical and Computer and Engineering, Michigan State University, East Lansing, MI 48824. E-mail: {barjaste, radha}@egr.msu.edu.*
- *R. Forsati, D. Ross, and A.H. Esfahanian are with Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824. E-mail: {forsati, rossdenn, esfahanian}@cse.msu.edu.*

Manuscript received 17 Aug. 2015; revised 4 Jan. 2016; accepted 13 Jan. 2016. Date of publication 27 Jan. 2016; date of current version 27 Apr. 2016.
Recommended for acceptance by Y. Koren.
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.2016.2522422

decision. Subsequently, the availability of auxiliary information such as users' profile information [2], social context (trust and distrust relations) of users [15], information embedded in the review text [33], and features of items [17] provide tangible benefits to the recommender. Hence, an intriguing research question, which is the main focus of this paper is: *How can side information about users and items be effectively incorporated in factorization methods to overcome the cold-start problem?*

Although there are many different algorithms in the literature to augment matrix factorization with side information, such as shared subspace learning [55] and kernelized matrix factorization [64], the dominant paradigm in existing methods is to perform (a) the *completion* of rating matrix and (b) the *transduction* of knowledge from existing ratings to cold-start items/users simultaneously. While these methods are able to generate good results in practice, they have notable drawbacks: 1) these methods propagate the completion and transduction errors repetitively and in an uncontrolled way, 2) many of state-of-art algorithms are usually application based, e.g., see [32], [61], and do not offer a general framework for alleviating cold-start problems.

The overarching goal of our work is to answer the above question by proposing an efficient matrix factorization approach using similarity information. In fact, not only we propose a general framework for dealing with cold-start problems, but also we study a somewhat more general problem where both cold-start users and cold-start items are present simultaneously and address these two challenges simultaneously. In particular, by considering the drawbacks of the existing methods, we propose a two-stage algorithm that *decouples* the completion and transduction stages. First, we exclude the cold-start items and users and complete the rating matrix. Next, we transduct the knowledge to cold-start users/items using the recovered submatrix in addition to the available side information about the users and items. Hence, there is no error propagation of completion and transduction. Interestingly, beyond just dealing with cold-start problem, the proposed algorithm also provides an effective way to exploit the side information about users (or items) to mitigate the data sparsity and compensate for the lack of rating data.

The main focus of this work deals with cold-start problems, however, our general framework will also apply to other problems involving matrix completion with side information. One such example being network completion [38], where a small sample of a network is observed and side information about the nodes is available. Unlike most existing methods, we also provide a theoretical performance guarantee on the estimation error of our proposed algorithm. We further complement our theoretical results with comprehensive experiments on a few benchmark datasets to demonstrate the merits and advantages of proposed framework in dealing with cold-start problems. Our results demonstrate the superiority of our proposed framework over several of state-of-art cold-start recommender algorithms.

Outline. The remainder of the paper is organized as follows. In Section 2 we give a brief survey of related work. In Section 3, we begin by establishing notation and providing background on matrix factorization. We then, in Section 4, motivate and describe the proposed algorithm. The proof of recovery error is shown in Section 5. The experimental results are provided in Section 6 and concluding remarks are given in Section 7.

2 Related Research

Here we draw connections to, and put our work in context of, some of the more recent work on handling cold-start problems. The different approaches can be roughly divided into the following four categories.

Naive methods. The first category includes *naive* algorithms that try to recommend items to users based on their popularity [44], or based on a random selection [35]. Naive methods treat all cold-start users/items in a same way and assume that all users/items contribute the same to recommendations. This has the effect of tremendously reducing the accuracy due to a lack of any filtering.

Warm-start methods. For cold-start scenarios, since there is no historical data available for either users or items, warm-start methods either ask users to rate a set of items or import their preferences from another source of auxiliary information in order to expand the user profile [10], [35], [62], [63]. In particular, these methods explicitly ask a new user to rate k representative items in order to regulate the taste of new user for dealing with cold-start user problems. Similarly, a new item can be forced to be rated by k representative users in cold-start item scenarios.

Feature combination. As mentioned earlier, in recent years there has been an upsurge of interest in utilizing other rich sources of side information about items and users along with the rating matrix to increase the accuracy of the recommendation and dealing with cold-start challenges [58]. Therefore, *feature combination* approaches, as the third category of cold-start recommendations, became quite appealing. These methods employ and combine features related to users (e.g., profile) or items (e.g., metadata) to increase the accuracy while minimizing the user interactions.

The recent advances in matrix factorization methods suggest *subspace sharing* or *matrix co-factorization* can effectively incorporate side information [21], [23], [24], [36], where several matrices (rating and side information matrices) are simultaneously decomposed, sharing some factor matrices. Ocepek et al. [43] impute the missing values into the matrix factorization to boost the performance on cold-start. The kernelized matrix factorization approach studied by Zhou et al. [64], incorporates the auxiliary information into the matrix factorization to assess the similarity of latent features using the available similarity matrices. Saveski and Mantrach [55] matrix factorization method, in a common low-dimensional space, collectively decomposes the content and the collaborative matrices. We also note that several recent studies extend the maximum margin matrix factorization, which was developed for collaborative filtering [60], to incorporate side information [1]. These studies aim to overcome the data sparsity problem by reducing the number of sampled entries and is different from our work. Menon et al. in [40] proposed a matrix factorization analogue of logistic regression that applies a principled confidence-weighting scheme to its objective, where explicit features are also combined. Porteous et al. [49] introduce a Bayesian matrix factorization model that performs regression against

TABLE 1
Summary of Notations and Their Meaning

Symbol	Meaning
$\mathcal{U} = \{u_1, \ldots, u_n\}$	The set n of users
$\mathcal{I} = \{i_1, \ldots, i_m\}$	The set m of items
k	The latent dimension
$\mathbf{R} \in \mathbb{R}^{n \times m}$	The partially observed rating matrix
$\boldsymbol{\Omega}_\mathbf{R} \subseteq [n] \times [m]$	The set of existing ratings in **R**
$r \leq \min(n, m)$	The rank of rating matrix
$\mathbf{M} \in \mathbb{R}^{p \times q}$	A fully recoverable sub-matrix
$\boldsymbol{\Omega}_\mathbf{M} \subseteq \boldsymbol{\Omega}$	The set of existing ratings in **M**
$\mathbf{A} \in \mathbb{R}^{n \times n}$	The similarity matrix of users
$\mathbf{B} \in \mathbb{R}^{m \times m}$	The similarity matrix of items

side information. They also introduce a non-parametric mixture model for the prior of the rows and columns of the factored matrices that gives a different regularization for each latent class. Lika et al. [31] proposed an approach that incorporates classification methods in a pure CF system while the use of demographic data help for the identification of other users with similar behavior.

An alternative way for feature combination is to map the available auxiliary features to the latent features of the factorization model. Elbadrawy and Karypis [13] proposed an approach that learns a mapping function to transform the feature vectors of items into their latent space. Gantner et al. [17] also proposed a matrix factorization model that maps the features to the latent features of the model. Boltzmann machines [22] and aspect models [56] are other factor models that utilize side information for cold-start recommendation. Another family of feature combination methods includes those that rely on explicit features of items/users to compute the similarity between items/users by extracting different key features such as textual similarity or semantic similarity [33], [61].

Model combination. Finally, model combination methods combine the outputs of different recommeders by various strategies such as: voting [47], weighting the output score of recommenders [9], switching between recommenders [6], or filtering and re-ranking the outputs [7]. The drawback of these approaches is that they require building different and separate recommender systems to each source of used information and combining their outputs [22]. Park and Chu [44] cast the cold-start recommendation as a regression problem and applied a combination of all information of users/items to incorporate the side information.

Finally, we note that although various hybrid methods such as factorization machines [51], content-boosted collaborative filtering [39], probabilistic models [48], pairwise kernel methods [4], and filterbots-based methods [45] have been developed to blend collaborative filtering with side information, they are specifically designed to address the data sparsity problem and fail to cope with cold-start users or items problem which is the main focus of this paper.

3 The Setting

In this section we establish the notations which are used throughout the paper as summarized in Table 1. Our general convention throughout this paper is to use lower case letters such as u for scalars and bold face lower case letters such as $\mathbf{u}$ for vectors. The set of non-negative real numbers is denoted by $\mathbb{R}_+$. We use $[n]$ to denote a set on integers $\{1, 2, \ldots, n\}$. We use bold face upper case letters such as **M** to denote matrices. The transpose of a vector and a matrix denoted by $\mathbf{m}^\top$ and $\mathbf{M}^\top$, respectively. The Frobenius and spectral norms of a matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$ is denoted by $\|\mathbf{M}\|_\mathrm{F}$, i.e, $\|\mathbf{M}\|_\mathrm{F} = \sqrt{\sum_{i=1}^n \sum_{j=1}^m |\mathbf{M}_{ij}|^2}$ and $\|\mathbf{M}\|_2$, respectively. The nuclear norm of a matrix is denoted by $\|\mathbf{M}\|_* = \mathrm{trace}(\sqrt{\mathbf{M}^\top \mathbf{M}})$. We use $(\mathbf{M})^\dagger$ to denote the Moore-Penrose pseudo inverse of matrix **M**. The dot product between two vectors **m** and **n** is denoted by $\mathbf{m}^\top \mathbf{n}$.

