Unveiling Hidden Implicit Similarities for Cross-Domain Recommendation

Quan Do, Wei Liu, *Member, IEEE*, Jin Fan, and Dacheng Tao, *Fellow, IEEE*

Abstract—E-commerce businesses are increasingly dependent on recommendation systems to introduce personalized services and products to targeted customers. Providing effective recommendations requires sufficient knowledge about user preferences and product (item) characteristics. Given the current abundance of available data across domains, achieving a thorough understanding of the relationship between users and items can bring in more collaborative filtering power and lead to a higher recommendation accuracy. However, how to effectively utilize different types of knowledge obtained across domains is still a challenging problem. In this paper, we propose to discover both explicit and implicit similarities from latent factors across domains based on matrix tri-factorization. In our research, common factors in a shared dimension (users or items) of two coupled matrices are discovered, while at the same time, domain-specific factors of the shared dimension are also preserved. We will show that such preservation of both common and domain-specific factors are significantly beneficial to cross-domain recommendations. Moreover, on the non-shared dimension, we propose to use the middle matrix of the tri-factorization to match the unique factors, and align the matched unique factors to transfer cross-domain implicit similarities and thus further improve the recommendation. This research is the first that proposes the transfer of knowledge across the non-shared (non-coupled) dimensions. Validated on real-world datasets, our approach outperforms existing algorithms by more than two times in terms of recommendation accuracy. These empirical results illustrate the potential of utilizing both explicit and implicit similarities for making across-domain recommendations.

Index Terms—Cross-domain learning, recommendation system, matrix factorization

1 Introduction

Product or content providers usually offer a wide variety of products. On the one hand, this massive product selection meets a wide range of different consumer needs and tastes. On the other hand, browsing over a long product list is not a user-friendly task for any consumer to choose products of her preferences. Effective matching between product properties and consumer interests enhances user satisfaction and eventually leads to more sale. Realizing that recommendation systems can be a competitive advantage, lots of providers have been employing them to analyze users' past behaviors to provide personalized product recommendations [1].

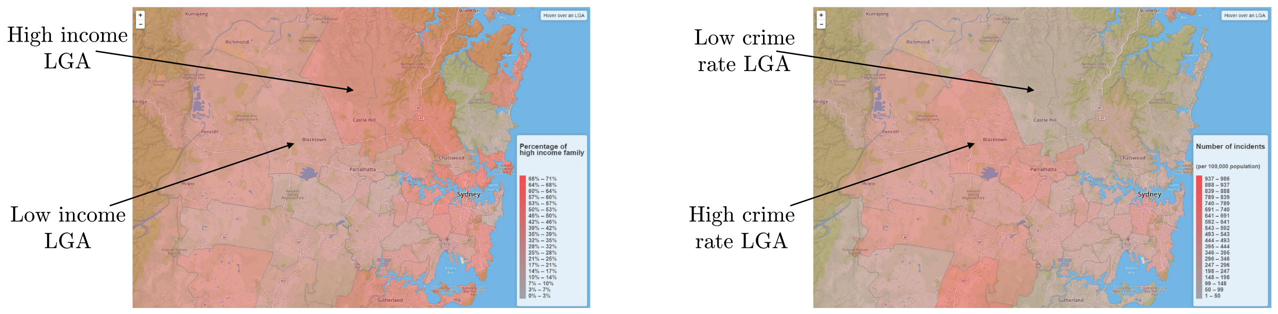
Recommendation systems have gained their importance and popularity among product providers. Two key techniques are widely chosen for developing recommendation systems: content-based approach [2] and collaborative filtering (CF) based approach [3]. The former focuses on information of users or items for making recommendations whereas the latter is based on latent similarities of the user interests and those of the item characteristics for predicting items particular users would be interested. This paper focuses on CF based approach.

Even though CF based methods have shown their effectiveness in providing accurate recommendations, they require adequate ratings to analyze the latent similarities of the users and the items [4]. This thorough understanding is a foundation for achieving reliable recommendation results. The requirement of having sufficient ratings leads to two problems. The first is that newly established businesses need time to collect ratings. The second one is the question raised after they have enough data: once they can achieve a reliable recommendation, will their performance be further improved if they have additional knowledge from external data sources? These two problems are motivations for us to conduct this research.

The above problems are addressable by joint analysis of information coming from different sources [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. Recent revolutions on the Internet have opened a great opportunity for this joint analysis by making many public and highly correlated datasets available. Finding a closely related dataset from another domain is easier than ever. For example, some users buy mobile phones on Amazon, and some other do so at Walmart; they may provide feedback on how much they like them. These ratings from the different users (i.e., Amazon and Walmart users) for the same set of items can be formed two rating matrices coupled in their item dimensions. Thus,

- *Q. Do and W. Liu are with the Advanced Analytics Institute, University of Technology Sydney, Ultimo, NSW 2007, Australia. E-mail: {Quan.Do, Wei.Liu}@uts.edu.au.*
- *J. Fan is with the College of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. E-mail: FanJin@hdu.edu.cn.*
- *D. Tao is with the UBTECH Sydney Artificial Intelligence Center and the School of Information Technologies, the Faculty of Engineering and Information Technologies, University of Sydney, Camperdown, NSW 2006, Australia. E-mail: Dacheng.Tao@Sydney.edu.au.*

Manuscript received 12 Apr. 2018; revised 9 May 2019; accepted 6 June 2019. Date of publication 20 June 2019; date of current version 7 Dec. 2020. (Corresponding author: Quan Do.) Recommended for acceptance by D. J. Cook. Digital Object Identifier no. 10.1109/TKDE.2019.2923904



(a) Percentage of high income family per LGA (b) Number of "break and enter dwelling" incidents per LGA

Fig. 1. (Best viewed in color) Implicit correlations between income and crime rate. Local government areas (LGA) in New South Wales (NSW) state with more high-income families have a lower "break and enter dwelling" incidents. Data in (a) is from Australian Bureau of Statistics and (b) is from the NSW Bureau of Crime Statistics and Research.

they can be jointly used to better understand user preferences and item characteristics. How to joint analysis of this rich information across domains to improve recommendation performance is still a question that needs to be better addressed.

Coupled datasets across domains possess explicit similarities. In case of the aforementioned rating matrices on Amazon and Walmart, they both contain user preferences for the same set of items. Thus, explicit features in the identical item dimension are conventionally used to do coupled learning between datasets [19], [20], [21]. Besides these explicit similarities, we hypothesize that cross-domain datasets also have other implicit correlations in their remaining dimensions. Our intuition comes from consumer segmentation which is a popular concept in marketing. Consumers can be grouped by their behaviors, for example, "tech leaders" is a group of consumers with a strong desire to own new smartphones with the latest technologies or another user group who only changes a new phone when the current one is out of order. Although users on Amazon and Walmart are different, users sharing similar interests may share similar behaviors on the same set of products. We think that this type of correlation also implicitly exists in many other scenarios in the real world, for example, local suburbs with more high-income families may lead to a lower crime rate as shown in Fig. 1, or users with interest on action books may also like related movie genres (e.g., action films). Thus, using both explicit and implicit similarities properly will have a great potential to be applied to many applications.

Different approaches have been proposed to perform a joint analysis of multiple datasets [22]. However, all of the existing algorithms use explicit similarities as a bridge to collaborate among datasets. The popular Collective Matrix Factorization (CMF) [19] jointly analyzes datasets by assuming them to have an identical low-rank factor in their coupled dimension. In this case, the shared identical factor captures the explicit similarities across domains. Li et al. [23] suggest correlated datasets to share explicit hidden rating patterns. The similarities between rating patterns are then transfered from one to another dataset. Gao et al. [24] extend Li's idea to include unique patterns in each dataset. Weike et al. [4] regularizes factors of the target user-by-item matrix with those, called principal coordinates, from user profile and item information matrices. Although these explicit similarities showed their effectiveness in improving recommendation, there are still rich implicit features that were not used but have great potential to further improve the recommendation.

Motivated by this literature gap, we propose a Hidden Implicit Similarities Factorization model (*HISF*) as the first algorithm to utilize both explicit and implicit similarities across datasets. Our idea for explicit similarities, differed from CMF which assumes an identical coupled factor for both datasets, is that cross-domain datasets have their unique knowledge of their domains. Thus, HISF assumes the coupled factors not to be equal, but they contain common parts, which are shared between datasets across domains, and domain-specific parts, which are unique for each domain.

Also, our another contribution is with implicit similarities. The fact that non-coupled dimensions in the aforementioned example contain non-overlapping users prevents a direct knowledge sharing in their non-coupled factors. However, their latent behaviors are correlated and should be shared. These latent behaviors can be captured in low-rank factors by matrix tri-factorization. As factorization is equivalent to spectral clustering [25], different users with similar preferences are grouped in non-coupled user factors. Developing on this concept, we hypothesize that latent clusters in these non-coupled factors may have a close relationship. Therefore, we align correlated clusters in non-coupled factors to be as close as possible. This idea matches the fundamental concept of CF in the sense that similar user groups who rate similarly will continue to do so.

In short, our key contributions are:

1) *Sharing common latent variables while preserving unique ones in the coupled factors* (Section 4.1): We propose coupled factors to include both common and their own unique parts (e.g., common $\mathbf{V}^{(0)}$ and domain-specific $\mathbf{V}^{(1)}$, $\mathbf{V}^{(2)}$ in Fig. 2.) This model better captures the true explicit characteristics of cross-domain datasets.
2) *Aligning implicit similarities in non-coupled factors* (Section 4.2): We introduce a method to leverage implicit similarities. HISF is the first factorization algorithm that utilizes both explicit and implicit similarities across domains for recommendation accuracy improvement.
3) *Introducing an algorithm that optimizes all factors* (Section 4.3): All factors are optimized by an algorithm following alternating least squared approach

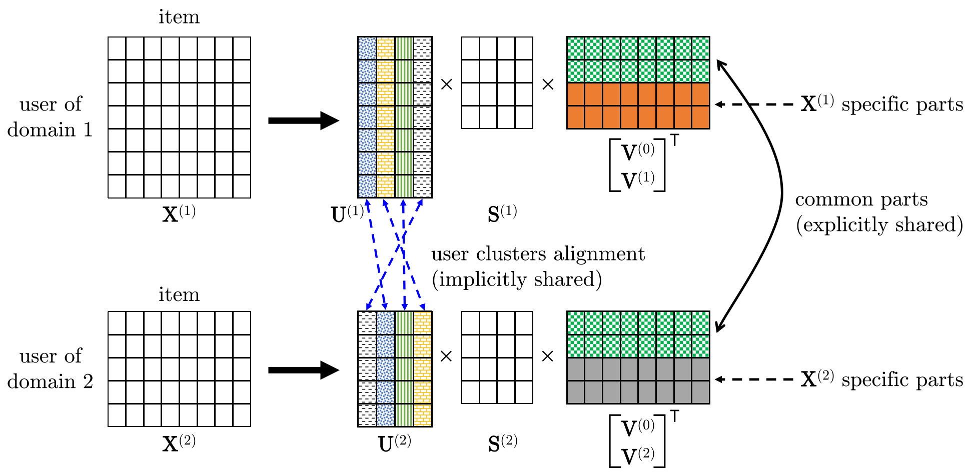

item
user of domain 1
$X^{(1)}$
$U^{(1)}$
$S^{(1)}$
$\begin{bmatrix} V^{(0)} \\ V^{(1)} \end{bmatrix}^T$
$X^{(1)}$ specific parts
common parts (explicitly shared)
user clusters alignment (implicitly shared)
item
user of domain 2
$X^{(2)}$
$U^{(2)}$
$S^{(2)}$
$\begin{bmatrix} V^{(0)} \\ V^{(2)} \end{bmatrix}^T$
$X^{(2)}$ specific parts

Fig. 2. (Best viewed in color) Our Hidden Implicit Similarities Factorization model for two rating matrices $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. They are coupled in their item dimension. Our proposed joint analysis decomposes $\mathbf{X}^{(1)}$ into $\mathbf{U}^{(1)}$, $\mathbf{S}^{(1)}$, common $\mathbf{V}^{(0)}$, and its specific $\mathbf{V}^{(1)}$. At the same time, $\mathbf{X}^{(2)}$ is factorized into $\mathbf{U}^{(2)}$, $\mathbf{S}^{(2)}$, common $\mathbf{V}^{(0)}$, and its specific $\mathbf{V}^{(2)}$. Note that our model also matches clusters of users with similar interests in non-coupled factors $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ by aligning correlated clusters (captured in their columns) as close as possible.

(Algorithm 1). We test it with real-world datasets, and the empirical results suggest our proposed HISF the best choice for joint analysis of datasets across domains (Section 6).

4) *Utilizing knowledge from potentially unlimited number of sources* (Section 5): We propose a generalized model utilizing both similarities from multiple datasets. Our experiments demonstrate its advantages of leveraging correlations from more than two datasets, suggesting the idea can be extended to multi-sources (Section 6).

2 Background and Our Notation

This section provides a background on factorization and a reference for the notation we use.

2.1 Rating Matrix

Recommendation systems are based on past user preferences such as feedbacks users provide to the systems. For example, users of Facebook click ”thumb-up” to like a post. For another example, Amazon’s users rate products from 1 to 5 stars on Amazon website. These kinds of rating are often formed a matrix with users in one dimension and items in the other. Its entries are ratings that users provided. Rating matrices are usually sparse as the users only rate a few items.

