

# Discovering both Explicit and Implicit Similarities for Cross-Domain Recommendation

Quan Do<sup>1</sup>, Wei Liu<sup>1</sup>, and Fang Chen<sup>2</sup>

<sup>1</sup> Advanced Analytics Institute, University of Technology Sydney, Sydney, Australia  
{Quan.Do,Wei.Liu}@uts.edu.au,

<sup>2</sup> Analytics Research Group, Data61  
Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia  
Fang.Chen@data61.csiro.au,

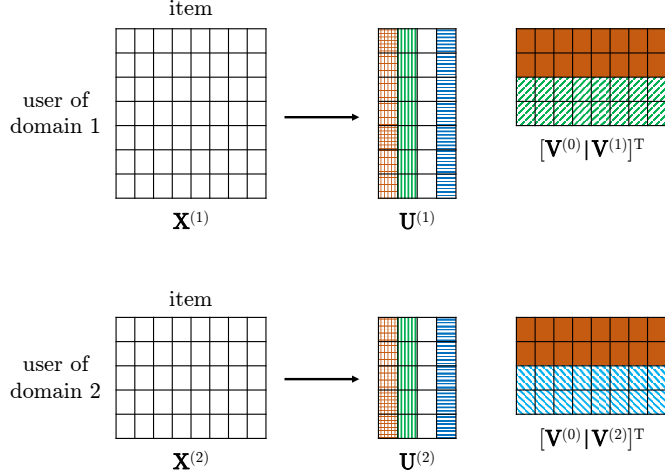
**Abstract.** *Recommender System has become one of the most important techniques for businesses today. Improving its performance requires a thorough understanding of latent similarities among users and items. This issue is addressable given recent abundance of datasets across domains. However, the question of how to utilize this cross-domain rich information to improve recommendation performance is still an open problem. In this paper, we propose a cross-domain recommender as the first algorithm utilizing both explicit and implicit similarities between datasets across sources for performance improvement. Validated on real-world datasets, our proposed idea outperforms the current cross-domain recommendation methods by more than 2 times. Yet, the more interesting observation is that both explicit and implicit similarities between datasets help to better suggest unknown information from cross-domain sources.*

**Keywords:** Cross-Domain learning; Recommendation System; Matrix Factorization

## 1 Introduction

Recommender systems have been received increasing attention and popularity from many products and services providers. Two approaches have been widely used for building recommender systems: content based [10] and collaborative filtering (CF) based [7]. A content based approach focuses on users' profile or items' information for making prediction whereas a CF method bases on latent similarities among users and items for recommending items for particular users. This paper focuses on improving CF based approaches.

As CF based methods rely on latent similarities among users and items for making recommendation, they require to have sufficient ratings to achieve a reliable result. There are two scenarios that may occur. Firstly, newly established services may take time to acquire enough ratings. Secondly, even they have enough ratings for making reliable recommendations, how can recommendation performance be improved using external data sources? Solving these two problems is our main focus.



**Fig. 1.** Our Cross-Domain Recommender model.  $\mathbf{X}^{(1)}$  matches  $\mathbf{X}^{(2)}$  in their item mode. Our proposed cross-domain factorization decomposes  $\mathbf{X}^{(1)}$  into  $\mathbf{U}^{(1)}$ ,  $[\mathbf{V}^{(0)}|\mathbf{V}^{(1)}]$  and  $\mathbf{X}^{(2)}$  into  $\mathbf{U}^{(2)}$ ,  $[\mathbf{V}^{(0)}|\mathbf{V}^{(2)}]$  where  $\mathbf{V}^{(0)}$  is a common part in coupled mode of both  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  while  $\mathbf{V}^{(1)}$  and  $\mathbf{V}^{(2)}$  are domain specific parts. Note that we also propose to utilize the columns of non-coupled  $\mathbf{U}^{(1)}$  and  $\mathbf{U}^{(2)}$  such that clusters of similar users (denoted by the same color patterns in their columns) in them are as close as possible.

The above issues are addressable given recent innovations on Internet and social media that have made many datasets publicly available [2, 9, 11, 5]. It is therefore easy to find a correlated dataset from another domain. For example, New South Wales (NSW) state’s crime statistics report can find NSW’s demography closely related. These correlations across domains possess explicit similarities, e.g., NSW crime report and NSW demography contain information of the same local areas, which are conventionally used to coupled analyze them [12, 1]. Nevertheless, datasets across domains also have implicit similarities, for example demography and crime behavior may have similar hidden patterns. Both similarities, if utilized properly, can provide rich insights to improve recommendation performance on the above mentioned two scenarios.