In collaborative filtering we assume that there is a set of n users $\mathcal{U} = \{u_1, \ldots, u_n\}$ and a set of m items $\mathcal{I} = \{i_1, \ldots, i_m\}$ 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.

An efficient and effective approach for recommender systems is to factorize the user-item rating matrix **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., [29], [34], [37], [53]) and probabilistic approaches [42]. We use $\boldsymbol{\Omega}_\mathbf{R}$ to denote the set of observed ratings in the user-item matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$, i.e., $\boldsymbol{\Omega}_\mathbf{R} = \{(i, j) \in [n] \times [m] : \mathbf{R}_{ij} \ \text{ has been observed}\}$.

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 **M** as cold-start. To make recommendations to cold-start users/items we assume that besides the observed entries in the matrix **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. 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 cold-start item and user problems.

4 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, which will be formalized later, share the same latent information; that is, the row and column vectors in **R** share an underlying subspace spanned by the leading eigen-vectors of 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 and items; otherwise there would not be any hope to benefit from these information in tackling the cold-start problems.

4.1 Subspace Sharing and Error Propagation

Before delving into the algorithm, we discuss the matrix co-factorization method used to exploit the similarity information. This has been proven to be very effective for handling cold-start problems [55] and motivates 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 (a.k.a matrix co-factorization) based on a joint matrix factorization to jointly learn a common subset of basis vectors for the rating matrix $\mathbf{R}$ and the similarity matrices $\mathbf{A}$ and $\mathbf{B}$ for users and items as formulated in the following optimization problem:

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

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 $\mathbf{W}$ and $\mathbf{Z}$.

The main issue with this approach 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 uncontrollably. The issue with error propagation becomes even worse due to the non-convexity of optimization problems in Eq. (1)– jointly in parameters $\mathbf{U}$ and $\mathbf{V}$.

In an effort to alleviate this difficulty, we propose an alternative approach that diverges from these algorithms and transfers information from similarity matrices to a 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 with a very high probability, 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.

4.2 A Decoupled Solution

With our assumption on the correlation of rating and similarity matrices in place, we can now describe our algorithm. To this end, we construct an orthogonal matrix $\mathbf{U_A} = [\mathbf{u}_1^{\mathbf{A}}, \ldots, \mathbf{u}_s^{\mathbf{A}}] \in \mathbb{R}^{n\times s}$ whose column space subsumes the row space of the rating matrix. We also construct another orthogonal matrix $\mathbf{U_B} = [\mathbf{u}_1^{\mathbf{B}}, \ldots, \mathbf{u}_s^{\mathbf{B}}] \in \mathbb{R}^{m\times s}$ whose column space subsumes the column space of the rating matrix. To construct subspaces $\mathbf{U_A}$ and $\mathbf{U_B}$ 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 $\mathbf{U_A}$ and $\mathbf{U_B}$ 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 parts, parallel and orthogonal: $\mathbf{u}_i = \mathbf{u}_i^{\|} + \mathbf{u}_i^{\perp}$, where $\mathbf{u}_i^{\|}$ is the part that is spanned by the subspace $\mathbf{U_A}$ extracted from the similarity information and $\mathbf{u}_i^{\perp}$ is the part orthogonal to $\mathbf{U_A}$.

In a similar way, for the latent features vector of jth item, i.e., $\mathbf{v}_j$, we have its decomposition as $\mathbf{v}_i = \mathbf{v}_i^{\|} + \mathbf{v}_i^{\perp}$, where $\mathbf{v}_i^{\|}$ is the part that is spanned by the subspace $\mathbf{U_B}$ extracted from the similarity information about users and $\mathbf{v}_i^{\perp}$ is the part orthogonal to $\mathbf{U_B}$. We note that the orthogonal components of left and singular vectors $\mathbf{u}_i^{\perp}$ and $\mathbf{v}_i^{\perp}$ capture 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 that 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:

$$\begin{aligned} \mathbf{R} = \sum_{i=1}^r \mathbf{u}_i\mathbf{v}_i^\top &= \sum_{i=1}^r (\mathbf{u}_i^{\|} + \mathbf{u}_i^{\perp})(\mathbf{v}_i^{\|} + \mathbf{v}_i^{\perp})^\top \\ &= \sum_{i=1}^r \mathbf{u}_i^{\|}\mathbf{v}_i^{\|\top} + \sum_{i=1}^r \mathbf{u}_i^{\|}\mathbf{v}_i^{\perp\top} + \sum_{i=1}^r \mathbf{u}_i^{\perp}\mathbf{v}_i^{\|\top} + \sum_{i=1}^r \mathbf{u}_i^{\perp}\mathbf{v}_i^{\perp\top} \\ &= \mathbf{R}_* + \mathbf{R}_{\mathrm{L}} + \mathbf{R}_{\mathrm{R}} + \mathbf{R}_{\mathrm{E}}, \end{aligned} \tag{2}$$

where $\mathbf{R}_*$ is the part of the rating matrix that is fully spanned by the subspaces $\mathbf{U_A}$ and $\mathbf{U_B}$, the matrix $\mathbf{R}_{\mathrm{L}}$ is the part where only the left singular vectors are spanned by $\mathbf{U_A}$ and the right singular vectors are orthogonal to the subspace spanned by $\mathbf{U_B}$, and the matrix $\mathbf{R}_{\mathrm{R}}$ is the part that the left singular vectors are orthogonal to the subspace spanned by $\mathbf{U_A}$ and the right singular vectors are spanned by $\mathbf{U_B}$. Finally, the matrix $\mathbf{R}_{\mathrm{E}}$ is the error matrix where both left and singular vectors are orthogonal to the subspaces spanned by $\mathbf{U_A}$ and $\mathbf{U_B}$, 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 sections, we first devise an effective method to recover the rating matrix $\mathbf{R}$ from the sub-matrix $\mathbf{M}$ and subspaces $\mathbf{U_A}$ and $\mathbf{U_B}$, 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 extract a sub-matrix $\mathbf{M} \in \mathbb{R}^{p\times q}$. To be able to reconstruct $\mathbf{M}$ properly, number of observed elements should be at least $\Omega(r(p+q)\log^2(2p))$ [50, Theorem 1.1]. In order to satisfy the matrix completion conditions, we sort the rows based on the number of observed elements to create a sub-matrix on the top with the least sparse rows. Then we pick the largest sub-matrix from top that satisfies the matrix completion conditions. The next step, after extracting the sub-matrix $\mathbf{M}$, is to complete $\mathbf{M}$ to get the fully recovered matrix $\widehat{\mathbf{M}}$. To do so, we use the matrix factorization formulation which has achieved great success and popularity among the existing matrix completion techniques [42], [57], [60]. In particular, we solve the following convex optimization algorithm [8] to fully recover the submatrix:

$$\begin{aligned}\widehat{\mathbf{M}} = \arg\min_{\mathbf{X}\in\mathbb{R}^{p\times q}} \quad & \|\mathbf{X}\|_* \\ \text{s.t.} \quad & \mathbf{X}_{ij} = \mathbf{M}_{ij},\ \forall\ (i,j)\in\Omega_{\mathbf{M}},\end{aligned} \tag{3}$$

where $\Omega_{\mathbf{M}} \subseteq \Omega$ is the set of observed ratings in $\mathbf{M}$.

Algorithm 1. Factorization with Decoupled Completion and Transduction

```
Input:
  ❶ $\mathbf{R}\in\mathbb{R}^{n\times m}, r$: observed matrix and its rank
  ❷ $\mathbf{A}\in\mathbb{R}^{n\times n}$: the users' similarity matrix
  ❸ $\mathbf{B}\in\mathbb{R}^{m\times m}$: the items' similarity matrix
Extract the maximal recoverable rating sub-matrix $\mathbf{M}\in\mathbb{R}^{p\times q}$ (according to Theorem 5.1)
Complete the sub-matrix $\mathbf{M}$ to get $\widehat{\mathbf{M}}$ (according to equation (3)
Decompose $\widehat{\mathbf{M}}$ as $\widehat{\mathbf{M}} = \sum_{i=1}^r \widehat{\mathbf{u}}_i\widehat{\mathbf{v}}_i^\top$
Extract subspaces $\mathbf{U_A}$ and $\mathbf{U_B}$ by spectral clustering from similarity matrices $\mathbf{A}$ and $\mathbf{B}$, respectively
Compute $\widehat{\mathbf{a}}_i = (\widehat{\mathbf{U}}_\mathbf{A}^\top\widehat{\mathbf{U}}_\mathbf{A})^\dagger\widehat{\mathbf{U}}_\mathbf{A}^\top\widehat{\mathbf{u}}_i, i=1,2,\ldots,r$
Compute $\widehat{\mathbf{b}}_i = (\widehat{\mathbf{U}}_\mathbf{B}^\top\widehat{\mathbf{U}}_\mathbf{B})^\dagger\widehat{\mathbf{U}}_\mathbf{B}^\top\widehat{\mathbf{v}}_i, i=1,2,\ldots,r$
Compute $\widehat{\mathbf{R}} = \mathbf{U_A}(\sum_{i=1}^r \widehat{\mathbf{a}}_i\widehat{\mathbf{b}}_i^\top)\mathbf{U}_\mathbf{B}^\top$
Output: $\widehat{\mathbf{R}}$
```

We note that based on matrix completion theory, it is guaranteed that the low rank matrix $\mathbf{M}$ can be perfectly recovered provided that the number of observed entries is sufficient.