TABLE 1
Symbols and Their Descriptions

Symbol	Description
$\mathbf{X}^{(i)}$	Rating matrix from i-th dataset
$\mathbf{U}^{(i)}$	The first dimension factor of $\mathbf{X}^{(i)}$
$\mathbf{V}^{(0)}$	Common parts of the coupled factors
$\mathbf{V}^{(i)}$	Domain-specific parts of the coupled factor of $\mathbf{X}^{(i)}$
$\mathbf{S}^{(i)}$	Weighting factor of $\mathbf{X}^{(i)}$
$\mathbf{A}^{\mathrm{T}}$	Transpose of $\mathbf{A}$
$\mathbf{A}^{+}$	Moore-Penrose pseudo inverse of $\mathbf{A}$
$\mathbf{I}$	The identity matrix
$\lVert\mathbf{A}\rVert$	Frobenius norm
n, m, p	Dimension length
c	Number of common clusters in coupled factors
r	Rank of decomposition
$\Omega_{\mathbf{X}}$	Number of observations in $\mathbf{X}$
$\frac{\partial}{\partial \mathbf{x}}$	Partial derivative with respect to $\mathbf{x}$
$\mathcal{L}$	Loss function
λ	Regularization parameter
$\times$	Multiplication

2.2 Our Notation

A rating matrix from n users for m items is denoted by a boldface capital, e.g., $\mathbf{X}$. Each row of the matrix is a vector of a user’s ratings for all items while each column is a vector of ratings from all users for a specific item. We denote vectors with boldface lowercases, i.e., $\mathbf{u}$. A boldface capital and lowercase with indices in their subscript are respectively used for an entry of a matrix and a vector. Table 1 lists all other symbols we thoroughly use in this paper.

2.3 Matrix Tri-Factorization

Matrix tri-factorization decomposes an input matrix into three low-rank matrices, called factor matrices or simply factors. When the given matrix is complete, decomposing it into factors can be done by Singular Vector Decomposition (SVD). However, where it is incomplete, computing its exact decomposition is an intractable task [26]. Thus, a more efficient and feasible approach is to approximate the incomplete $\mathbf{X}$ of n users by m item as a matrix multiplication of $\mathbf{U} \in \mathbb{R}^{n \times r}$, $\mathbf{S} \in \mathbb{R}^{r \times r}$ and $\mathbf{V}^{\mathrm{T}} \in \mathbb{R}^{r \times m}$:

$$\mathbf{X} \approx \mathbf{U} \times \mathbf{S} \times \mathbf{V}^{\mathrm{T}},$$

where r is the rank of the factorization, $\mathbf{U}^{\mathrm{T}} \times \mathbf{U} = \mathbf{I}$ and $\mathbf{V}^{\mathrm{T}} \times \mathbf{V} = \mathbf{I}$.

In this case, $\mathbf{U}$ is the user factor, $\mathbf{V}$ is the item factor and $\mathbf{S}$ is the weight between $\mathbf{U}$ and $\mathbf{V}$. These factors can be found by solving the optimization of the following:

$$min\mathcal{L} = \left\lVert \mathbf{X} - \mathbf{U} \times \mathbf{S} \times \mathbf{V}^{\mathrm{T}} \right\rVert^2. \quad (1)$$

2.4 Joint Analysis of Different Datasets

Joint analysis of two matrices coupled in one dimension can be done by minimizing a coupled loss function [19]:

$$\mathcal{L} = \left\|\mathbf{X}^{(1)} - \mathbf{U}^{(1)} \times \mathbf{S}^{(1)} \times \mathbf{V}^{\mathrm{T}}\right\|^2 + \left\|\mathbf{X}^{(2)} - \mathbf{U}^{(2)} \times \mathbf{S}^{(2)} \times \mathbf{V}^{\mathrm{T}}\right\|^2,$$

where $\mathbf{V}$ is the common coupled factor for both datasets.

3 Problem Definition

We first define our problem formally here and then discuss our solution in the next section. Suppose $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are two correlated rating matrices from two different domains. Also, suppose that they have an explicitly coupled relationship in their second dimension. For instance, $\mathbf{X}^{(1)}$ is a rating matrix from n users for m items and $\mathbf{X}^{(2)}$ is another one from p users for the same m items. In this case, they have an identical item dimension. This assumption is reasonable as we can find many cross-domain datasets that possess this coupled relationship.

Given $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are coupled in the item dimension, one can combine them (on their coupled dimension) to form a big matrix and decompose it. By doing so, the factor in the coupled dimension can be updated by both datasets. As a result, this combination is the same as CMF [19]. This is probably not a proper approach especially when $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are from two different domains. Their coupled factors have some explicit similarities, yet they also possess their own domain's characteristics. Thus, assuming their coupled factors to be identical potentially loses the unique yet critical knowledge of their own domains. Moreover, $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ also have implicit correlations in their non-coupled user dimension. These similarities, if used properly, may provide rich insights for better understanding their underlying structure. To this end, two questions are raised: how to better use these explicit similarities? And how we can exploit the implicit similarities to fully understand correlations between $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$?

As for the explicit similarities, our key hypothesis is that two cross-domain datasets also have their unique patterns. Our idea is to come up with a way to integrate these domain-specific patterns together with the common parts in the coupled factors. We formulate the coupled item factors not to be identical, but to contain both common and domain-specific parts. Fig. 2 illustrates an example of common $\mathbf{V}^{(0)}$ and domain-specific $\mathbf{V}^{(1)}, \mathbf{V}^{(2)}$ in the coupled factors.

Besides, our another key hypothesis is with implicit similarities. Since factorization is equivalent to spectral clustering [27], we propose non-coupled user factors to share closely related latent user clusters. Even though users in $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are different users, they are clustered by their preferences. Two implicitly related user clusters in the two domains potentially provide some insights to understand their behaviors. We thus align them to be as close as possible so the knowledge between user clusters can be exploited. Fig. 2 demonstrates a case of user clusters alignment between $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}$.

4 Our Hidden Implicit Similarities Factorization Model (HISF)

In this section, we introduce how we exploit explicit and implicit similarities among different datasets. We first explain a case where $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are coupled in their second dimension. An extension to more matrices will be then discussed in the next section. We make two key improvements: integrating both common and domain-specific parts in the item low-rank factors and aligning similar user clusters in the user factors.

4.1 Explicit Similarities - Sharing Common and Preserving Domain-Specific Item Latent Variables in V

Even though $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ contain the same m items, it is unreasonable to force them to share the same low-rank item factors (i.e., $\mathbf{V}^{(1)}$ equals $\mathbf{V}^{(2)}$). The fact that both matrices capture ratings of the same items indicates that they are strongly correlated. Nevertheless, they also have their own unique features of their domain. For this reason, we include both common and unique parts in item factors to better capture correlations of the coupled dimension among different datasets. Thus, the coupled loss function becomes:

$$\mathcal{L} = \left\|\mathbf{X}^{(1)} - \mathbf{U}^{(1)}\mathbf{S}^{(1)}\begin{bmatrix}\mathbf{V}^{(0)}\\ \mathbf{V}^{(1)}\end{bmatrix}^{\mathrm{T}}\right\|^2 + \left\|\mathbf{X}^{(2)} - \mathbf{U}^{(2)}\mathbf{S}^{(2)}\begin{bmatrix}\mathbf{V}^{(0)}\\ \mathbf{V}^{(2)}\end{bmatrix}^{\mathrm{T}}\right\|^2 + \lambda\theta,$$

where $\mathbf{V}^{(0)} \in \mathbb{R}^{\mathrm{m}*\mathrm{c}}$ is the common part, $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)} \in \mathbb{R}^{\mathrm{m}*(\mathrm{r}-\mathrm{c})}$ are domain-specific parts, r is rank of the decomposition, c is the number of common rows such that $1 \leq \mathrm{c} \leq \mathrm{r}$ and θ is a squared L2 regularization term. These common $\mathbf{V}^{(0)}$ and domain-specific $\mathbf{V}^{(1)}$, $\mathbf{V}^{(2)}$ parts in coupled factors are illustrated in Fig. 2.

4.2 Implicit Similarities - Aligning Related User Latent Clusters Across Domains

Besides the explicit similarities in common parts of coupled factors (as described above), the non-coupled dimension of datasets from different sources also possess a strong relationship. Even though users in $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are different users, they may have similar behaviors which can be grouped by latent factors. For instance, different users with similar preferences on sci-fi movies can be grouped as sci-fi fans. This intuition inspires us to hypothesize that non-coupled user factors share closely related latent user clusters. Thus, we first find cluster matchings between user clusters in $\mathbf{U}^{(1)}$ and those in $\mathbf{U}^{(2)}$. Based on these matchings, centroids of their correlated clusters are then regularized to be as close as possible. Fig. 2 demonstrates a case of user clusters alignment between $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}$.

We use a few examples to explain how challenging it is to match user clusters across domains and how clustering can be achieved with matrix factorization. We then introduce our solution that first finds related user clusters across domains and then aligns them together.

4.2.1 Factorization as a Clustering Method

Bauckhage [27] proved that matrix factorization is equivalent to k-means clustering when each row of the factor contains $(k-1)$ 0 elements and only one element that is 1. This condition does not apply in the setting of this paper, but we can assign 1 to the element with the maximum value of each row and 0 to the rest to help guide the clustering.

We now make an example where $\mathbf{X}^{(1)}$ contains ratings of two groups of users with similar preferences as shown in Fig. 3: one group giving brown circle ratings and another one giving green triangle ratings. When $\mathbf{X}^{(1)}$ is factorized, these two user groups are captured in columns of the user factor,

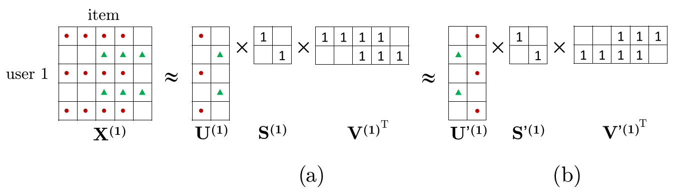


Fig. 3. (Best viewed in color) Matrix factorization as a clustering method. Suppose $\mathbf{X}^{(1)}$ contains ratings of users for items and there are two user groups represented by their rating similarities (brown circles and green triangles). When $\mathbf{X}^{(1)}$ is factorized with matrix tri-factorization, these two user groups are captured in columns of the user factor. Two possible cases can happen: (a) users with brown circles are in the first column and those with green triangles are in the second column of $\mathbf{U}^{(1)}$; or (b) users with brown circles are in the second column and those with green triangles are in the first column of $\mathbf{U}'^{(1)}$.

i.e., one column of the user factor contains users with brown circles and another one captures users with green triangles. This example illustrates that factorization can cluster users based on their preferences. However, it does not provide any information on whether the first column is the group of users with brown circles or green triangles as both $\mathbf{U}^{(1)}\times\mathbf{S}^{(1)}\times\mathbf{V}^{(1)^{\mathrm{T}}}$ and $\mathbf{U}'^{(1)}\times\mathbf{S}'^{(1)}\times\mathbf{V}'^{(1)^{\mathrm{T}}}$ are equally good solutions.

4.2.2 A Challenge to Align Similar Clusters Across Domains

In case there is another dataset from a different domain $\mathbf{X}^{(2)}$ having ratings from different users on the same set of items as shown in Fig. 4. By factorizing $\mathbf{X}^{(2)}$, we can group users with blue circle and those with yellow triangle preferences in columns of user factor $\mathbf{U}^{(2)}$ or $\mathbf{U}'^{(2)}$.

Although the behaviors of users in $\mathbf{X}^{(1)}$ and those in $\mathbf{X}^{(2)}$ are highly correlated, finding a match between user clusters in $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ is not an obvious task because of two reasons. First, users in $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are different. It means that we do not know about the users nor their order in the rating matrices. Second, as alternating least square method [28] is often used to find the factors, factors are initiated randomly. Therefore, a particular user cluster may never be fixed in a particular column of the user factor. In other words, we have no information of whether users with brown circles are to be captured in the first or the second column of the user factor of $\mathbf{X}^{(1)}$. In the same way, users with blue circles can be captured in the first or the second column of the user factor of $\mathbf{X}^{(2)}$. Therefore, there are four possible cases for matching user clusters of $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ as demonstrated in Fig. 5. In the event the input datasets have N clusters, the situation is more complex as there will be N^2 possible cases.

4.2.3 Matching Correlated Clusters Across Domains

To find matches of users clusters across domains, we revisit our intuition about similar user groups. We assume that users

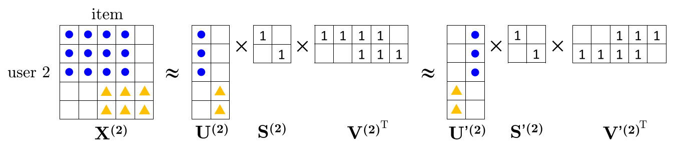


Fig. 4. (Best viewed in color) Suppose $\mathbf{X}^{(2)}$ is obtained from another domain where different users rate the same set of items as that of $\mathbf{X}^{(1)}$. $\mathbf{X}^{(2)}$ also has two user groups having similar preferences with those in $\mathbf{X}^{(1)}$: circle and triangle preferences.