Formally, suppose we have rating matrices  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  from two different domains with only a few entries observed. Suppose  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  are explicitly coupled in one dimension, i.e., one mode of  $\mathbf{X}^{(1)}$  and one mode of  $\mathbf{X}^{(2)}$  are coupled. This is a reasonable assumption as many datasets across domains possess this characteristic. For instance,  $\mathbf{X}^{(1)}$  contains population profiles of a country’s cities and  $\mathbf{X}^{(2)}$  includes those cities’ crime reports;  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  are coupled in their first mode (city mode). By joint analyzing cross-domain  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$ , we want to learn ratings from observed  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  to predict their missing entries with high accuracy. In other words, our question is how to utilize the ratings in a domain to help predicting unknowns in another one and vice versa.

**Table 1.** Collaborative factors used by different algorithms for cross-domain recommendation. Our proposed method is the only one that utilizes both similarities.

Algorithm	Explicit Similarities	Implicit Similarities
CMF [12]	✓	
CBT [8]	✓	
CLFM [4]	✓	
Our proposed CDRec	✓	✓

Existing algorithms were trying to solve the above problem by using explicit similarities to collaborate between datasets. Collective Matrix Factorization (CMF) [12] and its extensions [1] suggested both datasets would have the same factor in their coupled mode. In this case, the low rank common factor captures the explicit similarities between datasets across domains. Gao et al. [4] and Li et al. [8] assumed cross-domain datasets would share explicit latent rating patterns. The ratings’ similarities were then used to collaborate between them. Nevertheless, just only explicit is not enough to really improve the accuracy.

We propose a Cross-Domain Recommender (**CDRec**) as the first method that analyzes both explicit and implicit similarities. One of our key hypothesis, extended from CMF where both datasets have the same factor in their coupled mode, is that two datasets across domains also possess their own specific patterns. Our idea is to find a way to combine these unique patterns into the common factor. One plausible solution is to allow the coupled factors to have both common and domain-specific parts. Figure 1 illustrates an example of common  $\mathbf{V}^{(0)}$  and unique  $\mathbf{V}^{(1)}, \mathbf{V}^{(2)}$  in coupled factors.

In addition, our another key hypothesis for implicit similarities, extended from the concept of factorization as a clustering method [3], is that non-coupled factors share common clusters. For example, in Figure 1, even though users in  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  are different, their behaviors may be grouped by latent similarities. Because of that, we want to align common clusters in  $\mathbf{U}^{(1)}$  and  $\mathbf{U}^{(2)}$  to be closer. One reasonable solution is to regularize the centroids of common clusters in  $\mathbf{U}^{(1)}$  and  $\mathbf{U}^{(2)}$ . This solution matches the fundamental concept of CF in which similar users rate similarly.

In short, our main contributions are:

- 1) **Preserving common and specific parts in the coupled factor** (subsection 4.1): We extend existing common coupled factor by introducing specific parts (e.g., factor  $\mathbf{V}$  in Figure 1.) These common and specific parts better capture the true explicit characteristics of datasets across domains.
- 2) **Aligning non-coupled factors’ similarities** (subsection 4.2): We present a method to utilize implicit similarities. CDRec is the first factorization method utilizing both explicit and implicit similarities for cross domain recommender systems’ performance improvement (Figure 1).
- 3) **Proposing an algorithm that solves optimization problem** (subsection 4.3): We optimize the process of utilizing both similarities following Alternating Least Squared (ALS) (Algorithm 1). Our empirical results on real-world

datasets suggest our proposed algorithm the best choice for cross-domain recommendation (section 5).

## 2 Notations

We denote matrices by boldface capitals, e.g.,  $\mathbf{X}$ ;  $\mathbf{I}$  is the identity matrix. Boldface lowercases are for vectors, i.e.,  $\mathbf{u}$  is a column vector and  $\mathbf{u}^T$  is a row vector. A boldface capital and lowercase with indices in its subscript are used for an entry of a matrix and a vector, respectively. Transpose of  $\mathbf{X}$  is denoted by  $\mathbf{X}^T$ .

## 3 Related work

Joint analyzing cross-domain datasets has attracted huge research effort to extract more meaningful insights. Many methods were proposed for making accurate recommendations. Some popular algorithms are being discussed below.