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 $\mathbf{U_A}$ and $\mathbf{U_B}$ 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 next step, the rating information in the recovered matrix $\widehat{\mathbf{M}}$ is transduced to the cold-start users and items. To motivate the transduction step, let us focus on the $\mathbf{R}_*$ matrix as defined in Eq. (2). Since $\mathbf{u}_i^{\|}$ and $\mathbf{v}_i^{\|}$ are fully spanned by the subspaces $\mathbf{U_A}$ and $\mathbf{U_B}$ following our construction above, we can write them as:

$$\mathbf{u}_i^{\|} = \mathbf{U_A}\mathbf{a}_i, \mathbf{v}_i^{\|} = \mathbf{U_B}\mathbf{b}_i, i=1,2,\ldots,r, \tag{4}$$

where $\mathbf{a}_i\in\mathbb{R}^s$ and $\mathbf{b}_i\in\mathbb{R}^s, i=1,2,\ldots,r$ are the orthogonal projection of the singular vectors onto the corresponding subspaces. By substituting the equations in Eq. (4) into the decomposition of $\mathbf{R}_*$ we get:

$$\mathbf{R}_* = \sum_{i=1}^r \mathbf{U_A}\mathbf{a}_i\mathbf{b}_i^\top\mathbf{U}_\mathbf{B}^\top = \mathbf{U_A}\left(\sum_{i=1}^r \mathbf{a}_i\mathbf{b}_i^\top\right)\mathbf{U}_\mathbf{B}^\top. \tag{5}$$

From the above derivation, we observe that the key for the recovery of the matrix $\mathbf{R}_*$ is to estimate the vectors $\mathbf{a}_i, \mathbf{b}_i, i=1,2,\ldots,r$. Next 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 under some mild conditions 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 $\mathbf{a}_i, \mathbf{b}_i, i=1,2,\ldots,r$ in Eq. (5) 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 $\mathbf{U_A}$ where the sampled rows correspond to the subset of rows in the matrix $\widehat{\mathbf{M}}$. Similarly we construct a submatrix of $\mathbf{U_B}$ denoted by $\widehat{\mathbf{U}}_\mathbf{B}\in\mathbf{R}^{q\times s}$ by sampling the rows of $\mathbf{U_B}$ 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:

$$\begin{aligned}\widehat{\mathbf{a}}_i &= \arg\min_{\mathbf{a}\in\mathbb{R}^s} \left\|\widehat{\mathbf{u}}_i - \widehat{\mathbf{U}}_\mathbf{A}\mathbf{a}\right\|_2^2, \\ \widehat{\mathbf{b}}_i &= \arg\min_{\mathbf{b}\in\mathbb{R}^s} \left\|\widehat{\mathbf{v}}_i - \widehat{\mathbf{U}}_\mathbf{B}\mathbf{b}\right\|_2^2.\end{aligned} \tag{6}$$

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

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

where in the last equality we used the fact that $(\widehat{\mathbf{U}}_\mathbf{A}^\top\widehat{\mathbf{U}}_\mathbf{A})^\dagger\widehat{\mathbf{U}}_\mathbf{A}^\top\widehat{\mathbf{u}}_i$ and $(\widehat{\mathbf{U}}_\mathbf{B}^\top\widehat{\mathbf{U}}_\mathbf{B})^\dagger\widehat{\mathbf{U}}_\mathbf{B}^\top\widehat{\mathbf{v}}_i$ are the optimal solutions to the ordinary least squares regression problems in Eq. (6). 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 $\widehat{\mathbf{R}} = \widehat{\mathbf{R}}_*$.

5 Analysis of Recovery 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 5.1. *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 $\mathbf{R}$ and δ is a small positive number $0 < \delta < 1$. 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 this inequality μ is sometimes known as incoherence [8], [50] which is the prevailing parameter used in the analysis of matrix completion algorithms. This parameter provides a measure of how much the singular vectors or $\mathbf{R}$ are de-localized in the sense of having small inner product with the standard basis to make the full recovery possible.

Definition 5.2. *An $n \times n$ matrix $\mathbf{M}$ with singular value decomposition $\mathbf{M} = \mathbf{U\Sigma V}^\top$ is μ-*incoherent *if*

$$\max_{i,j} |\mathbf{U}_{ij}| \leq \frac{\sqrt{\mu}}{\sqrt{n}} \text{ and } \max_{i,j} |\mathbf{V}_{ij}| \leq \frac{\sqrt{\mu}}{\sqrt{n}}. \tag{7}$$

Remark 5.3. The recovery error is stated in terms of the norm of the 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 $O(1)\|\mathbf{R}_\mathrm{E}\|_\mathrm{F}$ error bound. Also, the error decreases by reducing the number of cold-start items and users.

To facilitate our analysis, let us define two auxiliary projection matrices. Since we assume that the cold-start users and items are uniformly sampled, the sub-matrix $\mathbf{M}$ can be formed by uniform sampling from two permutation matrices corresponding to rows and columns and applying to the rating matrix $\mathbf{R}$. In particular, let $\mathbf{\Pi}_1 \in \{0,1\}^{p\times n}$ be a random matrix distributed as the first p rows of uniformly random permutation matrix of size n where selected rows correspond to the rows in $\mathbf{M}$. In a similar way $\mathbf{\Pi}_1 \in \{0,1\}^{n\times q}$ be a random matrix distributed as the first q columns of uniformly random permutation matrix of size n where selected columns correspond to the columns in $\mathbf{M}$. Therefore, we can write $\mathbf{M} = \mathbf{\Pi}_1\mathbf{R}\mathbf{\Pi}_2$. Based on Eq. (2) we have:

$$\begin{aligned}\|\mathbf{R} - \widehat{\mathbf{R}}\|_\mathrm{F} &= \|\mathbf{R}_* + \mathbf{R}_\mathrm{L} + \mathbf{R}_\mathrm{R} + \mathbf{R}_\mathrm{E} - \widehat{\mathbf{R}}\|_\mathrm{F} \\ &\leq \underbrace{\|\mathbf{R}_* + \mathbf{R}_\mathrm{L} + \mathbf{R}_\mathrm{R} - \widehat{\mathbf{R}}\|_\mathrm{F}}_{\text{(I)}} + \underbrace{\|\mathbf{R}_\mathrm{E}\|_\mathrm{F}}_{\text{(II)}}.\end{aligned} \tag{8}$$

In what follows, we upper bound the first term (I) in above inequality as a function of (II) which immediately implies Theorem 5.1. To do so, we begin by relating the left and singular vectors $\widehat{\mathbf{u}}_i, i \in [r]$ and $\widehat{\mathbf{v}}_i, i \in [r]$ of the recovered sub-matrix $\widehat{\mathbf{M}}$ to the left and right singular vectors of the underlying matrix $\mathbf{R}$, i.e., $\mathbf{u}_i, \mathbf{v}_i, i \in [r]$. To bound (I) in terms of (II) we proceed by writing the matrices in terms of parallel and orthogonal parts of the left and right singular vectors of matrix $\mathbf{R}$ as:

$$\begin{aligned}&\|\widehat{\mathbf{R}} - \mathbf{R}_* - \mathbf{R}_\mathrm{L} - \mathbf{R}_\mathrm{R}\|_\mathrm{F} \\ &= \|\mathbf{U_A}\left(\sum_{i=1}^r \widehat{\mathbf{a}}_i\widehat{\mathbf{b}}_i^\top\right)\mathbf{U}_\mathbf{B}^\top - \sum_{i=1}^r \mathbf{u}_i^{\|}\mathbf{v}_i^{\|\top} - \sum_{i=1}^r \mathbf{u}_i^{\|}\mathbf{v}_i^{\perp\top} \\ &\quad - \sum_{i=1}^r \mathbf{u}_i^{\perp}\mathbf{v}_i^{\|\top}\|_\mathrm{F}.\end{aligned} \tag{9}$$