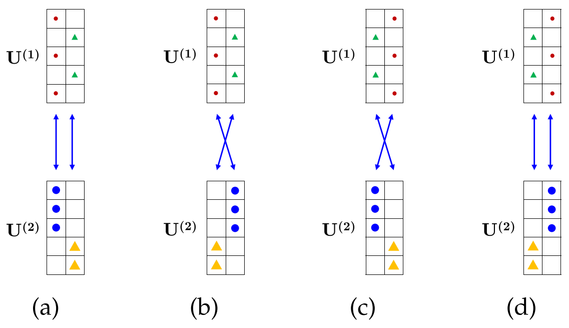


Fig. 5. (Best viewed in color) User clusters' alignment between $\mathbf{X}^{(1)}$ in Fig. 3 and $\mathbf{X}^{(2)}$ in Fig. 4. There are four possible cases: in (a) and (d) the user cluster in the first column of $\mathbf{U}^{(1)}$ matches the user cluster in the first column of $\mathbf{U}^{(2)}$ and the user cluster in the second column of $\mathbf{U}^{(1)}$ matches the user cluster in the second column of $\mathbf{U}^{(2)}$; in (b) and (c) the user cluster in the first column of $\mathbf{U}^{(1)}$ matches the user cluster in the second column of $\mathbf{U}^{(2)}$ and the user cluster in the second column of $\mathbf{U}^{(1)}$ matches the user cluster in the first column of $\mathbf{U}^{(2)}$. These show the challenge to determine how clusters in $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are aligned at each iteration.

have similar behavior on the common item set. When we extract user clusters by matrix tri-factorization as discussed in Section 4.2.1, item factors $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$ have common and different columns. These columns of $\mathbf{V}$ define new coordinates of low dimensional (linear transformation) column-wise (item side) information of $\mathbf{X}$. The common columns of $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$ thus provide new coordinates of common item information of $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$; each of new coordinates (each common columns of $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$) defines a cluster of items (linear transformation of the original item side of $\mathbf{X}$) having the same characteristics (e.g., sci-fi movies or comedy movies). The rows of $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ are the values (weights) of different user clusters (captured in columns of $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ respectively) in these common new coordinates. If a user cluster in $\mathbf{X}^{(1)}$ and another one in $\mathbf{X}^{(2)}$ are geometrically close in the new coordinates, it indicates these user clusters have similar interests on items of the same characteristics. Hence, we match the $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ weights that are close geometrically in the new coordinates defined by $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$.

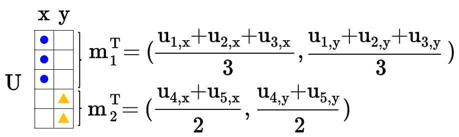
We use this principle to find matches of user clusters across domains. A user cluster $\mathbf{U}^{(1)}_{*,i}$ matches with $\mathbf{U}^{(2)}_{*,j}$ if $\mathbf{S}^{(1)}_{i,*}$ is the closest to $\mathbf{S}^{(2)}_{j,*}$. Thus, we compare the euclidean distances between rows of $\mathbf{S}^{(1)}$ and those of $\mathbf{S}^{(2)}$ to match user clusters whose distances are the shortest.

4.2.4 Aligning Correlated Clusters

Once matches between user clusters across domains are found, we align them to be as close as possible in optimization processes. This alignment is to enforce similar users in latent feature space to rate new items similarly. Here, we propose to regularize the difference between centroids of the closely related user clusters across $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$.

$$\mathcal{L} = \left\| \mathbf{X}^{(1)} - \mathbf{U}^{(1)}\mathbf{S}^{(1)} \begin{bmatrix} \mathbf{V}^{(0)} \\ \mathbf{V}^{(1)} \end{bmatrix}^{\mathrm{T}} \right\|^2 + \left\| \mathbf{X}^{(2)} - \mathbf{U}^{(2)}\mathbf{S}^{(2)} \begin{bmatrix} \mathbf{V}^{(0)} \\ \mathbf{V}^{(2)} \end{bmatrix}^{\mathrm{T}} \right\|^2 + \sum_{\mathrm{l}=1}^{\mathrm{r}} \left\| \mathbf{m}_{\mathrm{l}}^{(1)^{\mathrm{T}}} - \mathbf{m}_{*}^{(2)^{\mathrm{T}}} \right\|^2 + \lambda\theta, \tag{2}$$

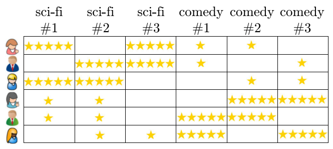
where $\mathbf{m}_{\mathrm{l}}^{(1)^{\mathrm{T}}}$ denotes a row vector of the l-th user cluster's centroid in $\mathbf{U}^{(1)}$ and $\mathbf{m}_{*}^{(2)^{\mathrm{T}}}$ denotes a row vector of the



$$\mathrm{m}_1^{\mathrm{T}} = \left(\frac{\mathrm{u}_{1,\mathrm{x}}+\mathrm{u}_{2,\mathrm{x}}+\mathrm{u}_{3,\mathrm{x}}}{3}, \frac{\mathrm{u}_{1,\mathrm{y}}+\mathrm{u}_{2,\mathrm{y}}+\mathrm{u}_{3,\mathrm{y}}}{3}\right)$$

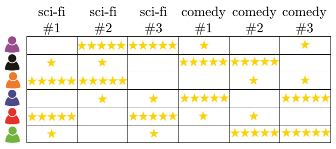
$$\mathrm{m}_2^{\mathrm{T}} = \left(\frac{\mathrm{u}_{4,\mathrm{x}}+\mathrm{u}_{5,\mathrm{x}}}{2}, \frac{\mathrm{u}_{4,\mathrm{y}}+\mathrm{u}_{5,\mathrm{y}}}{2}\right)$$

Fig. 6. An illustration of how centroid of a cluster is computed. $\mathbf{U}$ captures two user clusters: the first three users are in the first cluster and the last two are in the second one. x and y denote the first and the second column of $\mathbf{U}$, respectively.



	sci-fi #1	sci-fi #2	sci-fi #3	comedy #1	comedy #2	comedy #3
	★★★★★		★★★★★	★	★	
		★★★★★	★★★★★	★		★
	★★★★★	★★★★★			★	★
	★	★			★★★★★	★★★★★
	★	★		★★★★★	★★★★★	
		★	★	★★★★★		★★★★★

(a) $\mathbf{X}^{(1)}$



	sci-fi #1	sci-fi #2	sci-fi #3	comedy #1	comedy #2	comedy #3
		★★★★★	★★★★★	★		★
	★	★		★★★★★	★★★★★	
	★★★★★	★★★★★			★	★
		★	★	★★★★★		★★★★★
	★★★★★		★★★★★	★	★	
	★		★		★★★★★	★★★★★

(b) $\mathbf{X}^{(2)}$

Fig. 7. Generated ratings of two domains $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. Blank entries are unobserved ratings. Although users in $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are different, they share implicit similarities in their preferences: those who like sci-fi genre rate sci-fi movies highly and those who are comedy fans do so for comedy movies.

matched user cluster's centroid in $\mathbf{U}^{(2)}$; an example of how $\mathbf{m}_l^{(1)^{\mathrm{T}}}$ and $\mathbf{m}_*^{(2)^{\mathrm{T}}}$ are computed is shown in Fig. 6.

When the user clusters across domains are matched as Fig. 5a and 5d, then:

$$\sum_{l=1}^{r} \left\| \mathbf{m}_l^{(1)^{\mathrm{T}}} - \mathbf{m}_*^{(2)^{\mathrm{T}}} \right\|^2 = \sqrt{(\mathbf{m}_{1,\mathrm{x}}^{(1)} - \mathbf{m}_{1,\mathrm{x}}^{(2)})^2 + (\mathbf{m}_{1,\mathrm{y}}^{(1)} - \mathbf{m}_{1,\mathrm{y}}^{(2)})^2} + \sqrt{(\mathbf{m}_{2,\mathrm{x}}^{(1)} - \mathbf{m}_{2,\mathrm{x}}^{(2)})^2 + (\mathbf{m}_{2,\mathrm{y}}^{(1)} - \mathbf{m}_{2,\mathrm{y}}^{(2)})^2}.$$

However, if the user clusters across domains are matched as Fig. 5b and 5c, the order of columns in $\mathbf{m}_*^{(2)^{\mathrm{T}}}$ has to be adjusted to match with the order of columns $\mathbf{m}_l^{(1)^{\mathrm{T}}}$ such that

$$\sum_{l=1}^{r} \left\| \mathbf{m}_l^{(1)^{\mathrm{T}}} - \mathbf{m}_*^{(2)^{\mathrm{T}}} \right\|^2 = \sqrt{(\mathbf{m}_{1,\mathrm{x}}^{(1)} - \mathbf{m}_{2,\mathrm{y}}^{(2)})^2 + (\mathbf{m}_1^{(1,y)} - \mathbf{m}_{2,\mathrm{x}}^{(2)})^2} + \sqrt{(\mathbf{m}_{2,\mathrm{x}}^{(1)} - \mathbf{m}_{1,\mathrm{y}}^{(2)})^2 + (\mathbf{m}_{2,\mathrm{y}}^{(1)} - \mathbf{m}_{1,\mathrm{x}}^{(2)})^2}.$$

To elaborate how this alignment works, we run Algorithm 1 with a synthetic data in Fig. 7 where $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ each has six unique users who rate the same list of six movies (three sci-fi and three comedy movies). Some users in $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ like sci-fi genre and provide high ratings for sci-fi movies; some other prefer comedy category and rate comedy movies five stars. Thus, users in each domain can be grouped by their preferences: one cluster of sci-fi and another one of comedy fans. Furthermore, each group of users in $\mathbf{X}^{(1)}$ shares its implicit preference with a corresponding group of users in $\mathbf{X}^{(2)}$. As a result, when Algorithm 1 factorizes $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ with rank $r = 2$, each user cluster is captured in a column of the user factors. In addition, their user clusters are aligned so that those with similar preferences are close to each other.

We then plot the resulted user clusters of several iterations into an x-y coordinates in Fig. 8 where the x-axis is the first column of $\mathbf{U}^{(1)}$ and the y-axis is its second column. If the first and the second column of $\mathbf{U}^{(2)}$ match with the first and the second column of $\mathbf{U}^{(1)}$ respectively, x is the first column and y is the second column of $\mathbf{U}^{(2)}$. Otherwise, we

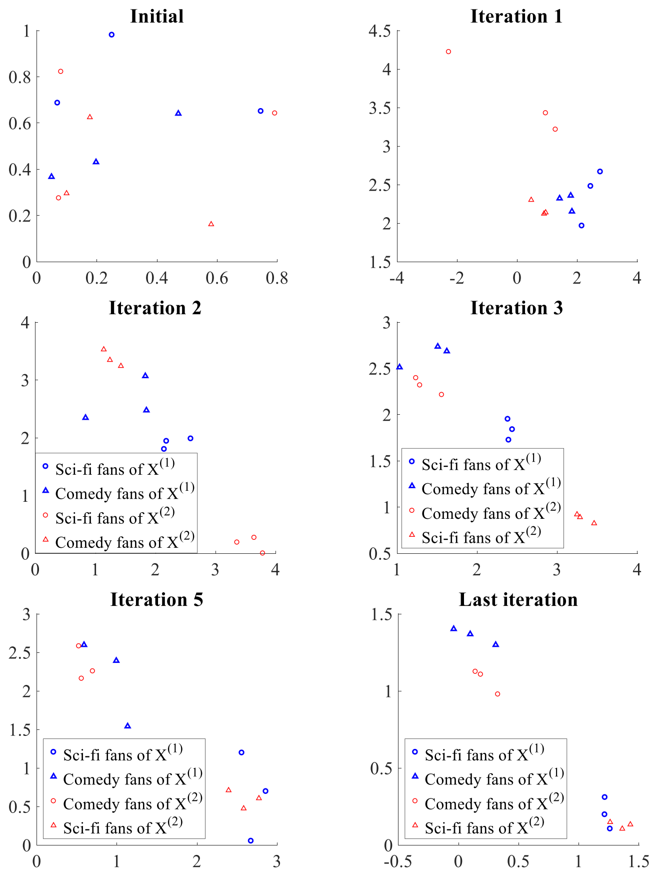


Fig. 8. (Best viewed in color) An illustration of how well our cluster alignment method works. Sci-fi fans and comedy ones from $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ in Fig. 7 are captured in user factors $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$. Circles are users captured in the first column of $\mathbf{U}$, and triangles are for those captured in the second column of $\mathbf{U}$. As $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are randomly initiated, users are scattered. User clusters are formed and aligned over iterations. From iteration 2, two user clusters in $\mathbf{X}^{(1)}$ are clearly separated and gradually aligned with those in $\mathbf{X}^{(2)}$. There is a change in cluster matching in iteration 3: the first column of $\mathbf{U}^{(1)}$ is matched with the second column of $\mathbf{U}^{(2)}$ and vice versa. Since then, centroids of clusters are aligned to be as close as possible from iteration five till the last iteration.

change the order of columns in $\mathbf{U}^{(2)}$ before plotting its users so that matching columns between $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are aligned, i.e., y is the first and x is the second column of $\mathbf{U}^{(2)}$.