**Collective Matrix Factorization (CMF)** To deal with two datasets coupled in one of their modes, Singh et al. [12] and later Acar et al. [1] assumed two datasets had a common low rank subspace in their coupled dimension. Suppose  $\mathbf{X}_1$  and  $\mathbf{X}_2$  are joint in their first mode, the authors modeled this CMF with a coupled loss function:  $\mathcal{L} = \|\mathbf{X}_1 - \mathbf{U}\mathbf{V}_1^T\|^2 + \|\mathbf{X}_2 - \mathbf{U}\mathbf{V}_2^T\|^2$  where common  $\mathbf{U}$  represents explicit similarities between two datasets.

**CodeBook Transfer (CBT)** Targeting on improving recommendation on one domain by utilizing latent rating patterns from another domain, Li et al. [8] suggested one as a source domain  $\mathbf{X}_{\text{src}}$  and the other one as a target domain  $\mathbf{X}_{\text{tgt}}$ . Then the  $\mathbf{X}_{\text{src}}$  was decomposed into tri-factor:  $\mathbf{X}_{\text{src}} \approx \mathbf{U}_{\text{src}}\mathbf{S}_{\text{src}}\mathbf{V}_{\text{src}}^T$ . Rating patterns ( $\mathbf{S}_{\text{src}}$ ) were used as the CodeBook to be transferred from  $\mathbf{X}_{\text{src}}$  to  $\mathbf{X}_{\text{tgt}}$ . Thus,  $\mathbf{X}_{\text{tgt}}$  became  $\mathbf{X}_{\text{tgt}} \approx \mathbf{U}_{\text{tgt}}\mathbf{S}_{\text{src}}\mathbf{V}_{\text{tgt}}^T$ . This explicit knowledge transferred from the source improved the accuracy of recommendation in the target domain.

**Cluster-Level Latent Factor Model (CLFM)** The assumption that two datasets from different domains have the same rating patterns is unrealistic in practice. They may share some common patterns while possess their own characteristics. This motivated Gao et al. [4] to propose CLFM for cross-domain recommendation. In specific, the authors partitioned the rating patterns across domains into common and domain-specific parts:

$$\begin{aligned}\mathbf{X}_1 &\approx \mathbf{U}_1[\mathbf{S}_0|\mathbf{S}_1]\mathbf{V}_1^T \\ \mathbf{X}_2 &\approx \mathbf{U}_2[\mathbf{S}_0|\mathbf{S}_2]\mathbf{V}_2^T\end{aligned}$$

where  $\mathbf{S}_0 \in \mathbb{R}^{R_1 \times C}$  is the common patterns and  $\mathbf{S}_1, \mathbf{S}_2 \in \mathbb{R}^{R_1 \times (R_2 - C)}$  are domain-specific parts;  $C$  is the number of common columns.

This model allows CLFM to learn the only shared latent space  $\mathbf{S}_0$ , having two advantages. Firstly, as  $\mathbf{S}_0$  captures the similar rating patterns across domains, it

helps to overcome the sparsity of each datasets. Secondly, domain-specific  $\mathbf{S}_1$  and  $\mathbf{S}_2$  contain domains' discriminant characteristics. As a result, diversity of ratings in each domain is still preserved, improving recommendation performance.

## 4 Our proposed Cross-Domain Recommender (CDRec)

We propose a model that utilizes both explicit and implicit similarities between datasets across domains. Without loss of generality, assume  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  are coupled in their second mode, i.e.  $\mathbf{X}^{(1)}$  is a rating matrix from  $I$  users for  $J$  items and  $\mathbf{X}^{(2)}$  is another rating matrix from  $K$  users for the same  $J$  items. We follow the CMF model [12] to extract low rank user factors and item ones:  $\mathbf{X}^{(1)} \approx \mathbf{U}^{(1)}\mathbf{V}^T$  and  $\mathbf{X}^{(2)} \approx \mathbf{U}^{(2)}\mathbf{V}^T$ . Nevertheless, we make two key extensions:

### 4.1 Preserving common and domain-specific parts in $\mathbf{V}$

As  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  come from different domains, it is implausible to suggest them to have the same  $\mathbf{V}$ . They are highly correlated in a sense that they have something in common, yet also possess their own domain-specific parts. We, therefore, propose to include both common and domain-specific parts in  $\mathbf{V}$  factors in the coupled loss function:

$$\min \mathcal{L} = \left\| \mathbf{X}^{(1)} - \mathbf{U}^{(1)} [\mathbf{V}^{(0)} | \mathbf{V}^{(1)}]^T \right\|^2 + \left\| \mathbf{X}^{(2)} - \mathbf{U}^{(2)} [\mathbf{V}^{(0)} | \mathbf{V}^{(2)}]^T \right\|^2$$

where  $\mathbf{V}^{(0)} \in \mathbb{R}^{J \times C}$  is the common part and  $\mathbf{V}^{(1)}, \mathbf{V}^{(2)} \in \mathbb{R}^{J \times (R-C)}$  are domain-specific parts;  $C$  is the number of common columns.