We continue by writing $\widehat{\mathbf{a}}_i$ $\widehat{\mathbf{b}}_i$ in above equation in terms of the parallel and orthogonal components of latent feature vectors $\mathbf{u}_i, \mathbf{v}_i, i \in [r]$. To this end, we note that $\widehat{\mathbf{u}}_i^{\|} = \mathbf{\Pi}_1\mathbf{u}_i$, $\widehat{\mathbf{v}}_i^{\|} = \mathbf{\Pi}_1\mathbf{v}_i$ and $\widehat{\mathbf{U}}_\mathbf{A} = \mathbf{\Pi}_1\mathbf{U_A}$, $\widehat{\mathbf{U}}_\mathbf{B} = \mathbf{\Pi}_2\mathbf{U_B}$. Since $\widehat{\mathbf{a}}_i$ and $\widehat{\mathbf{b}}_i$ are the optimal solutions to optimization problems in (6), using $(\mathbf{AB})^\mathrm{T} = \mathbf{B}^\mathrm{T}\mathbf{A}^\mathrm{T}$ and $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ we can rewrite $\widehat{\mathbf{a}}_i$ and $\widehat{\mathbf{b}}_i$ as $\widehat{\mathbf{a}}_i = \mathbf{a}_i + (\widehat{\mathbf{U}}_\mathbf{A}^\top)^\dagger\mathbf{\Pi}_1\mathbf{u}_i^\perp$ and $\widehat{\mathbf{b}}_i = \mathbf{b}_i + (\widehat{\mathbf{U}}_\mathbf{B}^\top)^\dagger\mathbf{\Pi}_2\mathbf{v}_i^\perp$. Substituting with the above equations, we get

$$\begin{aligned}\widehat{\mathbf{R}} &= \mathbf{U_A}\left(\sum_{i=1}^r \widehat{\mathbf{a}}_i\widehat{\mathbf{b}}_i^\top\right)\mathbf{U}_\mathbf{B}^\top \\ &= \mathbf{U_A}\left(\sum_{i=1}^r\left(\mathbf{a}_i + \left(\widehat{\mathbf{U}}_\mathbf{A}^\top\right)^\dagger\mathbf{\Pi}_1\mathbf{u}_i^\perp\right)\left(\mathbf{b}_i + \left(\widehat{\mathbf{U}}_\mathbf{B}^\top\right)^\dagger\mathbf{\Pi}_2\mathbf{v}_i^\perp\right)^\top\right) \\ \mathbf{U}_\mathbf{B}^\top &= \underbrace{\mathbf{U_A}\left(\sum_{i=1}^r \mathbf{a}_i\mathbf{b}_i^\top\right)\mathbf{U}_\mathbf{B}^\top}_{\text{III}} + \mathbf{U_A}\left(\sum_{i=1}^r \mathbf{a}_i\mathbf{v}_i^{\perp\top}\right)\mathbf{\Pi}_2^\top\left(\widehat{\mathbf{U}}_\mathbf{B}\right)^\dagger\mathbf{U}_\mathbf{B}^\top \\ &\quad + \mathbf{U_A}\left(\widehat{\mathbf{U}}_\mathbf{A}^\top\right)^\dagger\mathbf{\Pi}_1\left(\sum_{i=1}^r \mathbf{u}_i^\perp\mathbf{b}_i^\top\right)\mathbf{U}_\mathbf{B}^\top \\ &\quad \underbrace{+ \mathbf{U_A}\left(\widehat{\mathbf{U}}_\mathbf{A}^\top\right)^\dagger\mathbf{\Pi}_1\left(\sum_{i=1}^r \mathbf{u}_i^\perp\mathbf{v}_i^{\perp\top}\right)\mathbf{\Pi}_2^\top\left(\widehat{\mathbf{U}}_\mathbf{B}\right)^\dagger\mathbf{U}_\mathbf{B}^\top}_{\text{(IV)}}.\end{aligned} \tag{10}$$

Based on our decomposition of eigenevectors as,

$$\mathbf{u}_i^{\|} = \mathbf{U_A}\mathbf{a}_i, \quad \mathbf{v}_i^{\|} = \mathbf{U_B}\mathbf{b}_i, i = 1, 2, \ldots, r, \tag{11}$$

and the definition of $\mathbf{R}_*$ we have

$$\mathbf{R}_* = \sum_{i=1}^r \mathbf{u}_i^{\|}\mathbf{v}_i^{\|\top} = \mathbf{U_A}\left(\sum_{i=1}^r \mathbf{a}_i\mathbf{b}_i^\top\right)\mathbf{U}_\mathbf{B}^\top, \tag{12}$$

which cancels the term (III) in above equality. To further simplify the above result, first we note that the norm of last term (IV) in above equality can be bounded as:

$$\begin{aligned}&\left\|\mathbf{U_A}\left(\widehat{\mathbf{U}}_\mathbf{A}^\top\right)^\dagger\mathbf{\Pi}_1\left(\sum_{i=1}^r \mathbf{u}_i^\perp\mathbf{v}_i^{\perp\top}\right)\mathbf{\Pi}_2^\top\left(\widehat{\mathbf{U}}_\mathbf{B}\right)^\dagger\mathbf{U}_\mathbf{B}^\top\right\|_\mathrm{F} \\ &\leq \left\|\left(\widehat{\mathbf{U}}_\mathbf{A}^\top\right)^\dagger\left(\widehat{\mathbf{U}}_\mathbf{B}\right)^\dagger\right\|_2\left\|\sum_{i=1}^r \mathbf{u}_i^\perp\mathbf{v}_i^{\perp\top}\right\|_\mathrm{F} \\ &\leq \left\|\left(\widehat{\mathbf{U}}_\mathbf{A}^\top\right)^\dagger\left(\widehat{\mathbf{U}}_\mathbf{B}\right)^\dagger\right\|_2\|\mathbf{R}_\mathrm{E}\|_\mathrm{F}.\end{aligned}$$

By substituting (10) and (12) into (9) and using the fact that $\|\mathbf{A}+\mathbf{B}\|_\mathrm{F} \leq \|\mathbf{A}\|_\mathrm{F} + \|\mathbf{B}\|_\mathrm{F}$, we can simplify the error for (I) as:

$$\begin{aligned}&\|\widehat{\mathbf{R}} - \mathbf{R}_* - \mathbf{R}_\mathrm{L} - \mathbf{R}_\mathrm{R}\|_\mathrm{F} \\ &\leq \left\|\left(\widehat{\mathbf{U}}_\mathbf{A}^\top\right)^\dagger\left(\widehat{\mathbf{U}}_\mathbf{B}\right)^\dagger\right\|_2\|\mathbf{R}_\mathrm{E}\|_\mathrm{F} + \left\|\left(\widehat{\mathbf{U}}_\mathbf{A}^\top\right)^\dagger\right\|_2\|\mathbf{R}_\mathrm{E}\|_\mathrm{F} \\ &\quad + \left\|\left(\widehat{\mathbf{U}}_\mathbf{B}\right)^\dagger\right\|_2\|\mathbf{R}_\mathrm{E}\|_\mathrm{F}.\end{aligned} \tag{13}$$

We are only left to bound the spectral norms in the last inequality. To do so, we need the following result [20]:

Lemma 5.4. *Let $\mathbf{U}$ be an n by k matrix with orthonormal columns. Take μ to be the coherence of $\mathbf{U}$. Select $\epsilon \in (0,1)$ and a* *nonzero failure probability δ. Let $\mathbf{\Pi}$ be a random matrix distributed as the first p columns of a uniformly random permutation matrix of size n, where*

$$p \geq \frac{2\mu}{(1-\epsilon)^2} k \log\left(\frac{k}{\delta}\right)$$

Then with probability exceeding $1-\delta$, the matrix $\mathbf{U}^\top \mathbf{\Pi}$ has full row rank and satisfies

$$\left\| \left(\mathbf{U}^\top \mathbf{\Pi}\right)^\dagger \right\|_2^2 \leq \frac{n}{\epsilon p}.$$

Equipped with Lemma 5.4 under the assumption on p and q made in the theorem the following the sampled subspaces $\widehat{\mathbf{U}}_\mathbf{A}$ and $\widehat{\mathbf{U}}_\mathbf{B}$ are full rank and their norm is bounded by $\|\widehat{\mathbf{U}}_\mathbf{A}\|_2 \leq \frac{n}{\epsilon p}$ and $\|\widehat{\mathbf{U}}_\mathbf{B}\|_2 \leq \frac{m}{\epsilon q}$, respectively. Therefore, the optimal solution to the orthogonal projections onto these spaces can be written as $\widehat{\mathbf{a}}_i = (\widehat{\mathbf{U}}_\mathbf{A}^\top \widehat{\mathbf{U}}_\mathbf{A})^{-1} \widehat{\mathbf{U}}_\mathbf{A}^\top \widehat{\mathbf{u}}_i$ and $\widehat{\mathbf{b}}_i = (\widehat{\mathbf{U}}_\mathbf{B}^\top \widehat{\mathbf{U}}_\mathbf{B})^{-1} \widehat{\mathbf{U}}_\mathbf{B}^\top \widehat{\mathbf{v}}_i$. Applying the bound in Lemma 5.4 to inequality (13) we get:

$$\|\widehat{\mathbf{R}} - \mathbf{R}_* - \mathbf{R}_\mathrm{L} - \mathbf{R}_\mathrm{R}\|_\mathrm{F} \leq \left(\frac{4nm}{pq} + \frac{2n}{p} + \frac{2m}{q}\right) \|\mathbf{R}_\mathrm{E}\|_\mathrm{F}, \quad (14)$$

where for simplicity we used $\epsilon = \frac{1}{2}$. Combining this with (8) yields:

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

which gives the bound stated in the theorem and completes the proof.