Initially, the algorithm randomizes the user factors. So the users are scattered randomly in the x-y space. Iteration 1 starts grouping users to different clusters and aligning them. After iteration 2, users with similar behaviors in $\mathbf{U}^{(1)}$ and those in $\mathbf{U}^{(2)}$ are formed, and we can see the first and the second user clusters of $\mathbf{X}^{(1)}$ are aligned with the first and the second user clusters of $\mathbf{X}^{(2)}$, respectively. In other words, sci-fi fans in the first column of $\mathbf{U}^{(1)}$ (blue circles) are matched with those in the first column of $\mathbf{U}^{(2)}$ (red circles) whereas comedy fans in the second column of $\mathbf{U}^{(1)}$ (blue triangles) are matched with those in the second column of $\mathbf{U}^{(2)}$ (red triangles). Iteration 3 makes a correction on cluster alignment: sci-fi fans, now in the second column of $\mathbf{U}^{(2)}$, are matched with those in the first column of $\mathbf{U}^{(1)}$ (blue circles) whereas comedy fans (in the first column of $\mathbf{U}^{(2)}$ (red circles) are matched with ones in the second column of $\mathbf{U}^{(1)}$ (blue triangles). In this case, the order of columns $\mathbf{U}^{(2)}$ is corrected so that users in $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are in the same coordinates. In the next iterations, centroids of corresponding clusters in $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are adjusted to be close as observed in iteration 5's result or the last iteration's.

Algorithm 1. HISF: Utilizing Both Explicit and Implicit Similarities from two Matrices

```
    Input: $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathcal{E}$
    Output: $\mathbf{U}^{(1)}, \mathbf{S}^{(1)}, \mathbf{V}^{(0)}, \mathbf{V}^{(1)}, \mathbf{U}^{(2)}, \mathbf{S}^{(2)}, \mathbf{V}^{(2)}$
 1: Randomly initialize all factors
 2: Initialize $\mathcal{L}$ by a small number
 3: repeat
 4:     Pre$\mathcal{L} = \mathcal{L}$
 5:     Find matches between clusters in $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$
 6:     Solve $\mathbf{U}^{(1)}$ by (5)
 7:     Solve $\mathbf{U}^{(2)}$ by (6)
 8:     Solve common $\mathbf{V}^{(0)}$ by (7)
 9:     Solve domain-specific $\mathbf{V}^{(1)}$ by (8)
10:     Solve domain-specific $\mathbf{V}^{(2)}$ by (9)
11:     Solve $\mathbf{S}^{(1)}$ by (10)
12:     Solve $\mathbf{S}^{(2)}$ by (11)
13:     Compute $\mathcal{L}$ following (2)
14: until ($\frac{\text{Pre}\mathcal{L}-\mathcal{L}}{\text{Pre}\mathcal{L}} < \mathcal{E}$)
```

4.3 Optimization

Equation (2) is not a convex function. However, it is convex with respect to a factor when the others are fixed. Therefore, we use the alternating least square framework to take turns optimizing one factor in function (2) while fixing the others, as shown in Algorithm 1. Moreover, as data for updating rows of the factors are independent, our algorithm computes factors in a row-wise manner instead of full matrix operations so that the computation can be done in parallel with multiple CPU cores or a distributed system. To this end, we rewrite Equation (2) as the following:

$$\begin{aligned}\mathcal{L} = &\sum_{\mathrm{i,j}}^{\mathrm{n,m}}\left(\mathbf{X}^{(1)}_{\mathrm{i,j}} - \mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}}\mathbf{S}^{(1)}\begin{bmatrix}\mathbf{v}^{(0)}_{\mathrm{j}}\\ \mathbf{v}^{(1)}_{\mathrm{j}}\end{bmatrix}\right)^2\\ &+\sum_{\mathrm{k,j}}^{\mathrm{p,m}}\left(\mathbf{X}^{(2)}_{\mathrm{k,j}} - \mathbf{u}^{(2)^{\mathrm{T}}}_{\mathrm{k}}\mathbf{S}^{(2)}\begin{bmatrix}\mathbf{v}^{(0)}_{\mathrm{j}}\\ \mathbf{v}^{(2)}_{\mathrm{j}}\end{bmatrix}\right)^2 + \sum_{\mathrm{l=1}}^{\mathrm{r}}\left\|\mathbf{m}^{(1)^{\mathrm{T}}}_{\mathrm{l}} - \mathbf{m}^{(2)^{\mathrm{T}}}_{*}\right\|^2 + \lambda\theta.\end{aligned} \tag{3}$$

4.3.1 *Solving $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$*

Let $\mathbf{v}^{(01)}_{\mathrm{j}} = \mathbf{S}^{(1)}[\begin{smallmatrix}\mathbf{v}^{(0)}_{\mathrm{j}}\\ \mathbf{v}^{(1)}_{\mathrm{j}}\end{smallmatrix}]$ and $\mathbf{v}^{(02)}_{\mathrm{j}} = \mathbf{S}^{(2)}[\begin{smallmatrix}\mathbf{v}^{(0)}_{\mathrm{j}}\\ \mathbf{v}^{(2)}_{\mathrm{j}}\end{smallmatrix}]$, then Equation (3) becomes

$$\begin{aligned}\mathcal{L} =&\sum_{\mathrm{i,j}}^{\mathrm{n,m}}\left(\mathbf{X}^{(1)}_{\mathrm{i,j}} - \mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}}\mathbf{v}^{(01)}_{\mathrm{j}}\right)^2 + \sum_{\mathrm{k,j}}^{\mathrm{p,m}}\left(\mathbf{X}^{(2)}_{\mathrm{k,j}} - \mathbf{u}^{(2)^{\mathrm{T}}}_{\mathrm{k}}\mathbf{v}^{(02)}_{\mathrm{j}}\right)^2\\ &+\sum_{\mathrm{l=1}}^{\mathrm{r}}\left\|\mathbf{m}^{(1)^{\mathrm{T}}}_{\mathrm{l}} - \mathbf{m}^{(2)^{\mathrm{T}}}_{*}\right\|^2 + \lambda\theta.\end{aligned} \tag{4}$$

By fixing all $\mathbf{v}_{\mathrm{j}}$, Equation (4) is a convex function with respect to $\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}}$. As a result, $\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}}$ is optimal when the partial derivative of $\mathcal{L}$ with respect to it is set to zero.

$$\begin{aligned}\frac{\delta\mathcal{L}}{\delta\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}}} =& -2\sum_{\mathrm{j}}^{\mathrm{m}}\left(\mathbf{X}^{(1)}_{\mathrm{i,j}} - \mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}}\mathbf{v}^{(01)}_{\mathrm{j}}\right)\mathbf{v}^{(01)^{\mathrm{T}}}_{\mathrm{j}}\\ &+2\left(\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}} - \mathbf{b}^{\mathrm{T}}\right) + 2\lambda\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}}\\ =& -2\mathbf{x}^{(1)^{\mathrm{T}}}_{\mathrm{i},*}\mathbf{V}^{(01)} + 2\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}}\mathbf{V}^{(01)^{\mathrm{T}}}\mathbf{V}^{(01)}\\ &+2\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}} - 2\mathbf{b}^{\mathrm{T}} + 2\lambda\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}},\end{aligned}$$

where $\mathbf{b}^{\mathrm{T}} = -\mathbf{m}^{(1)^{\mathrm{T}}}_{l} + \mathbf{m}^{(2)^{\mathrm{T}}}_{*} + \mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}}$, l is the cluster user i belongs to and $\mathbf{x}^{(1)^{\mathrm{T}}}_{\mathrm{i},*}$ is a row vector of all observed $\mathbf{X}^{(1)}_{\mathrm{i,j}}$, $\forall\mathrm{j} \in [1, \mathrm{m}]$.

By setting $\frac{\delta\mathcal{L}}{\delta\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}}} = 0$, we achieve updating rule for $\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}}$:

$$\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}} = \left(\mathbf{V}^{(01)^{\mathrm{T}}}\mathbf{V}^{(01)} + (\lambda+1)\mathbf{I}\right)^{+}\left(\mathbf{x}^{(1)^{\mathrm{T}}}_{\mathrm{i},*}\mathbf{V}^{(01)} + \mathbf{b}^{\mathrm{T}}\right). \tag{5}$$

In the same way, optimal $\mathbf{u}^{(2)^{\mathrm{T}}}_{\mathrm{k}}$ can be derived from:

$$\mathbf{u}^{(2)^{\mathrm{T}}}_{\mathrm{k}} = \left(\mathbf{V}^{(02)^{\mathrm{T}}}\mathbf{V}^{(02)} + (\lambda+1)\mathbf{I}\right)^{+}\left(\mathbf{x}^{(2)^{\mathrm{T}}}_{\mathrm{k},*}\mathbf{V}^{(02)} + \mathbf{b}^{\mathrm{T}}\right), \tag{6}$$

where $\mathbf{b}^{\mathrm{T}} = -\mathbf{m}^{(1)^{\mathrm{T}}}_{l} + \mathbf{m}^{(2)^{\mathrm{T}}}_{*} + \mathbf{u}^{(2)^{\mathrm{T}}}_{\mathrm{k}}$, l is the cluster matched with the one user k belongs to and $\mathbf{x}^{(2)^{\mathrm{T}}}_{\mathrm{k},*}$ is a row vector of all observed $\mathbf{X}^{(2)}_{\mathrm{k,j}}$, $\forall\mathrm{j} \in [1, \mathrm{m}]$. $\mathbf{I}$ is the identity matrix.

4.3.2 *Solving Common $\mathbf{V}^{(0)}$*

Let $\mathbf{u}^{(1)^{\mathrm{T}}}_{\mathrm{i}} = [\mathbf{u}^{(10)}_{\mathrm{i}}|\mathbf{u}^{(11)}_{\mathrm{i}}]^{\mathrm{T}}\mathbf{S}^{(1)}$ and $\mathbf{u}^{(2)^{\mathrm{T}}}_{\mathrm{k}} = [\mathbf{u}^{(20)}_{\mathrm{k}}|\mathbf{u}^{(22)}_{\mathrm{k}}]^{\mathrm{T}}\mathbf{S}^{(2)}$ where $\mathbf{u}^{(10)^{\mathrm{T}}}_{\mathrm{i}}, \mathbf{u}^{(20)^{\mathrm{T}}}_{\mathrm{k}} \in \mathbb{R}^{1*\mathrm{c}}$ and $\mathbf{u}^{(11)^{\mathrm{T}}}_{\mathrm{i}}, \mathbf{u}^{(22)^{\mathrm{T}}}_{\mathrm{k}} \in \mathbb{R}^{1*\mathrm{r-c}}$, then Equation (2) can be rewritten as:

$$\begin{aligned}\mathcal{L} =&\sum_{\mathrm{i,j}}^{\mathrm{n,m}}\left(\mathbf{X}^{(1)}_{\mathrm{i,j}} - \mathbf{u}^{(10)^{\mathrm{T}}}_{\mathrm{i}}\mathbf{v}^{(0)}_{\mathrm{j}} - \mathbf{u}^{(11)^{\mathrm{T}}}_{\mathrm{i}}\mathbf{v}^{(1)}_{\mathrm{j}}\right)^2\\ &+\sum_{\mathrm{k,j}}^{\mathrm{p,m}}\left(\mathbf{X}^{(2)}_{\mathrm{k,j}} - \mathbf{u}^{(20)^{\mathrm{T}}}_{\mathrm{k}}\mathbf{v}^{(0)}_{\mathrm{j}} - \mathbf{u}^{(22)^{\mathrm{T}}}_{\mathrm{k}}\mathbf{v}^{(2)}_{\mathrm{j}}\right)^2\\ &+\sum_{\mathrm{l=1}}^{\mathrm{r}}\left\|\mathbf{m}^{(1)^{\mathrm{T}}}_{\mathrm{l}} - \mathbf{m}^{(2)^{\mathrm{T}}}_{*}\right\|^2 + \lambda\theta.\end{aligned}$$

Similar to the case of $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$, we now optimize $\mathbf{v}^{(0)}_{\mathrm{j}}$ while fixing all other parameters. Again, by setting the partial derivative of $\mathcal{L}$ with respect to $\mathbf{v}^{(0)}_{\mathrm{j}}$ to zero, we can achieve optimal value of $\mathbf{v}^{(0)}_{\mathrm{j}}$.

$$\begin{aligned}\frac{\delta\mathcal{L}}{\delta\mathbf{v}^{(0)}_{\mathrm{j}}} =& -2\sum_{\mathrm{i}}^{\mathrm{n}}\left(\mathbf{Y}^{(1)}_{\mathrm{i,j}} - \mathbf{u}^{(10)^{\mathrm{T}}}_{\mathrm{i}}\mathbf{v}^{(0)}_{\mathrm{j}}\right)\mathbf{u}^{(10)}_{\mathrm{i}}\\ &-2\sum_{\mathrm{k}}^{\mathrm{p}}\left(\mathbf{Y}^{(2)}_{\mathrm{k,j}} - \mathbf{u}^{(20)^{\mathrm{T}}}_{\mathrm{k}}\mathbf{v}^{(0)}_{\mathrm{j}}\right)\mathbf{u}^{(20)}_{\mathrm{k}} + 2\lambda\mathbf{v}^{(0)}_{\mathrm{j}}\\ =& -2\mathbf{U}^{(1)^{\mathrm{T}}}\mathbf{y}^{(1)}_{*,\mathrm{j}} + 2\mathbf{U}^{(1)^{\mathrm{T}}}\mathbf{U}^{(1)}\mathbf{v}^{(0)}_{\mathrm{j}}\\ &-2\mathbf{U}^{(2)^{\mathrm{T}}}\mathbf{y}^{(2)}_{*,\mathrm{j}} + 2\mathbf{U}^{(2)^{\mathrm{T}}}\mathbf{U}^{(2)}\mathbf{v}^{(0)}_{\mathrm{j}} + 2\lambda\mathbf{v}^{(0)}_{\mathrm{j}},\end{aligned}$$

where $\mathbf{Y}_{i,j}^{(1)} = \mathbf{X}_{i,j}^{(1)} - \mathbf{u}_i^{(11)^T}\mathbf{v}_j^{(1)}$ and $\mathbf{Y}_{k,j}^{(2)} = \mathbf{X}_{k,j}^{(2)} - \mathbf{u}_k^{(22)^T}\mathbf{v}_j^{(2)}$; $\mathbf{y}_{*,j}^{(1)}$ and $\mathbf{y}_{*,j}^{(2)}$ are column vectors of all observed $\mathbf{Y}_{i,j}^{(1)}$, $\forall i \in [1, n]$ and $\mathbf{Y}_{k,j}^{(2)}$, $\forall k \in [1, p]$, respectively.

Our updating rule for $\mathbf{v}_j^{(0)}$ is:

$$\mathbf{v}_j^{(0)} = \left(\mathbf{U}^{(1)^T}\mathbf{U}^{(1)} + \mathbf{U}^{(2)^T}\mathbf{U}^{(2)} + \lambda\mathbf{I}\right)^+ \left(\mathbf{U}^{(1)^T}\mathbf{y}_{*,j}^{(1)} + \mathbf{U}^{(2)^T}\mathbf{y}_{*,j}^{(2)}\right). \tag{7}$$

4.3.3 Solving Domain-Specific $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$

We now perform the similar operations with respect to $\mathbf{v}_j^{(1)}$ and $\mathbf{v}_j^{(2)}$.

$$\frac{\delta\mathcal{L}}{\delta\mathbf{v}_j^{(1)}} = -2\mathbf{U}^{(1)^T}\mathbf{z}_{*,j}^{(1)} + 2\mathbf{U}^{(1)^T}\mathbf{U}^{(1)}\mathbf{v}_j^{(1)} + 2\lambda\mathbf{v}_j^{(1)} = 0,$$

where $\mathbf{Z}_{i,j}^{(1)} = \mathbf{X}_{i,j}^{(1)} - \mathbf{u}_i^{(10)^T}\mathbf{v}_j^{(0)}$ and $\mathbf{Z}_{k,j}^{(2)} = \mathbf{X}_{k,j}^{(2)} - \mathbf{u}_k^{(20)^T}\mathbf{v}_j^{(0)}$; $\mathbf{z}_{*,j}^{(1)}$ and $\mathbf{z}_{*,j}^{(2)}$ are column vectors of all observed $\mathbf{Z}_{i,j}^{(1)}$, $\forall i \in [1, n]$ and $\mathbf{Z}_{k,j}^{(2)}$, $\forall k \in [1, p]$, respectively.

Then we derive the updating rule for $\mathbf{v}_j^{(1)}$:

$$\mathbf{v}_j^{(1)} = \left(\mathbf{U}^{(1)^T}\mathbf{U}^{(1)} + \lambda\mathbf{I}\right)^+ \mathbf{U}^{(1)^T}\mathbf{z}_{*,j}^{(1)}. \tag{8}$$

Analogy to optimizing $\mathbf{v}_j^{(1)}$, optimal $\mathbf{v}_j^{(2)}$ is obtained by:

$$\mathbf{v}_j^{(2)} = \left(\mathbf{U}^{(2)^T}\mathbf{U}^{(2)} + \lambda\mathbf{I}\right)^+ \mathbf{U}^{(2)^T}\mathbf{z}_{*,j}^{(2)}. \tag{9}$$

4.3.4 Solving Weighting Factor $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$

Let

$$\mathbf{s}^{(1)^T} = \left(\mathbf{S}_{1,1}^{(1)}, \mathbf{S}_{2,1}^{(1)}, \ldots, \mathbf{S}_{r,1}^{(1)}, \mathbf{S}_{1,2}^{(1)}, \mathbf{S}_{2,2}^{(1)}, \ldots, \mathbf{S}_{r,2}^{(1)}, \ldots, \mathbf{S}_{1,r}^{(1)}, \mathbf{S}_{2,r}^{(1)}, \ldots, \mathbf{S}_{r,r}^{(1)}\right)$$

and

$$\mathbf{a}^T = \Big(\mathbf{U}_{i,1}^{(1)}\mathbf{V}_{j,1}^{(1)}, \mathbf{U}_{i,2}^{(1)}\mathbf{V}_{j,1}^{(1)}, \ldots, \mathbf{U}_{i,r}^{(1)}\mathbf{V}_{j,1}^{(1)}, \mathbf{U}_{i,1}^{(1)}\mathbf{V}_{j,2}^{(1)}, \mathbf{U}_{i,2}^{(1)}\mathbf{V}_{j,2}^{(1)}, \ldots, \mathbf{U}_{i,r}^{(1)}\mathbf{V}_{j,2}^{(1)},$$
$$\ldots, \mathbf{U}_{i,1}^{(1)}\mathbf{V}_{j,r}^{(1)}, \mathbf{U}_{i,2}^{(1)}\mathbf{V}_{j,r}^{(1)}, \ldots, \mathbf{U}_{i,r}^{(1)}\mathbf{V}_{j,r}^{(1)}\Big).$$

Equation (3) then becomes:

$$\mathcal{L} = \sum_{i,j}^{n,m} \left\|\mathbf{X}_{i,j}^{(1)} - \mathbf{s}^{(1)^T}\mathbf{a}\right\|^2 + \lambda(\|\mathbf{s}^{(1)^T}\|^2) + const$$

where $const$ is the remaining regularization terms.

We can achieve optimal $\mathbf{s}^{(1)^T}$ when $\frac{\partial\mathcal{L}}{\partial\mathbf{s}^{(1)^T}} = 0$

$$\begin{aligned}&\frac{\partial\mathcal{L}}{\partial\mathbf{s}^{(1)^T}} = -2\mathbf{x}_{*,*}^{(1)^T}\mathbf{A} + 2\mathbf{s}^{(1)^T}\mathbf{A}^T\mathbf{A} + 2\lambda\mathbf{s}^{(1)^T} = 0\\ &\Leftrightarrow \mathbf{s}^{(1)^T} = \left(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}\right)^{-1}\left(\mathbf{x}_{*,*}^{(1)^T}\mathbf{A}\right),\end{aligned} \tag{10}$$

where $\mathbf{x}_{*,*}^{(1)^T}$ contains observed $\mathbf{X}_{i,j}^{(1)}$; $\mathbf{x}_{*,*}^{(1)^T} \in \mathbb{R}^{1X\Omega_{\mathbf{X}^{(1)}}}$; $\mathbf{A} \in \mathbb{R}^{\Omega_{\mathbf{X}^{(1)}}X(rXr)}$.

In an analogy way, the update rule for $\mathbf{s}^{(2)^T}$ is:

$$\Leftrightarrow \mathbf{s}^{(2)^T} = \left(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}\right)^{-1}\left(\mathbf{x}_{*,*}^{(2)^T}\mathbf{A}\right). \tag{11}$$

4.4 Complexity Analysis

The computational complexity of Algorithm 1 is

$$O((n + p + m + r)r^3 + (\Omega_{\mathbf{X}^{(1)}} + \Omega_{\mathbf{X}^{(2)}})r^2).$$

Proof. Algorithm 1 computes $\mathbf{U}^{(1)}$ in line 6, $\mathbf{U}^{(2)}$ in line 7, common $\mathbf{V}^{(0)}$ in line 8, domain specific $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$in line 9 and 10, $\mathbf{S}^{(1)}$ in line 11 and $\mathbf{S}^{(2)}$ in line 12. Finding $\mathbf{U}^{(1)}$ involves solving n $\mathbf{u}_i^{(1)^T}$ in Equation 5. Equation 5 prepares $\mathbf{V}^{(01)}$ of size $\frac{\Omega_{\mathbf{X}^{(1)}}}{n}r$ (average), computes $\mathbf{V}^{(01)^T}\mathbf{V}^{(01)}$ and $\mathbf{x}_{i,*}^{(1)^T}\mathbf{V}^{(01)}$, and performs Cholesky decompositions for the pseudo inverse. Preparing $\mathbf{V}^{(01)}$ requires $O(\frac{\Omega_{\mathbf{X}^{(1)}}}{n}r)$ operations while $\mathbf{V}^{(01)^T}\mathbf{V}^{(01)}$ and $\mathbf{x}_{i,*}^{(1)^T}\mathbf{V}^{(01)}$ takes $O(\frac{\Omega_{\mathbf{X}^{(1)}}}{n}r^2)$ each. Cholesky decomposition of $r \times r$ matrices is $O(r^3)$. Thus, solving $\mathbf{U}^{(1)}$ takes $O(nr^3 + \Omega_{\mathbf{X}^{(1)}}r^2)$.

In a similar way, solving $\mathbf{U}^{(2)}$ takes $O(pr^3 + \Omega_{\mathbf{X}^{(2)}}r^2)$; $\mathbf{V}^{(0)}$ takes $O(mc^3 + (\Omega_{\mathbf{X}^{(1)}} + \Omega_{\mathbf{X}^{(2)}})c^2)$, $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$ take $O(m(r-c)^3 + (\Omega_{\mathbf{X}^{(1)}} + \Omega_{\mathbf{X}^{(2)}})(r-c)^2)$, $\mathbf{S}^{(1)}$ takes $O(rr^3 + \Omega_{\mathbf{X}^{(1)}}r^2)$ and $\mathbf{S}^{(2)}$ takes $O(rr^3 + \Omega_{\mathbf{X}^{(2)}}r^2)$. As a result, the complexity of the Algorithm 1 is $O((n + p + m + r)r^3 + (\Omega_{\mathbf{X}^{(1)}} + \Omega_{\mathbf{X}^{(2)}})r^2)$. □

5 Extension to Three or More Matrices

We can find more than two correlated matrices in many cases. For example, Amazon has ratings from different domains, e.g., Books, Movies and TV, Electronics, Digital Musics, etc. These rating matrices may have a close relationship that can help to collaboratively improve recommendation accuracy. In this case, suppose we have three correlated matrices $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$ and $\mathbf{X}^{(3)}$. They are coupled in their second dimension, i.e., $\mathbf{X}^{(1)}$ is a rating matrix from n users for m items, $\mathbf{X}^{(2)}$ is another rating matrix from k users for the same m items and $\mathbf{X}^{(3)}$ is another rating matrix from l users for the same m items. Our idea above can be extended to utilize common parts of coupled factors of the three matrices (explicit similarities) and align clusters among non-coupled factors (implicit similarities). Thus, we propose the following extension for utilizing explicit and implicit similarities among three or more matrices:

$$\mathcal{L} = \sum_{i=1}^{3}\left\|\mathbf{X}^{(i)} - \mathbf{U}^{(i)}\mathbf{S}^{(i)}\begin{bmatrix}\mathbf{V}^{(0)}\\ \mathbf{V}^{(i)}\end{bmatrix}^T\right\|^2 + sim_{\text{implicit}} + \lambda\theta, \tag{12}$$

where sim_{implicit} is the regularization term of implicit similarities across domains and is defined by:

$$\begin{aligned}sim_{\text{implicit}} = &\sum_{l=1}^{r}\left\|\mathbf{m}_l^{(1)^T} - \mathbf{m}_*^{(2)^T}\right\|^2 + \sum_{l=1}^{r}\left\|\mathbf{m}_l^{(2)^T} - \mathbf{m}_*^{(3)^T}\right\|^2\\ &+ \sum_{l=1}^{r}\left\|\mathbf{m}_l^{(3)^T} - \mathbf{m}_*^{(1)^T}\right\|^2.\end{aligned}$$

Updating rules for optimizing $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, $\mathbf{U}^{(3)}$, $\mathbf{S}^{(1)}$, $\mathbf{S}^{(2)}$, $\mathbf{S}^{(3)}$, $\mathbf{V}^{(1)}$, $\mathbf{V}^{(2)}$ and $\mathbf{V}^{(3)}$ can be achieved similarly to the case of two matrices in Section 4.3. Moreover, following the same derivations for common parts of two matrices in Section 4.3.2, we reach an updating rule for $\mathbf{v}_j^{(0)}$, which is the common parts of three matrices:

TABLE 2
Dimension and Number of Known Entries for Training, Validation, and Testing of Census Data on New South Wales (NSW) ($\mathbf{X}^{(1)}$) and Victoria (VIC) ($\mathbf{X}^{(2)}$) States as well as Crime Statistics of NSW ($\mathbf{X}^{(3)}$)

Characteristics	$\mathbf{X}^{(1)}$	$\mathbf{X}^{(2)}$	$\mathbf{X}^{(3)}$
Dimension	$154 \times 7{,}889$	$81 \times 7{,}889$	154×62
Training	91,069	47,900	661
Validation	4,793	2,521	34
Testing	23,965	12,605	173

$$\mathbf{v}_{\mathrm{j}}^{(0)} = \left(\mathbf{U}^{(1)^{\mathrm{T}}}\mathbf{U}^{(1)} + \mathbf{U}^{(2)^{\mathrm{T}}}\mathbf{U}^{(2)} + \mathbf{U}^{(3)^{\mathrm{T}}}\mathbf{U}^{(3)} + \lambda\mathbf{I}\right)^{+} \times \left(\mathbf{U}^{(1)^{\mathrm{T}}}\mathbf{y}_{*,\mathrm{j}}^{(1)} + \mathbf{U}^{(2)^{\mathrm{T}}}\mathbf{y}_{*,\mathrm{j}}^{(2)} + \mathbf{U}^{(3)^{\mathrm{T}}}\mathbf{y}_{*,\mathrm{j}}^{(3)}\right). \tag{13}$$

By optimizing Equation (12), we leverage both explicit similarities (in a form of common $\mathbf{V}^{(0)}$ parts) and implicit similarities (in a form of aligned user groups in $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$) among $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$ and $\mathbf{X}^{(3)}$. Utilization of correlations from four or more matrices can be easily extended in the same way. Therefore, our proposed method does not limit itself to a certain number of correlated matrices.