6 Experiments

In this section, we conduct exhaustive experiments on multiple datasets and compare DecRec over a set of baseline algorithms to demonstrate the merits and advantages of DecRec [11]. We conduct our experiments on three well-known datasets: MovieLens (1 M and 100 K),[1] Epinions[2] and NIPS.[3] Using these, we explore several 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. Further, to what degree can the available side information help in making more accurate recommendations existing items for existing users?
- *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?

TABLE 2
Statistics of Real Datasets Used in Our Experiments

Statistics	MovieLens 100 K	MovieLens 1 M	Epinioins	NIPS
Number of users	943	6,040	8,577	2,073
Number of items	1,682	3,706	3,769	1,740
Number of ratings	100,000	1,000,209	203,275	3,990
Range of ratings	1-5	1-5	1-5	0-1

6.1 Datasets

Since our proposed general framework is applicable solution for both cold-start and network completion problems, we selected real well-known datasets for each each problem. We used MovieLens and Epinions as two samples of cold-start problem and used NIPS dataset as a network completion problem. The statistics of all datasets are given in Table 2.

MovieLens dataset. We used two of the well known MovieLens datasets, 100 K and 1 M. They consist of ratings (1-5) from users on movies. In addition to rating data, these datasets also contain features for both users and movies. For each movie we used features such as title, year, genre, etc[4] and for each user we extracted features such as gender, age, occupation, and location. Then for both users and items we computed their cosine similarities to be used as side information.

Epinions dataset. This dataset is obtained from a user-oriented product review website that has a trust network of users. Users can specify whether they trust other users or not. This trust network allows us to make a 0/1 trust connectivity vector for each user with all other users. From this we then computed the cosine similarity of trust vectors and that became our similarity matrix of users.

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 [54]. The contents of the papers are pre-processed such that all words are converted to lower cases 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.

6.2 Metrics

We adopt the widely used the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) metrics for prediction accuracy [25]. Let $\mathcal{T}$ denote the set of ratings that we want to predict, i.e., $\mathcal{T} = \{(i,j) \in [n] \times [m], \mathbf{R}_{ij} \text{ needs to be predicted}\}$ and let $\widehat{\mathbf{R}}$ denote the prediction matrix obtained by a recommendation algorithm. Then,

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

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

[4] In addition to the features provided by MovieLens, there are many other features available at www.imdb.comthat we also utilized as side information.

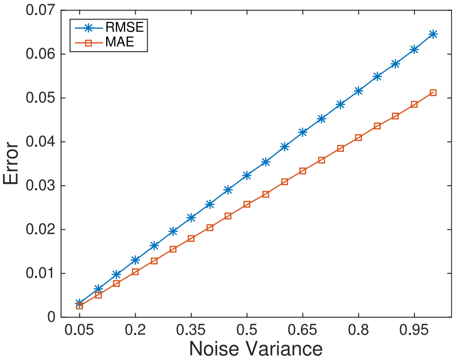

Fig. 1. RMSE and MAE of synthetic for different noise variances on similarity matrices.

The RMSE metric is defined as:

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

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. 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:

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

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

6.3 Robustness to Noise in Similarity Matrix

In this section we present the sensitivity analysis of the similarity matrix to noise. Our goal is to analyze the behavior of DecRec by varying the noise of similarity matrix. We present the analysis on two synthetic and real datasets.

Synthetic dataset. We generated a synthetic dataset to evaluate our approach before applying it to real datasets. First, we generated two matrices $\mathbf{U} \in [0,1]^{4{,}000\times r}$ and $\mathbf{V} \in [0,1]^{2{,}000\times r}$. Then by using $\mathbf{U}$ and $\mathbf{V}$ we generated a rating matrix $\mathbf{R}^{4{,}000\times 2{,}000} = \mathbf{U}\mathbf{V}^\top$ that includes 4,000 users and 2,000 items. Then we generated a similarity matrix $\mathbf{A}^{4{,}000\times 4{,}000} = \mathbf{U}\mathbf{U}^\top$ for users and a similarity matrix $\mathbf{B}^{2{,}000\times 2{,}000} = \mathbf{V}\mathbf{V}^\top$ for items. Finally, 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 the two similarity matrices between users and items, respectably. To investigate the effects of noise variance, we varied the variance from 0.05 to 0.95 with step size of 0.05 and calculate the accuracy of DecRec. As Fig. 1 shows, by increasing the noise variance, both RMSE and MAE of the results on test data increase.

Real dataset. To further investigate the effects of noise, we also chose MovieLens 1M. We generated noise that also follows a Gaussian distribution $\mathcal{N}(0, 0.5)$ for adding to the similarity matrices. We varied the noise variance from 0.05 to 0.95 and Fig. 2 shows the RMSE and MAE of the predictions. As it is shown in both Figs. 1 and 2, by adding noise to the similarity matrices, we observe an increase in prediction errors of both synthetic and real datasets. Hence, we can show the stability of DecRec with respect to noise.

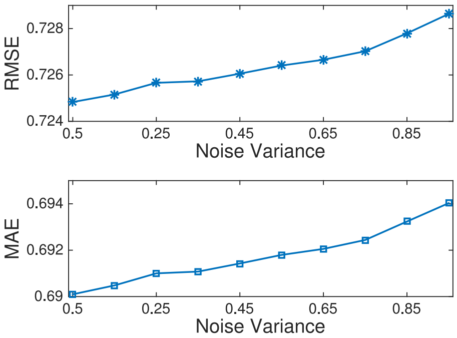

Fig. 2. RMSE and MAE of MovieLens 1 M for different noise variances on similarity matrices.

6.4 Experiments Setup

To better evaluate the effect of utilizing the features of users and items, we studied the following recommendation scenarios.

- *I [existing users/existing items]*: There are many different algorithms for predicting the rating of existing users on those existing items that they have not rated yet. We randomly removed 20 percent of the available ratings from the rating matrix and used the rest of the ratings for our training to predict the removed values.
- *II [existing users/cold-start items]*: In this case we have a number of cold-start items with no cold-start users. To simulate this scenario, we considered 80 percent of the items as training items and the remaining 20 percent as cold-start items. Then we tried to predict the ratings of existing users on the cold-start items.
- *III [cold-start users/existing items]*: Often, in a recommender system a vast majority of the users are new users. To simulate the cold-start user scenario, we divided the users into a disjoint training (80 percent) set and a test (%20) set. Then we predicted the rating of of cold-start users on the existing items.
- *IV [cold-start users/cold-start items]*: In this final scenario we divided both users and items into two subsets of existing and cold-start users and items, respectively. We aimed to recommend the cold-start items to the cold-start users, which is the most challenging form of cold-start problem.

We also analyzed the running time needed for training of our algorithm and cold-start scenario baseline algorithms to better evaluate the cold-start scenarios. It is also worth mentioning that we tuned the parameter of our algorithm by running it on training sets and reported these values for each result.

6.5 The Baseline Algorithms

To evaluate the performance of our proposed DecRec algorithm, we considered a variety baseline approaches. The baseline algorithms are chosen from four types of categories: (i) state-of-the-art algorithms for rating predictions, (ii) those that can only deal with cold-start items, (iii) ones that can deal with cold-start users and (iv) algorithms that are capable of dealing with both cold-start items and users. In particular, we considered the following basic algorithms[5]:

- **Random Strategy (RS)** [35]: 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.
- **User KNN (U-KNN)** [28]: Predicts the rates using the similarity with the K nearest neighbor where users have weights.
- **Item KNN (I-KNN)** [28]: Is a weighted item-based KNN approach for rate prediction.
- **Global Average (GA)**: Uses the average of ratings over all ratings.
- **Item Average (I-A)**: Uses the average rating of an item for its prediction.
- **User Average (U-A)**: Uses the average rating of a user for its prediction.
- **Slope One (SO)** [30]: Pre-computes the average difference between two items that are rated by users. SO is a frequency weighted slope one rating prediction.
- **BiPolar Slope One (BPSO)** [30]: Is a Bi-polar frequency weighted slope one rating prediction.
- **Social Matrix Factorization (SMF)** [26]: Is a matrix factorization algorithm that incorporates the social network for prediction.
- **CoClustering (CC)** [19]: Is a weighted co-clustering algorithm that involves simultaneous clustering of users and items.
- **Latent Feature Log Linear Model (LFLLM)** [41]: Is a scalable log-linear model that exploits the side information for dyadic prediction.
- **User Item Baseline (U-I-B)** [28]: Is a rating prediction method that uses the average rating value along with a regularized user and item bias.
- **FactorWise Matrix Factorization (FWMF)**: Is a matrix factorization based model with a factor-wise learning
- **Biased Matrix Factorization (BMF)** [52]: Is a matrix factorization that learns by stochastic gradient descent with explicit bias for users and items.
- **SVD++ (SVDPP)** [27]: Is a matrix factorization that takes into account what users have rated and directly profiles users and items.
- **Sigmoid SVD++ (SSVDPP)** [27]: Is a version of SVD ++ that variant that uses a sigmoid function.
- **Sigmoid User Asymmetric Factor Model (SU-AFM)** [46]: Is an asymmetric factor model that represents the items in terms of those users that rated them.
- **Sigmoid Item Asymmetric Factor Model (SI-AFM)** [46]: Is an asymmetric factor model that represents users in terms of the items they rated.
- **Sigmoid Combined Asymmetric Factor Model (SCAFM)** [46]: Is an asymmetric factor model that represents items in terms of the users that rated them, and users in terms of the items they rated.
- **Content Based Filtering (CBF)** [55]: This algorithm builds a profile for each user based on the properties of the past user's preferred items.
- **Local Collective Embeddings (LCE)** [55]: Is a matrix factorization method that exploits properties of items and past user preferences while enforcing the manifold structure exhibited by the collective embeddings. We also conduct experiments with a version of LCE without laplacian regularization, which will be referred as LCE-NL [55].
- **Kernelized Matrix Factorization (KMF)** [64]: Is a matrix completion based algorithm, which incorporates external side information of the users or items into the matrix factorization process.
- **Extended LCE (ELCE)**: Is an extended version of LCE (collaborative factorization method as formulated in Eq. (1)), meant to handle the challenge of the presence of both cold-start users and items simultaneously.
- **DecRec**: Our proposed algorithm.

6.6 Existing Users/Existing Items

Scenario I is the standard case for recommender systems that all users and items have a history of rating and being rated, respectively. Each user usually rates only a number of items and there are no ratings for other items. The goal here is to predict the ratings for those unrated items. There are many different techniques for predicting the rates but we can divide them into two major categories: *neighbor based approaches* and *latent factor models*. We selected our baseline algorithms from each of these categories for our companyarison.

To show the results, we applied DecRec on MovieLens 100 K, MovieLens 1 M and Epinions datasets. GroupLens Research[6] made five sets available, which are 80 percent/20 percent splits of the MovieLens 100 K into training and test data. We also split the MovieLens 1M and Epinions into 80 percent/20 percent randomly five times to conduct a five-fold cross-validation and recorded the average values of our metrics.

Table 3 shows the average RMSE and MAE of five-fold cross-validation for all baseline algorithms and DecRec and the smallest RMSE and MAE values for each column is indicated by boldface. The results suggested that among neighbor-based approaches, I-KNN is the best-performing algorithm for MovieLens 1 M and 100 K and U-KNN is the second best-performing one. That is because of the fact that the similarity between movies (genre, director, etc) and similarity between taste of users, play an important role in prediction accuracy. I-KNN and U-I-B outperformed other neighbor based methods on Epinions in respect to RMSE measures while BPSO and I-KNN outperformed others in respect to MAE measures. Similarity of items, and having the same pattern of ratings for similar items, on Epinions helped I-KNN's results perform better than other neighbor based methods.

[5] Some baseline algorithms used for our experiments are implemented in the MyMedia recommender framework [18].

[6] grouplens.org/

TABLE 3
Results of Scenario I on MovieLens 100 K and 1 M and Epinions

	Algorithms	Hyperparameters	MovieLens 100 K		MovieLens 1 M		Epinions	
			RMSE	MAE	RMSE	MAE	RMSE	MAE
Neighbor-Based Methods	GA	—	1.1190	0.9399	1.116	0.9327	1.1692	0.8878
	I-A	—	1.0220	0.8159	0.9759	0.7790	1.0695	0.8140
	U-A	—	1.0390	0.8350	1.034	0.8272	1.1276	0.8769
	U-KNN	$k = 80$	0.9355	0.7398	0.8952	0.7030	2.3999	2.2229
	I-KNN	$k = 80$, sh $= 10$, $\lambda_u = 25$, $\lambda_v = 10$	0.9241	0.7270	0.8711	0.6830	1.0279	0.6993
	U-I-B	$\lambda_u = 5$, $\lambda_v = 2$	0.9419	0.7450	0.9081	0.7190	1.0290	0.8010
	SO	—	0.9397	0.7403	0.9020	0.7120	1.0865	0.7067
	BPSO	—	0.9744	0.7482	0.9390	0.7199	1.0449	0.6813
	CC	$C_i = 3$, $C_u = 3$, $T = 30$	0.9559	0.7526	0.9118	0.7134	1.0573	0.7890
Latent Factor Models	SMF	$p = 10$, $\lambda_u = 0.015$, $\lambda_v = 0.015$, $\lambda_b = 0.01$ $\lambda_s = 1$, $\eta = 0.01$, $\eta_b = 1$, $T = 30$	1.0134	0.7884	1.2284	0.9315	1.1224	0.8436
	FWMF	$p = 5$, $T = 5$, sh $= 150$	0.9212	0.7252	0.8601	0.6730	1.5090	1.0597
	BMF	$p = 160$, $\lambda_b = 0.003$, $\eta = 0.07$, $T = 100$ $\lambda_u = 0.08$, $\lambda_v = 0.1$	0.9104	0.7194	0.8540	0.6760	1.0240	0.7918
	MF	$p = 10$, $T = 75$, $\lambda_g = 0.05$, $\eta = 0.005$	0.9133	0.7245	0.8570	0.6751	1.0908	0.8372
	KMF	$\sigma_r = 0.4$, D $= 10$, $\eta = 0.003$, $\gamma = 0.1$	0.7947	0.6893	0.7492	**0.6514**	0.9015	0.7873
	LFLLM	$p = 10$, $\lambda_b = 0.01$, $\lambda_u = 0.015$, $\lambda_v = 0.015$ $\eta = 0.01$, $T = 30$, $\eta_b = 1$	0.9550	0.7617	0.9012	0.7082	1.2891	1.0386
	SI-AFM	$\eta_b = 0.7$, $\lambda_g = 0.015$, $\eta = 0.001$, $p = 10$ $\lambda_b = 0.33$, $T = 1$	0.9568	0.7628	1.035	0.8488	1.1534	0.8816
	SU-AFM	$\eta_b = 0.7$,$\lambda_g = 0.015$ $\lambda_b = 0.33$, $T = 1$, $\eta = 0.001$, $p = 10$	0.9569	0.7634	0.9062	0.7189	1.069	0.8398
	SCAFM	—	0.9499	0.7559	0.9121	0.7239	1.0600	0.8312
	SVDPP	$\eta_b = 0.07$,$\lambda_g = 1$ $\lambda_b = 0.005$, $p = 50$, $\eta = 0.01$, $T = 50$	0.9065	0.7135	0.8510	0.6680	1.0550	0.8220
	SSVDPP	$\eta_b = 0.7$, $T = 30$, $p = 10$,$\lambda_g = 0.015$, $\eta = 0.001$, $\lambda_b = 0.33$	1.185	0.9147	0.9402	0.7352	1.3328	0.9022
	RS	—	1.6960	1.3860	1.7070	1.3940	1.9096	1.5789
	DecRec	$r = 10$	**0.7002**	**0.6628**	**0.7157**	0.6721	**0.7157**	**0.6796**

Table 3 also shows the comparison between latent factor methods in which DecRec (KMF) algorithm achieved the first (second) best performance of RMSE and MAE for MovieLens 100 K and RMSE for MovieLens 1 M and achieved the second (first) best performance of MAE for MovieLens 1 M. On Epinions, too, DecRec outperformed other latent factor methods.

Table 3 clearly shows that DecRec achieved the best performance for all datasets among all methods of both categories, latent factor and neighbor based methods (except for MAE on MovieLens 1 M), confirming the performance advantage of DecRec over all baseline algorithms. Hence, our proposed decoupled method by incorporating the auxiliary information, reveals the need for preventing error propagation along with using side information to obtain more accurate predictions.

6.7 Existing Users/Cold-Start Items

To simulate cold-start item problems (scenario II), we divided the items into two disjoint training and test subsets. We used 80 percent of the items as existing items for training and the remaining 20 percent as cold-start items for testing.

As we mentioned earlier, our general framework can be employed to deal with link prediction challenges as well. In order to kill two birds with one stone, we selected the NIPS dataset to not only simulate the cold-start item scenario, but also to show the results of DecRec for link prediction challenge. NIPS has rich side information for the items (papers) and shows the relationship (0 or 1) between papers and authors. Since in NIPS the values are either 0 or 1, predicting the authors of new papers can be perceived as a link prediction problem.

We compared our algorithm with four competitive recommendation methods: CBF, KMF, LCE and LCE-NL on NIPS dataset. Table 4 shows the average RMSE and MAE of five-fold cross-validation for these algorithms for cold-start item scenario. The parameter[7] setting of each algorithm is also given in the Table 4 for reproducing the experiments.