6 Experiments and Analysis

We evaluate our proposed HISF in comparison with existing algorithms, including CMF [19], CST [4], CBT [23] and CLFM [24]. This evaluation thoroughly studies two test cases: one with two matrices and another one with three matrices. Our goal is to evaluate how well these algorithms suggest unknown information based on the observed cross-domain ratings. For this purpose, we compare them based on the commonly used root mean squared error (RMSE) metric.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{\mathrm{i,j}}^{\mathrm{n,m}}\left(\mathbf{U}_{\mathrm{i}} \times \mathbf{S} \times \mathbf{V}_{\mathrm{j}}^{\mathrm{T}} - \mathbf{X}_{\mathrm{i,j}}\right)^2}{\Omega_{\mathbf{X}}}},$$

where $\Omega_{\mathbf{X}}$ is the number of observations of $\mathbf{X}$.

6.1 Data for the Experiments

We use three publicly available datasets for our experiments. Their characteristics are summarized in Tables 2 and 3.

6.1.1 Dataset #1

Australian Bureau of Statistics (ABS)[1] publishes comprehensive census data of New South Wales (NSW) and Victoria (VIC) states. The dataset for NSW comprises of populations and family profiles of 154 areas, so-called "local government areas" (LGA), and of 81 LGAs in VIC. 7,889 aspects of population and family profile are extracted from the below census categories:

- Rent (weekly) by Landlord Type
- Rent (weekly) by Family Composition for Couple Families
- Rent (weekly) by Family Composition for One Parent Families
- Total Family Income (weekly) by Number of Children for Couple Families
- Total Family Income (weekly) by Number of Children for One Parent Families
- Total Household Income (weekly) by Rent (weekly)
- Family Composition by Mortgage Repayment
- Family Composition by Income Comparison for Parents/Partners
- Family Composition and Social Marital Status by Number of Dependent Children
- Selected Labour Force, Education and Migration Characteristics
- Family Composition and Labour Force Status of Parent(s)/Partners by Total Family Income (weekly)
- Non-School Qualification: Level of Education by Age by Sex
- Non-School Qualification: Field of Study by Age by Sex
- Labour Force Status by Age by Sex
- Industry of Employment by Sex
- Occupation by Sex

We form a matrix $\mathbf{X}^{(1)}$ (LGA by population and family profile) for NSW and another matrix $\mathbf{X}^{(2)}$ for VIC. Values of the matrix are normalized by row. We then randomly select 10% of the data and use its 80 percent for training and 20 percent for testing.

6.1.2 Dataset #2

Bureau of Crime Statistics and Research (BOCSAR)[2] provides a statistical record of criminal incidents within 154 LGAs of New South Wales. There are 62 specific offence categories. We form a matrix $\mathbf{X}^{(3)}$ (LGA by offence categories) whose entries represents how many cases were reported for an offence category in an LGA. Values of the matrix are normalized by row. We randomly select 10 percent of the data for the experiments. Among them, 80 percent of $\mathbf{X}^{(3)}$ are for training and the rest are for testing.

6.1.3 Dataset #3

Three matrices of ratings for books, movies and electronics are extracted from Amazon website [29]. The book data contains ratings from 305,475 users on 888,057 books; the movie data is ratings from the same 305,475 users on 128,097 movies and TV programs; the electronics data is of the same users on 196,894 items. All ratings are from 1 to 5. For this experiment, the data is constructed as the following:

- We first adopt the same sub-sampling approach as in [4] by randomly extracting $10^4 \times 10^4$ dense rating matrices

TABLE 3
Dimension and Number of Known Entries for Training, Validation, and Testing of Amazon Dataset on Books ($\mathbf{X}^{(4)}$), Movies ($\mathbf{X}^{(5)}$) and Electronics ($\mathbf{X}^{(6)}$)

Characteristics	$\mathbf{X}^{(4)}$	$\mathbf{X}^{(5)}$	$\mathbf{X}^{(6)}$
Dimension	$5{,}000 \times 5{,}000$	$5{,}000 \times 5{,}000$	$5{,}000 \times 5{,}000$
Training	158,907	94,665	41,126
Validation	8,363	4,982	2,164
Testing	18,585	11,071	4,809

1. ABS: http://www.abs.gov.au/websitedbs/censushome.nsf/home/datapacks

2. BOCSAR: http://www.bocsar.nsw.gov.au/Pages/bocsar_crime_stats/bocsar_crime_stats.aspx

TABLE 4
Mean and Standard Deviation of Tested RMSE on ABS NSW and ABS VIC Data with Different Algorithms

Rank	Dataset	CMF	CBT	CLFM	CST	HISF1	HISF2
5	ABS NSW $\mathbf{X}^{(1)}$	0.0226 ± 0.0026	0.0839 ± 0.0002	0.0838 ± 0.0002	0.0248 ± 0.0005	0.0183 ± 0.0028	**0.0132** ± 0.0002
	ABS VIC $\mathbf{X}^{(2)}$	0.0364 ± 0.0031	0.0844 ± 0.0003	0.0845 ± 0.0004	0.0271 ± 0.0015	0.0335 ± 0.0091	**0.0266** ± 0.0030
7	ABS NSW $\mathbf{X}^{(1)}$	0.0222 ± 0.0009	0.0836 ± 0.0004	0.0842 ± 0.0006	0.0244 ± 0.0005	0.0192 ± 0.0032	**0.0131** ± 0.0003
	ABS VIC $\mathbf{X}^{(2)}$	0.0428 ± 0.0020	0.0845 ± 0.0004	0.0849 ± 0.0004	0.0288 ± 0.0025	0.0318 ± 0.0038	**0.0239** ± 0.0025
9	ABS NSW $\mathbf{X}^{(1)}$	0.0241 ± 0.0011	0.0841 ± 0.0002	0.0848 ± 0.0009	0.0231 ± 0.0009	0.0192 ± 0.0020	**0.0143** ± 0.0004
	ABS VIC $\mathbf{X}^{(2)}$	0.0476 ± 0.0040	0.0852 ± 0.0003	0.0848 ± 0.0003	0.0283 ± 0.0024	0.0304 ± 0.0036	**0.0221** ± 0.0019
11	ABS NSW $\mathbf{X}^{(1)}$	0.0265 ± 0.0026	0.0846 ± 0.0007	0.0841 ± 0.0007	0.0223 ± 0.0006	0.0186 ± 0.0012	**0.0143** ± 0.0003
	ABS VIC $\mathbf{X}^{(2)}$	0.0501 ± 0.0029	0.0858 ± 0.0005	0.0851 ± 0.0007	0.0267 ± 0.0020	0.0302 ± 0.0025	**0.0242** ± 0.0015
13	ABS NSW $\mathbf{X}^{(1)}$	0.0237 ± 0.0024	0.0851 ± 0.0002	0.0850 ± 0.0005	0.0222 ± 0.0010	0.0177 ± 0.0023	**0.0151** ± 0.0004
	ABS VIC $\mathbf{X}^{(2)}$	0.0489 ± 0.0032	0.0860 ± 0.0003	0.0852 ± 0.0003	0.0290 ± 0.0026	0.0286 ± 0.0038	**0.0227** ± 0.0015
15	ABS NSW $\mathbf{X}^{(1)}$	0.0229 ± 0.0029	0.0853 ± 0.0005	0.0847 ± 0.0005	0.0219 ± 0.0007	0.0180 ± 0.0031	**0.0150** ± 0.0000
	ABS VIC $\mathbf{X}^{(2)}$	0.0514 ± 0.0041	0.0862 ± 0.0002	0.0854 ± 0.0006	0.0285 ± 0.0023	0.0274 ± 0.0054	**0.0215** ± 0.0005
Hotelling's T squared tests		1.15×10^{-6}	3.97×10^{-17}	1.13×10^{-18}	4.91×10^{-8}	-	

For CST, when $\mathbf{X}^{(1)}$ is the target, $\mathbf{X}^{(2)}$ is used as an auxiliary data and vice versa. HISF1 denotes the algorithm with only explicit similarities whereas HISF2 uses both implicit and explicit similarities. Best results for each rank are in bold. The Hotelling's T *squared tests row presents p-value of Hotelling's* T *squared tests between each algorithm and our proposed HISF.*

from these three matrices. Then, we take three sub-matrices of 5,000×5,000 each, as summarized in Table 3. All sub-matrices share the same users, but no common items.

- All ratings are normalized by 5 so that their values are from 0.2 to 1.

6.2 Experimental Settings

We performed factorization with different ranks. Also, each algorithm was run five times. The mean and standard deviation of the results are reported in the next Section. Furthermore, we assume small changes across consecutive iterations indicate an algorithm's convergence. Thus, we stopped the algorithms when changes were less than 10^{-5}.

6.3 Empirical Results

Three scenarios are tested as the following:

6.3.1 Case #1. Latent Demographic Profile Similarities and Latent LGA Groups Similarities can Help to Collaboratively Suggest Unknown Information in These States

We use $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ as described in Section 6.1 which are from different LGAs of two states. Nevertheless, both of them are ratings for the same demographic categories. They share some common explicit demography similarities as well as implicit LGAs' latent ones. We would like to assess how well both explicit similarities in demography dimension and implicit ones in LGA dimension collaboratively suggest unknown information.

Table 4 shows mean and standard deviation of RMSE of all algorithms on tested ABS data for New South Wales ($\mathbf{X}^{(1)}$) and Victoria ($\mathbf{X}^{(2)}$) states. Both CBT and CLFM that assume two states have similar demography patterns in latent sense clearly perform the worst. The results demonstrate that these explicit similarities (in the form of latent demography patterns) do not fully capture the correlation nature of two datasets in this case. Thus, they do not help both CBT and CLFM to improve their performance significantly. CMF applies another approach to take advantages of explicit correlations between NSW state's population and family profile and those of VIC state. Specifically, CMF's assumption on the same population and family profile factor between NSW and VIC helps improve its performance over that of CBT and CLFM. CST allows a more flexible utilization of explicit correlations between NSW's population and family profile and those of VIC state than CMF does. As a result, CST achieves a little bit higher accuracy than CMF in recommending NSW's and VIC's missing information.

Nevertheless, the prediction accuracy can be improved even more as illustrated with our proposed idea of explicit and implicit similarities discovery. Utilizing them helps our proposed HISF to achieve about two times higher accuracy compared with CMF, about up to 47 percent improvement for NSW and up to 25 percent for VIC compared to CST. These impressive results are achieved when the numbers of common columns are 3, 5, 5, 7, 7 and 7 for the decomposition rank of 5, 7, 9, 11, 13 and 15, respectively. It means that common parts together with domain-specific parts better capture the true correlation nature of datasets. These explicit similarities together with implicit similar group alignments allow better knowledge leveraging between datasets, thus, improving recommendation accuracy.

To confirm the statistical significance of our method, we perform Hotelling's T squared tests [30] which is the multivariate version of the t-tests in univariate statistics. Our objective is to validate if our proposed algorithm differs significantly from baselines. We use this multivariate Hotelling's T squared tests here because each population involves observations from two variables: NSW (ABS NSW $\mathbf{X}^{(1)}$) and VIC (ABS VIC $\mathbf{X}^{(2)}$). For testing the null hypothesis that each pair of algorithms (CMF versus HISF, CBT versus HISF, CLFM versus HISF and CST versus HISF) has identical mean RMSE vectors, let

$$\begin{aligned}
&\mathrm{H}_0\text{: population mean RMSEs are identical for all of the variables} \\
&\qquad (\mu_{NSW1} = \mu_{NSW2} \text{ and } \mu_{VIC1} = \mu_{VIC2}) \\
&\mathrm{H}_1\text{: at least one pair of these means is different} \\
&\qquad (\mu_{NSW1} \neq \mu_{NSW2} \text{ or } \mu_{VIC1} \neq \mu_{VIC2})
\end{aligned}$$

TABLE 5
Mean and Standard Deviation of Tested RMSE on ABS NSW Demography and BOCSAR NSW Crime Data with Different Algorithms

Dataset	Rank	CMF	CBT	CLFM	CST	HISF1	HISF2
5	Demography $\mathbf{X}^{(1)}$	0.0209 ± 0.0016	0.0840 ± 0.0001	0.0840 ± 0.0001	0.0304 ± 0.0080	$\mathbf{0.0166} \pm 0.0015$	0.0174 ± 0.0015
	Crime $\mathbf{X}^{(3)}$	0.2796 ± 0.0204	0.3411 ± 0.0035	0.3422 ± 0.0071	0.3216 ± 0.0052	0.3316 ± 0.0175	$\mathbf{0.2697} \pm 0.0073$
7	Demography $\mathbf{X}^{(1)}$	0.0223 ± 0.0024	0.0840 ± 0.0002	0.0855 ± 0.0006	0.0324 ± 0.0040	0.0151 ± 0.0021	$\mathbf{0.0143} \pm 0.0004$
	Crime $\mathbf{X}^{(3)}$	0.2907 ± 0.0265	0.3432 ± 0.0021	0.3912 ± 0.0188	0.3440 ± 0.0055	0.3155 ± 0.0100	$\mathbf{0.2716} \pm 0.0029$
9	Demography $\mathbf{X}^{(1)}$	0.0199 ± 0.0027	0.0838 ± 0.0002	0.0850 ± 0.0008	0.0337 ± 0.0050	0.0152 ± 0.0013	$\mathbf{0.0143} \pm 0.0003$
	Crime $\mathbf{X}^{(3)}$	0.2813 ± 0.0261	0.3562 ± 0.0134	0.3722 ± 0.0249	0.3434 ± 0.0087	0.3192 ± 0.0094	$\mathbf{0.2648} \pm 0.0058$
11	Demography $\mathbf{X}^{(1)}$	0.0212 ± 0.0049	0.0839 ± 0.0001	0.0843 ± 0.0004	0.0402 ± 0.0073	0.0168 ± 0.00147	$\mathbf{0.0146} \pm 0.0003$
	Crime $\mathbf{X}^{(3)}$	0.2689 ± 0.0143	0.3539 ± 0.0061	0.3712 ± 0.0199	0.3495 ± 0.0051	0.3089 ± 0.0118	$\mathbf{0.2618} \pm 0.0012$
13	Demography $\mathbf{X}^{(1)}$	0.0194 ± 0.0022	0.0837 ± 0.0001	0.0837 ± 0.0003	0.0383 ± 0.0113	0.0169 ± 0.0008	$\mathbf{0.0149} \pm 0.0003$
	Crime $\mathbf{X}^{(3)}$	0.2700 ± 0.0150	0.3481 ± 0.0070	0.3500 ± 0.0135	0.3599 ± 0.0024	0.2957 ± 0.0070	$\mathbf{0.2623} \pm 0.0024$
15	Demography $\mathbf{X}^{(1)}$	0.0173 ± 0.0014	0.0835 ± 0.0001	0.0834 ± 0.0002	0.0503 ± 0.0084	0.0172 ± 0.0015	$\mathbf{0.0149} \pm 0.0002$
	Crime $\mathbf{X}^{(3)}$	0.2647 ± 0.0031	0.3485 ± 0.0038	0.3580 ± 0.0099	0.3610 ± 0.0011	0.3114 ± 0.0097	$\mathbf{0.2625} \pm 0.0015$
Hotelling's T squared tests		8.22×10^{-4}	1.37×10^{-15}	2.51×10^{-15}	1.38×10^{-6}	-	

For CST, when $\mathbf{X}^{(1)}$ is the target, $\mathbf{X}^{(3)}$ is used as an auxiliary data and vice versa. HISF1 denotes the algorithm with only explicit similarities whereas HISF2 uses both implicit and explicit similarities. Best results for each rank are in bold. The Hotelling's T *squared tests row presents p-value of Hotelling's* T *squared tests between each algorithm and our proposed HISF.*

Because all p-values are smaller than α (0.05), we reject the null hypothesis. Therefore, the observed difference between the baselines and our proposed algorithm is statistically different.

6.3.2 Case #2. Latent LGA Similarities and Latent Similarities between Demography and Crime can Help to Collaboratively Suggest Unknown Crime and State Information

The advantages of both explicit and implicit similarities are further confirmed in Table 5. In this case, they are applied to other cross domains: ABS NSW demography ($\mathbf{X}^{(1)}$) and NSW Crime ($\mathbf{X}^{(3)}$). These datasets have explicit similarities in their LGA latent factors. At the same time, implicit similarities in demography profile and criminal behaviors are also utilized. Our proposed HISF leveraging both similarities outperforms existing algorithms. It is worth to note here that performance of CST, in this case, is worse than that of CMF (about two times worse for NSW and a bit lower for VIC). The results suggest that flexibility of utilizing explicit similarities does not work here. On the contrary, our proposed idea of utilizing both explicit and implicit similarities eventually achieves more stable and better results (demonstrated in Tables 4 and 5).

We perform similar Hotelling's T squared tests as of the case #1. Again, as all p-values are smaller than α (0.05) here, we reject the null hypothesis. It is therefore convincing to conclude that the mean RMSEs between the baselines and our proposed idea differ significantly.

We also show how HISF works with different values of c parameter in Fig. 9, 10. These figures illustrate the case where decomposition rank equals 11. All the curves of HISF have an identical trend. As the number of common row of explicit similarities (c parameter) increases, recommendation accuracy improves to reach its minimum rate. Then, the performance reduces as c increases. This pattern confirms the significance of preserving domain-specific parts of each dataset. In other words, common and domain-specific parts better capture the correlation nature of datasets across domains, thus, achieve a higher recommendation accuracy. Moreover, when c equals rank 11, HISF and CMF utilize the same explicit similarities (identical coupled factors). Yet, HISF uses an extra correlation from implicit similarities. This additional knowledge enables HISF to achieve a lower RMSE compared to CMF (Fig. 9b). All of these suggest the advantages of both explicit and implicit similarities in joint analysis of two matrices.

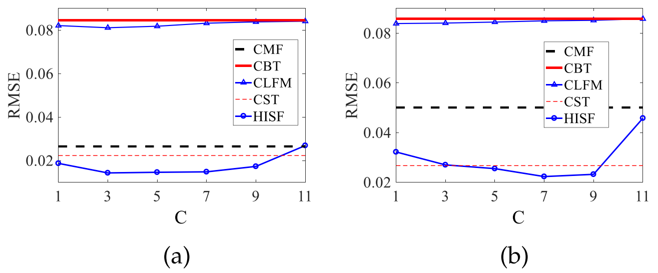


Fig. 9. Tested mean RMSEs under different number of common row c in coupled factor of HISF with Rank=11. (a) Results on ABS NSW dataset. (b) Results on ABS VIC dataset. CMF and CBT do not have c parameter, thus, their results are used as references. Lower is better. RMSE of HISF outperforms its competitors when c equals 7.

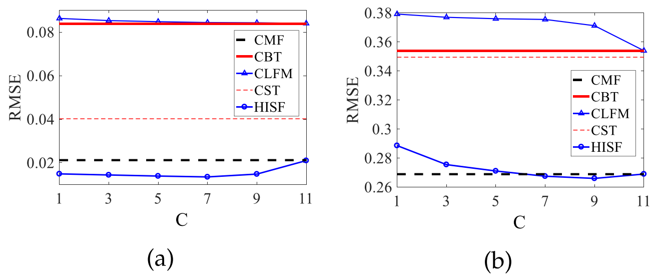


Fig. 10. Tested mean RMSEs under different common row c in coupled factor of HISF with Rank=11. (a) Results on ABS NSW demography. (b) Results on BOCSAR NSW Crime. CMF and CBT do not have c parameter, thus, their results are used as references. Lower is better. RMSE of HISF outperforms its competitors when c equals 9.

TABLE 6
Mean and Standard Deviation of Tested RMSE on Book, Movie, and Electronics Data with Different Algorithms

Rank	Dataset	CMF	CBT	CLFM	HISF-N
5	Books $\mathbf{X}^{(4)}$	0.2169 ± 0.0005	0.2187 ± 0.0026	0.2161 ± 0.0023	$\mathbf{0.1951} \pm 0.0005$
	Movies $\mathbf{X}^{(5)}$	0.2273 ± 0.0022	0.3342 ± 0.0039	0.3474 ± 0.0063	$\mathbf{0.2212} \pm 0.0010$
	Electronics $\mathbf{X}^{(6)}$	$\mathbf{0.2375} \pm 0.0017$	0.4206 ± 0.0036	0.4596 ± 0.0066	0.2642 ± 0.0011
7	Books $\mathbf{X}^{(4)}$	0.2170 ± 0.0009	0.3068 ± 0.0031	0.3059 ± 0.0024	$\mathbf{0.1977} \pm 0.0010$
	Movies $\mathbf{X}^{(5)}$	0.2279 ± 0.0009	0.4368 ± 0.0031	0.4337 ± 0.0055	$\mathbf{0.2213} \pm 0.0007$
	Electronics $\mathbf{X}^{(6)}$	$\mathbf{0.2348} \pm 0.0009$	0.6607 ± 0.0135	0.6389 ± 0.0116	0.2497 ± 0.0016
9	Books $\mathbf{X}^{(4)}$	0.2185 ± 0.0010	0.3163 ± 0.0004	0.3150 ± 0.0026	$\mathbf{0.2001} \pm 0.0005$
	Movies $\mathbf{X}^{(5)}$	0.2302 ± 0.0010	0.4661 ± 0.0046	0.4594 ± 0.0065	$\mathbf{0.2218} \pm 0.0008$
	Electronics $\mathbf{X}^{(6)}$	$\mathbf{0.2354} \pm 0.0019$	0.7098 ± 0.0232	0.6909 ± 0.0175	0.2411 ± 0.0019
11	Books $\mathbf{X}^{(4)}$	0.2219 ± 0.0014	0.3207 ± 0.0044	0.3204 ± 0.0021	$\mathbf{0.2012} \pm 0.0004$
	Movies $\mathbf{X}^{(5)}$	0.2319 ± 0.0021	0.4865 ± 0.0019	0.4795 ± 0.0043	$\mathbf{0.2247} \pm 0.0005$
	Electronics $\mathbf{X}^{(6)}$	$\mathbf{0.2417} \pm 0.0010$	0.7390 ± 0.0090	0.7223 ± 0.0123	0.2420 ± 0.0018
13	Books $\mathbf{X}^{(4)}$	0.2244 ± 0.0018	0.3267 ± 0.0020	0.3291 ± 0.0027	$\mathbf{0.2021} \pm 0.0010$
	Movies $\mathbf{X}^{(5)}$	0.2349 ± 0.0025	0.5014 ± 0.0057	0.4908 ± 0.0028	$\mathbf{0.2263} \pm 0.0009$
	Electronics $\mathbf{X}^{(6)}$	0.2422 ± 0.0012	0.7695 ± 0.0118	0.7684 ± 0.0155	$\mathbf{0.2410} \pm 0.0022$
15	Books $\mathbf{X}^{(4)}$	0.2260 ± 0.0014	0.3303 ± 0.0024	0.3353 ± 0.0025	$\mathbf{0.2022} \pm 0.0008$
	Movies $\mathbf{X}^{(5)}$	0.2365 ± 0.0018	0.5132 ± 0.0044	0.5070 ± 0.0065	$\mathbf{0.2270} \pm 0.0009$
	Electronics $\mathbf{X}^{(6)}$	0.2447 ± 0.0013	0.8067 ± 0.0126	0.7688 ± 0.0122	$\mathbf{0.2417} \pm 0.0024$
Hotelling's T squared tests		2.80×10^{-7}	1.15×10^{-5}	1.11×10^{-6}	-

CST is not applied here as it does not support two or more principal coordinates on one factor. Best results for each rank are in bold. The Hotelling's T *squared tests row presents p-value of Hotelling's* T *squared tests between each algorithm and our proposed HISF.*

6.3.3 *Case #3. Latent Users' Taste Similarities when Buying Books, Movies and Electronics Devices and Latent Item Group Similarities can Help to Collaboratively Improve Recommendation Accuracy*

Joint analysis of all three rating matrices from Amazon website, $\mathbf{X}^{(4)}$ for books, $\mathbf{X}^{(5)}$ for movies and $\mathbf{X}^{(6)}$ for electronics, is studied. All of them are explicitly from the same users. Nevertheless, their hidden similarities in personal preferences and items' characteristics can also help to better suggest missing information. We want to assess how these explicit together with implicit similarities can help to collaboratively improve recommendation accuracy of each.

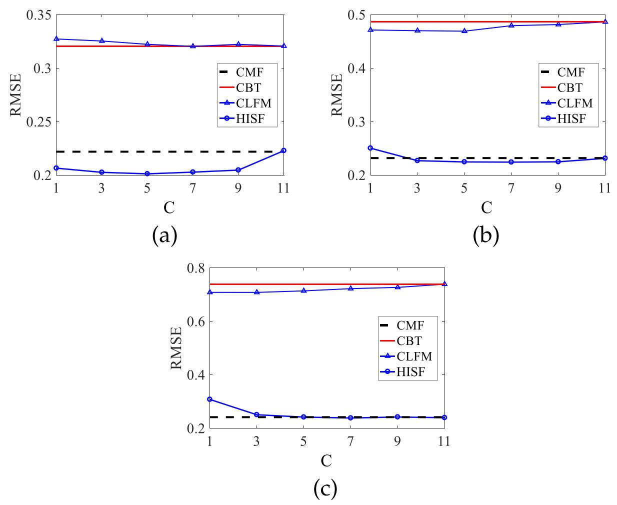


Fig. 11. Tested mean RMSEs under different common row c in coupled factor of HISF-N with Rank=11. (a) Results on Amazon Books. (b) Results on Amazon Movies. (c) Results on Amazon Electronics. CMF and CBT do not have c parameter, thus, their results are used as references. RMSE of HISF outperforms its competitors when c equals 7.