The results indicate that DecRec is the best performing algorithm among all baseline algorithms in cold-start item scenario in respect to RMSE, MAE and NDCG. Having the highest NDCG value among all competitive algorithms shows that DecRec can present the top-ranked items to users better than other algorithms. DecRec has also the lowest RMSE and MAE value from which we can conclude that predicting the ratings for cold-start items is more accurate. Hence, DecRec can better suggest the top-ranked cold-start items to users with higher accuracy. Table 4 also shows the running time of the algorithms. Here CBF takes the least time to generate its results due to the fact that it only is required to build the user profiles. It is worth mentioning that KPMF and DecRec are comparably fast, but LCE and LCE-NL take much more time due to their convergence conditions.

[7] The details of parameters are explained in [18].

TABLE 4
Results of All Three Cold-Start Scenarios on Real Datasets

Datasets	Method	Hyperparameters	Measures			
			NDCG@k	RMSE	MAE	Time(s)
NIPS *Cold-start item*	CBF	—	0.3861	0.7943	0.8881	**0.17597**
	LCE	$k = 500, \lambda_g = 0.5, \varepsilon = 0.001, T_m = 500, \beta = 0.05$	0.4240	0.7692	0.8675	709.869
	LCE-NL	$k = 500, \lambda_g = 0.5, \varepsilon = 0.001, T_m = 500, \beta = 0$	0.4186	0.7532	0.8562	1823.48
	KMF	$\sigma_r = 0.4$, D $= 10, \eta = 0.003, \gamma = 0.1$	0.1415	0.8804	0.9196	19.5413
	DecRec	$r = 1000$	**0.4626**	**0.5111**	**0.6805**	23.8410
Epinions *Cold-start user*	CBF	—	0.2201	0.6644	0.7741	**4.4800**
	LCE	$k = 500, \lambda_g = 0.5, \varepsilon = 0.001, T_m = 500, \beta = 0.05$	0.2327	0.6713	0.7786	1067.11
	LCE-NL	$k = 500, \lambda_g = 0.5, \varepsilon = 0.001, T_m = 500, \beta = 0$	0.2319	0.6712	0.7785	11969.9
	KMF	$\sigma_r = 0.4$, D $= 10, \eta = 0.003, \gamma = 0.1$	0.2084	0.8522	0.8882	1196.32
	DecRec	$r = 1063$	**0.2343**	**0.6618**	**0.7716**	144.660
MovieLens 100K *Cold-start user & item*	RS	—	0.1022	1.1981	0.9782	**0.0131**
	KMF	$\sigma_r = 0.4$, D $= 10, \eta = 0.003, \gamma = 0.1$	0.2423	0.9823	0.8730	11.540
	ELCE	$k = 500, \lambda_g = 0.5, \varepsilon = 0.001, T_m = 500, \beta = 0$	0.2681	0.8934	0.7626	16.4683
	DecRec	$r = 100$	**0.2641**	**0.8672**	**0.7230**	4.8729
MovieLens 1M *Cold-start user & item*	RS	—	0.0652	1.3820	0.9326	**0.0682**
	KMF	$\sigma_r = 0.4$, D $= 10, \eta = 0.003, \gamma = 0.1$	0.1834	0.9730	0.8442	63.732
	ELCE	$k = 500, \lambda_g = 0.5, \varepsilon = 0.001, T_m = 500, \beta = 0$	0.2662	0.8849	0.7684	166.201
	DecRec	$r = 100$	**0.2783**	**0.8524**	**0.7162**	10.578

6.8 Cold-Start Users/Existing Items

To simulate cold-start user problems (scenario III), we divided the users into two disjoint subsets of training and test. We used 80 percent of the users for training and the remaining 20 percent for testing of the cold-start user scenario. To show the relative results of DecRec, we compare it with competitive algorithms: CBF, LCE, LCE-NL and KPMF.

Since Epinions has the trust network among the users, which is a useful side information, we chose it to simulate this scenario. Table 4 shows the averaged (five-fold cross validation) performance results of the mentioned algorithms and it clearly shows that DecRec achieved the best performance of NDCG, RMSE and MAE on Epinions dataset.

Considering the NDCG values, DecRec outperforms all other methods while NDCG of LCE and LCE-NL are still close. DecRec, LCE and LCE-NL also outperform other methods in respect to RMSE and MAE. In this case, while DecRec and LCE and LCE-NL behave similarly, the major difference between DecRec and LCE(-NL) is the running time where CBF outperforms DecRec and DecRec outperforms the rest of methods. Though there are no major differences between the values of these metrics for DecRec and others, the consistency in outperforming other competitive methods on both cold-start user and item scenarios confirms the stability and performance advantage of DecRec over state-of-the-art algorithms.

Again, as CBF is only required to build the user profile, it has the shortest running time. Our algorithm has the second fastest running time, but it should be noted that the other algorithms take substantially longer to execute than either DecRec or CBF.

6.9 Cold-Start Users/Cold-Start Items

To simulate handling both cold-start users and items (scenario IV), we randomly selected 20 percent of the users and 20 percent of the items as cold-start users and items, respectively. In this scenario we tried to predict the ratings of cold-start users on cold-start items. Since there are no historical ratings for either users or items, to show the results of DecRec on this challenging scenario, we compared the results of DecRec with only RS, KMF, and ELCE. We did not include LCE, LCE-NL and CBF as they are not applicable in this scenario.

Table 4 shows the results of applying RS, KMF, ELCE and DecRec algorithms on MovieLens 100 K and 1 M. On both datasets, DecRec outperformed other baselines, followed by ELCE, which is a collaborative factorization method. RS generally performs worse than the other algorithms in cold-start users/items scenarios, which is a common problem in newly launched websites; and confirms that it is necessary to carefully use similarity information of users and items to have a more accurate recommendations.

7 Conclusions

We have proposed a novel factorization model, dubbed as DecRec, that explicitly exploits the similarity information about users and items to alleviate cold-start problems. Two key features of DecRec are the completion of a sub-matrix of the rating matrix, which is generated by excluding the cold-start users and items from the set of users and items, and transduction of knowledge from recovered sub-matrix of existing users and items to those of cold-start. In particular, DecRec decouples the completion from the knowledge transduction, which prevents the error propagation of completion and transduction. We also provide a theoretical performance guarantees on the estimation error of DecRec while most of the existing methods do not provide any theoretical support.

Experimental results on real datasets clearly indicated the performance advantage of DecRec over all competing methods not only in existing user/item scenario, but also in all three cold-start scenarios, particularly the cold-start user and item that is the most challenging scenario in recommendation systems.

Acknowledgments

The authors thank the anonymous reviewers and editor-in-chief for their constructive comments which led to the overall improvement of this paper. They also acknowledge valuable technical discussions with Mehrdad Mahdavi during this work. Iman Barjasteh and Rana Forsati contributed equally to this work.

References

[1] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert, "A new approach to collaborative filtering: Operator estimation with spectral regularization," *J. Mach. Learning Res.*, vol. 10, pp. 803–826, 2009.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[3] I. Barjasteh, R. Forsati, F. Masrour, A.-H. Esfahanian, and H. Radha, "Cold-start item and user recommendation with decoupled completion and transduction," in *Proc. 9th ACM Conf. Recommender Syst.*, 2015, pp. 91–98.

[4] J. Basilico and T. Hofmann, "Unifying collaborative and content-based filtering," in *21st Int. Conf. Mach. Learning*, 2004, p. 9.

[5] R. M. Bell and Y. Koren. "Lessons from the Netflix prize challenge,"*SIGKDD Explorations Newslett.*, vol. 9, no. 2, pp. 75–79, 2007.

[6] D. Billsus and M. J. Pazzani, "User modeling for adaptive news access," *User Model. User-Adapted Interaction*, vol. 10, nos. 2–3, pp. 147–180, 2000.

[7] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[8] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[9] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, "Combining content-based and collaborative filters in an online newspaper," in *Proc. ACM SIGIR Workshop Recommender Syst.*, vol. 60, 1999, pp. 1–12.

[10] G. Contardo, L. Denoyer, and T. Artieres, "Representation learning for cold-start recommendation," *arXiv preprint arXiv:1412.7156*, pp. 1–10, 2014.

[11] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learning Res.*, vol. 7, pp. 1–30, 2006.

[12] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer, "The yahoo! music dataset and Kdd-cup'11," *J. Mach. Learning Res. KDD Cup*, vol. 18, pp. 8–18, 2012.

[13] A. Elbadrawy and G. Karypis, "User-specific feature-based similarity models for top-n recommendation of new items," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 6, no. 3, p. 33, 2015.

[14] R. Forsati, I. Barjasteh, F. Masrour, A.-H. Esfahanian, and H. Radha, "Pushtrust: An efficient recommendation algorithm by leveraging trust and distrust relations," in *Proc. 9th ACM Conf. Recommender Syst.*, 2015, pp. 51–58.

[15] R. Forsati, M. Mahdavi, M. Shamsfard, and M. Sarwat, "Matrix factorization with explicit trust and distrust side information for improved social recommendation," *ACM Trans. Inform. Syst.*, vol. 32, no. 4, pp. 17, 2014.