Table 6 summarizes the performance of CMF, CBT, CLFM and our proposed algorithm on Books, Movies and Electronics datasets from Amazon. CST is not applied due to its limit on one auxiliary dataset for one factor; in this case, we have two side data for user dimension. The results in this three datasets are consistent with those for two matrices above. In particular, CBT and CLFM which assume shared rating patterns among the three perform the worst. CMF improves accuracy a lot in comparison with CBT and CLFM. Nevertheless, its performance is once more time outperformed by our proposed ideas. The results, in this case, demonstrate our idea can be generalized to the situation of more than two datasets.

We perform Hotelling's T squared tests for three variables, i.e., Books, Movies and Electronics, between each baseline and HISF-N to confirm our method's statistical significance.

Our Hotelling's T squared tests this time also result in very small p-values. This once again confirms the performance between them is significantly different. We, thus, can conclude that their observed difference is convincingly significant.

We again show how HISF-N works with different values of c parameter in Fig. 11. There are two conclusions we can draw by observing the figures. First, domain-specific parts are quite substantial in addition to the common parts. Second, our model reduces $\mathbf{X}^{(4)}$'s RMSE the most, then $\mathbf{X}^{(5)}$'s and $\mathbf{X}^{(6)}$'s one in comparison with other algorithms. This can be explained by their observed data size. Our model, which optimizes the loss function (12), gives more preference on optimizing the domain with more data while preserving comparable performance on the other domains.

7 Related Work

Joint analysis of two datasets explores the possibility of using their correlations to better understand the underlying nature of the datasets. This thorough understanding provides more meaningful insights to improve recommendation accuracy.

Many methods have been proposed to achieve this goal. In this section, we summarize popular fundamental ideas and some of their extensions.

7.1 Collective Matrix Factorization (CMF)

CMF was proposed to jointly analyze two datasets having one common dimension [19]. CMF was based on the assumption that they have a common low-rank subspace in their coupled dimension. The concept of explicit similarities as the common factor has been widely used. Bhargava et al. [31] proposed location, activity and time would provide a complete picture of users. Thus, different data sources were modeled to have common coupled factors to fuse their explicit similarities. Transfer by Collective Factorization (TCF) [32] was proposed to use ratings and binary like/ dislike from the same users and items. TCF assumed that both user and item factors would be the same between the inputs. There are also several ideas to extend this CMF's coupled loss function. Acar et al. [20] introduced the concept for the joint analysis of a tensor [26] and another matrix. The authors proposed Coupled Matrix and Tensor Factorization (CMTF) to capture explicitly similarities (common coupled factor) between the tensor and the matrix. Weighted Non-negative Matrix Co-Tri-Factorization (WNMCTF) [33] extended CMF's idea to non-negative matrix tri-factorization [34].

7.2 CodeBook Transer (CBT)

Targeting on improving recommendation on one domain by utilizing similar latent rating patterns from another domain, Li et al. [23] suggested one as a source domain and another one as a target domain. In this research, the authors employed matrix tri-factorization to decompose the source into user, item and weighting factors. This weighting factor defined rating patterns of the users for the items, being used as the codebook to be transferred to the target. In this case, the explicit similarities were defined as the rating patterns transferred from the source to the target to improve recommendation of the target. An extension, named Rating-Matrix Generative Model (RMGM) [35], combined both the steps of extracting the codebook and transferring it. In case there are more than two datasets, Moreno et al. [36] introduced multiple explicit codebooks shared among them. Each codebook was also assigned a different weight to utilize the correlations better. Even though codebook transfer worked on any datasets, they are only effective when all datasets have the same rating patterns. This condition limits their applications to a broader available data.

7.3 Cluster-Level Latent Factor Model (CLFM)

The assumption that two datasets from different domains have the same rating patterns would be unrealistic in practice. This intuition motivated Gao et al. [24] to propose a generalized codebook sharing model by introducing CLFM. In specific, CLFM only shared common parts of the rating patterns between datasets. By doing so, CLFM model learned the only explicitly shared latent parts between datasets. When the number of commonly shared parts equals decomposition rank, CLFM is CBT. In all other cases, CLFM model learns the only the shared latent space while preserving the unique characteristics of each data. This model, thus, can be used to joint analyze multiple datasets to overcome the sparsity of each of them.

7.4 Coordinate System Transfer (CST)

All of the above methods were based on an equally shared correlation between datasets. Basing on observations that the rating matrix for recommendation is often sparse while auxiliary data of the users and items can sometimes be found, Weike et al. [4] suggested that common coordinates, e.g., users tastes, would exist between the main matrix and its side data. As a result, these coordinates could be transferred from the auxiliary data to the main rating matrix. The author proposed Coordinate System Transfer (CST) model to construct principal coordinates in low-dimensional space for the users and items. It then utilized these explicit similarities on principal coordinates as regularization terms on the target data.

8 Conclusion

In this research, we propose to discover both explicit and implicit similarities between datasets across domains. We propose to share common and preserve specific latent variables in coupled factors to better capture the true explicit characteristics of datasets across domains. Besides, we present an idea to discover non-coupled factors' latent similarities and make use of them to further improve the recommendation. Moreover, on the non-shared dimension, we propose to use the middle matrix of the tri-factorization to match the unique factors, and align the matched unique factors to further transfer cross-domain implicit similarities and thus further improve the recommendation. The advantages of our ideas, validated with real-world datasets, suggest combining both explicit and implicit similarities has a significant impact on improving the performance of cross-domain recommendation. The empirical results also encourage us to extend our idea to a generalized model which enables joint analysis of both similarities from multiple correlated datasets. In the future, we would like to extend our model to very high dimensional tensors using distributed processing.

Acknowledgments

This research is partially funded by the Natural Science Foundation of China (Grant No. 61602141) and the Science and Technology Program of Zhejiang Province (No. 2018c04001).

References

[1] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Comput.*, vol. 42, pp. 30–37, 2009.
[2] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*, Springer, pp. 73–105, 2011.
[3] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*, Springer, pp. 145–186, 2011.
[4] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, "Transfer learning in collaborative filtering for sparsity reduction," in *Proc. AAAI Conf. Artif. Intell.*, 2010, pp. 230–235.
[5] W. Chen, W. Hsu, and M. L. Lee, "Making recommendations from multiple domains," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 892–900.
[6] C.-Y. Li and S.-D. Lin, "Matching users and items across domains to improve the recommendation quality," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 801–810.
[7] M. Jiang, P. Cui, X. Chen, F. Wang, W. Zhu, and S. Yang, "Social recommendation with cross-domain transferable knowledge," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3084–3097, Nov. 2015.

[8] Y. Wei, Y. Zheng, and Q. Yang, "Transfer knowledge between cities," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1905–1914.

[9] D. Yang, J. He, H. Qin, Y. Xiao, and W. Wang, "A graph-based recommendation across heterogeneous domains," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 463–472.

[10] Y.-F. Liu, C.-Y. Hsu, and S.-H. Wu, "Non-linear cross-domain collaborative filtering via hyper-structure transfer," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1190–1198.

[11] T. Iwata and T. Koh, "Cross-domain recommendation without shared users or items by sharing latent vector distributions," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2015, pp. 491–502.

[12] H. Jing, A. C. Liang, S. D. Lin, and Y. Tsao, "A transfer probabilistic collective factorization model to handle sparse data in collaborative filtering," in *Proc. IEEE Int. Conf. Data Mining*, 2014, pp. 250–259.

[13] L. Zhao, S. J. Pan, E. W. Xiang, E. Zhong, Z. Lu, and Q. Yang, "Active transfer learning for cross-system recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2013, pp. 1205–1211.

[14] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu, "Personalized recommendation via cross-domain triadic factorization," in *Proc. Int. Conf. World Wide Web*, 2013, pp. 595–606.

[15] B. Wang, M. Ester, Y. Liao, J. Bu, Y. Zhu, Z. Guan, and D. Cai, "The million domain challenge: Broadcast email prioritization by cross-domain recommendation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1895–1904.

[16] C. C. Hsu, M. Y. Yeh, and S. d. Lin, "A general framework for implicit and explicit social recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2228–2241, Dec. 2018.

[17] F. Wu, Z. Yuan, and Y. Huang, "Collaboratively training sentiment classifiers for multiple domains," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 7, pp. 1370–1383, Jul. 2017.

[18] J. D. Zhang, C. Y. Chow, and J. Xu, "Enabling kernel-based attribute-aware matrix factorization for rating prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 798–812, Apr. 2017.

[19] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 650–658.

[20] E. Acar, T. G. Kolda, and D. M. Dunlavy, "All-at-once optimization for coupled matrix and tensor factorizations," in *MLG'11: Proc. Mining Learn. Graphs*, pp. 40–49, 2011.

[21] K. Shin, L. Sael, and U. Kang, "Fully scalable methods for distributed tensor factorization," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 100–113, Jan. 2017.

[22] W. Pan, "A survey of transfer learning for collaborative recommendation with auxiliary data," *Neurocomputing*, vol. 177, pp. 447–453, 2016.

[23] B. Li, Q. Yang, and X. Xue, "Can movies and books collaborate?: Cross-domain collaborative filtering for sparsity reduction," in *Proc. Int. Joint Conf. Artif. Intell.*, 2009, pp. 2052–2057.

[24] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari, and J. Guo, "Cross-domain recommendation via cluster-level latent factor model," in *Proc. European Conf. Mach. Learn. Knowl. Discovery Databases*, 2013, pp. 161–176.

[25] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 126–135.

[26] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, pp. 455–500, 2009.

[27] C. Bauckhage, "k-means clustering is matrix factorization," *arXiv*, vol. 1512.07548, 2015.

[28] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. IEEE Int. Conf. Data Mining*, 2008, pp. 263–272.

[29] R. He and J. McAuley , "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proc. Int. Conf. World Wide Web*, 2016, pp. 507–517.

[30] H. Hotelling, "The generalization of student's ratio," *Ann. Math. Statist.*, Springer, pp. 54–65, 1931.

[31] P. Bhargava, T. Phan, J. Zhou, and J. Lee, "Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data," in *Proc. Int. Conf. World Wide Web*, 2015, pp. 130–140.

[32] W. Pan, N. N. Liu, E. W. Xiang, and Q. Yang, "Transfer learning to predict missing ratings via heterogeneous user feedbacks," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 2318–2323.

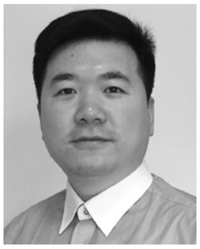
[33] J. Yoo and S. Choi, "Weighted nonnegative matrix co-tri-factorization for collaborative prediction," in *Proc. Asian Conf. Mach. Learn.: Adv. Mach. Learn.*, 2009, pp. 396–411.

[34] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Neural Inf. Process. Syst. 13*, 2001, pp. 556–562.

[35] B. Li, Q. Yang, and X. Xue, "Transfer learning for collaborative filtering via a rating-matrix generative model," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 617–624.

[36] O. Moreno, B. Shapira, L. Rokach, and G. Shani, "Talmud: Transfer learning for multiple domains," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 425–434.

Quan Do is working toward the PhD degree in the Advanced Analytics Institute, University of Technology Sydney, Australia. He conducts research on data mining and publishes papers on tensor factorization, recommendation systems, and cross-domain learning.



Wei Liu received the PhD degree from the University of Sydney. He is a senior lecturer with the Advanced Analytics Institute, School of Software, Faculty of Engineering and Information Technology, University of Technology (UTS), Sydney. Before joining UTS, he was a research fellow with the University of Melbourne and then a machine learning researcher with NICTA. He works in the areas of machine learning and data mining and has been publishing papers in research topics of tensor factorization, adversarial learning, graph mining, causal inference, and anomaly detection. He is a member of the IEEE.

Jin Fan received the BSc degree from Xian Jiaotong University, China, and the MSc and PhD degrees from Loughborough University, United Kingdom. She is currently an associate professor with the Department of Computer Science and Technology, Hangzhou Dianzi University, China. Her research interests include general areas of mobile sensing and related data analytics aspects.

Dacheng Tao (F'15) is a professor of computer science and ARC laureate fellow with the School of Information Technologies and the Faculty of Engineering and Information Technologies, and the inaugural director of the UBTECH Sydney Artificial Intelligence Centre, University of Sydney. He mainly applies statistics and mathematics to artificial intelligence and data science. His research interests include computer vision, data science, image processing, machine learning, and video surveillance. His research results have expounded in one monograph and more than 500 publications at prestigious journals and prominent conferences, such as the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Neural Networks and Learning Systems*, the *IEEE Transactions on Image Processing*, the *Journal of Machine Learning Research*, the *International Journal of Computer Vision*, NIPS, ICML, CVPR, ICCV, ECCV, ICDM; and ACM SIGKDD, with several best paper awards, such as the best theory/algorithm paper runner up award in IEEE ICDM'07, the best student paper award in IEEE ICDM'13, the distinguished student paper award in the 2017 IJCAI, the 2014 ICDM 10-year highest-impact paper award, and the 2017 IEEE Signal Processing Society Best Paper Award. He received the 2015 Australian Scopus-Eureka Prize, the 2015 ACS Gold Disruptor Award, and the 2015 UTS Vice Chancellor's Medal for Exceptional Research. He is a fellow of the IEEE, AAAS, OSA, IAPR, and SPIE.