[16] R. Forsati, A. Moayedikia, and M. Shamsfard, "An effective web page recommender using binary data clustering," *Inform. Retrieval J.*, vol. 18, no. 3, pp. 167–214, 2015.

[17] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, "Learning attribute-to-feature mappings for cold-start recommendations,," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 176–185.

[18] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "MyMediaLite: A free recommender system library," in *Proc. 5th ACM Conf. Recommender Syst.*, 2011, pp. 305–308.

[19] 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.

[20] A. Gittens, "The spectral norm error of the naive nystrom extension," *arXiv preprint arXiv:1110.5305*, pp. 1–9, 2011.

[21] Q. Gu and J. Zhou, "Learning the shared subspace for multi-task clustering and transductive transfer classification," in *Proc. 9th Int. Conf. Data Mining*, 2009, pp. 159–168.

[22] A. Gunawardana and C. Meek, "A unified approach to building hybrid recommender systems," in *Proc. 3rd ACM, Recommender Syst.*, 2009, pp. 117–124.

[23] S. K. Gupta, D. Phung, B. Adams, T. Tran, and S. Venkatesh, "Nonnegative shared subspace learning and its application to social media retrieval," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 1169–1178.

[24] S. K. Gupta, D. Phung, B. Adams, and S. Venkatesh, "Regularized nonnegative shared subspace learning," *Data Mining Knowl. Discovery*, vol. 26, no. 1, pp. 57–97, 2013.

[25] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *J. ACM Trans. Inform. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[26] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 135–142.

[27] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 426–434.

[28] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Trans. Knowl. Discovery Data*, vol. 4, no. 1, pp. 1, 2010.

[29] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[30] D. Lemire and A. Maclachlan, "Slope one predictors for online rating-based collaborative filtering," in *SDM*, vol. 5, pp. 1–5. SIAM, 2005.

[31] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2065–2073, 2014.

[32] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua, "Addressing cold-start in APP recommendation: Latent user models constructed from twitter followers," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 2013, pp. 283–292.

[33] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend," in *Proc. ACM Conf. Recommender Syst.*, 2014, pp. 105–112.

[34] J. Liu, C. Wu, and W. Liu, "Bayesian probabilistic matrix factorization with social relations and item contents for recommendation," *Decision Support Syst.*, vol. 55, no. 3, pp. 838–850, 2013.

[35] N. N. Liu, X. Meng, C. Liu, and Q. Yang, "Wisdom of the better few: Cold start recommendation via representative based rating elicitation," in *Proc. ACM Conf. Recommender Syst.*, 2011, pp. 37–44.

[36] M. Long, J. Wang, G. Ding, W. Cheng, X. Zhang, and W. Wang, "Dual transfer learning," in *Proc. SIAM Int. Conf. Data Mining*, 2012, pp. 540–551.

[37] H. Ma, H. Yang, M. R. Lyu, and I. King, "SOREC: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. Inform. Knowl. Manage.*, 2008, pp. 931–940.

[38] F. Masrour, I. Barjasteh, R. Forsati, A.-H. Esfahanian, and R. Hayder, "Network completion with node similarity: A matrix completion approach with provable guarantees," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, 2015, pp. 302–307.

[39] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Proc. 18th Nat. Conf. Artif. Intell.*, 2002, pp. 187–192.

[40] A. K. Menon, K.-P. Chitrapura, S. Garg, D. Agarwal, and N. Kota, "Response prediction using collaborative filtering with hierarchies and side-information," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 141–149.

[41] A. K. Menon and C. Elkan, "A log-linear model with latent features for dyadic prediction," in *Proc. 10th IEEE Int. Conf. Data Mining*, 2010, pp. 364–373.

[42] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inform. Process. Syst.*, 2007, pp. 1257–1264.

[43] U. Ocepek, J. Rugelj, and Z. Bosnić, "Improving matrix factorization recommendations for examples in cold start," *J. Expert Syst. Appl.*, 2015, pp. 6784–6794.

[44] S.-T. Park and W. Chu, "Pairwise preference regression for cold-start recommendation," in *Proc. ACM Conf. Recommender Syst.*, 2009, pp. 21–28.

[45] S.-T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste, "Naïve filterbots for robust cold-start recommendations," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 699–705.

[46] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. KDD Cup Workshop*, 2007, vol. 2007, pp. 5–8.

[47] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artif. Intell. Rev.*, vol. 13, nos. 5–6, pp. 393–408, 1999.

[48] A. Popescul, D. M. Pennock, and S. Lawrence, "Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments," in *Proc. 17th Conf. Uncertainty Artif. Intell.*, 2001, pp. 437–444.

[49] I. Porteous, A. U. Asuncion, and M. Welling, "Bayesian matrix factorization with side information and dirichlet process mixtures," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 1–6.

[50] B. Recht, "A simpler approach to matrix completion," *J. Mach. Learning Res.*, vol. 12, pp. 3413–3430, 2011.

[51] S. Rendle, "Factorization machines," in *Proc. 10th IEEE Int. Conf. Data Mining*, 2010, pp. 995–1000.

[52] S. Rendle and L. Schmidt-Thieme, "Online-updating regularized kernel matrix factorization models for large-scale recommender systems," in *Proc. ACM Conf. Recommender Syst.*, 2008, pp. 251–258.

[53] J. D. M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proc. 22nd Int. Conf. Mach. Learning*, 2005, pp. 713–719.

[54] S. Roweis. (2002, Jun.). Nips dataset [Online]. Available: http:// www. cs. nyu. edu/~roweis

[55] M. Saveski and A. Mantrach, "Item cold-start recommendations: Learning local collective embeddings," in *Proc. ACM Conf. Recommender Syst.*, 2014, pp. 89–96.

[56] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proc. 25th Annu. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 2002, pp. 253–260.

[57] H. Shan and A. Banerjee, "Generalized probabilistic matrix factorizations for collaborative filtering," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 1025–1030.

[58] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surveys*, vol. 47, no. 1, p. 3, 2014.

[59] Le. H. Son. (2014). Dealing with the new user cold-start problem in recommender systems: A comparative review. *Inform. Syst.* [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306437914001525

[60] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Proc. Adv. Neural Inform. Process. Syst.*, 2004, pp. 1329–1336.

[61] M. Trevisiol, L. M. Aiello, R. Schifanella, and A. Jaimes, "Cold-start news recommendation with domain-dependent browse graph," in *Proc. ACM Conf. Recommender Syst.*, 2014, vol. 14, pp. 81–88.

[62] X. Zhang, J. Cheng, S. Qiu, G. Zhu, and H. Lu, "Dualds: A dual discriminative rating elicitation framework for cold start recommendation," *Knowl.-Based Syst.*, vol. 73, pp. 161–172, 2015.

[63] K. Zhou, S.-H. Yang, and H. Zha, "Functional matrix factorizations for cold-start recommendation," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 2011, pp. 315–324.

[64] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro, "Kernelized probabilistic matrix factorization: Exploiting graphs and side information," in *Proc. SIAM Int. Conf. Data Mining*, 2012, vol. 12, pp. 403–414.

Iman Barjasteh received the BS degree from the Sharif University of Technology, Tehran, Iran, in 2011, and the MS degree from the Michigan State University, East Lansing, MI, in 2014. He is currently working toward the PhD degree at Michigan State University, MI. His research interests include machine learning, data mining with applications, and recommender systems.

Rana Forsati received the PhD degree from Shahid Beheshti University, Tehren, Iran, in 2014. She was a visiting research scholar at the University of Minnesota from 2013-2014. She currently is a postdoc researcher in the Computer Science & Engineering Department, Michigan State University. Her research interests include machine learning, data mining with applications in natural language processing, and recommender systems.

Dennis Ross received the BA degree from Albion College, Albion, MI and the MS degree from Michigan State University, East Lansing, MI, in 2014. He is currently working toward the PhD degree at Michigan State University. His is interested in extremal and algorithmic graph theory. He also works with big data and recommender systems.

Abdol-Hossein Esfahanian received the BS and MS degrees from the University of Michigan, in 1975 and 1977, respectively, and the PhD degree in computer science from the Northwestern University, in 1983. He is an associate professor of computer science and engineering at Michigan State University. He has been conducting research in applied graph theory, computer communications, fault tolerant computing, information technology, and data mining.

Hayder Radha received the MS degree from Purdue University, West Lafayette, IN, in 1986, and the MPhil and PhD degrees from Columbia University, New York, NY, in 1991 and 1993, respectively, all in electrical engineering. He is currently a professor of Electrical and Computer Engineering at MSU, and the director of the Wireless and Video Communications Laboratory. He was with Philips Research, Eindhoven, The Netherlands (1996-2000), where he was a principal member of the research staff and a consulting scientist with the Video Communications Research Department. He has written more than 200 peer-reviewed papers and holds more than 30 patents. His current research interests include compressed sensing, signal processing of network graphs, and analysis of social networks. He is a fellow of the IEEE.