Figure 1 illustrates an example of common  $\mathbf{V}^{(0)}$  and unique  $\mathbf{V}^{(1)}, \mathbf{V}^{(2)}$  in coupled factors. Common  $\mathbf{V}^{(0)}$  and domain-specific  $\mathbf{V}^{(1)}$  and  $\mathbf{V}^{(2)}$  better capture the characteristics of datasets across domains.

### 4.2 Utilizing implicit similarities in $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$

Besides explicit similarities in coupled mode as in subsection 4.1, cross-domain datasets also correlate in non-coupled mode. Following the concept of factorization as a clustering method [3], user groups in  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  are captured in  $\mathbf{U}^{(1)}$  and  $\mathbf{U}^{(2)}$ . Although users in  $\mathbf{X}^{(1)}$  and those in  $\mathbf{X}^{(2)}$  are different, their behaviors or preferences can be grouped together. This idea is the fundamental concept of CF where users rated similarly in observed items will also rate similarly for unobserved items. As a result, we suggest common clusters among columns of  $\mathbf{U}^{(1)}$  and  $\mathbf{U}^{(2)}$  to be closer. One reasonable solution is to regularize the centroids of clusters across  $\mathbf{U}^{(1)}$  and  $\mathbf{U}^{(2)}$ .

$$\min \mathcal{L} = \left\| \mathbf{X}^{(1)} - \mathbf{U}^{(1)} [\mathbf{V}^{(0)} | \mathbf{V}^{(1)}]^T \right\|^2 + \left\| \mathbf{X}^{(2)} - \mathbf{U}^{(2)} [\mathbf{V}^{(0)} | \mathbf{V}^{(2)}]^T \right\|^2 + \left\| \mathbf{u}_c^{(1)T} - \mathbf{u}_c^{(2)T} \right\|^2$$

where  $\mathbf{u}_c^{(1)T}$  and  $\mathbf{u}_c^{(2)T}$  denotes row vectors of columns' centroids in  $\mathbf{U}^{(1)}$  and  $\mathbf{U}^{(2)}$ , respectively.

Moreover, we also employ weighted  $\lambda$ -regularization [13] to our model below to prevent overfitting.

$$\begin{aligned} \min \mathcal{L} = & \left\| \mathbf{X}^{(1)} - \mathbf{U}^{(1)} [\mathbf{V}^{(0)} | \mathbf{V}^{(1)}]^\top \right\|^2 + \left\| \mathbf{X}^{(2)} - \mathbf{U}^{(2)} [\mathbf{V}^{(0)} | \mathbf{V}^{(2)}]^\top \right\|^2 \\ & + \left\| \mathbf{u}_c^{(1)\top} - \mathbf{u}_c^{(2)\top} \right\|^2 + \lambda \theta \end{aligned} \quad (1)$$

where  $\theta$  is the L2 regularization term such that

$$\theta = \|\mathbf{U}^{(1)}\|^2 + \|\mathbf{U}^{(2)}\|^2 + \|\mathbf{V}^{(0)}\|^2 + \|\mathbf{V}^{(1)}\|^2 + \|\mathbf{V}^{(2)}\|^2$$

### 4.3 Optimization

Even though (1) is a non-convex function with respect to all parameters, it is convex with respect to any of them when the others are fixed. Thus, we apply Alternating Least Square (ALS) algorithm [6] to alternately optimize the function with respect to one factor while fixing the others as in Algorithm 1. Moreover, to achieve efficiency, we perform our model optimization on each row of  $\mathbf{U}$  and  $\mathbf{V}$  factors instead of full matrix computation. So we can rewrite (1) as

$$\begin{aligned} \min \mathcal{L} = & \sum_{i,j}^{I,J} \left( \mathbf{X}_{i,j}^{(1)} - \mathbf{u}_i^{(1)\top} \begin{bmatrix} \mathbf{v}_j^{(0)} \\ \mathbf{v}_j^{(1)} \end{bmatrix} \right)^2 + \sum_{k,j}^{K,J} \left( \mathbf{X}_{k,j}^{(2)} - \mathbf{u}_k^{(2)\top} \begin{bmatrix} \mathbf{v}_j^{(0)} \\ \mathbf{v}_j^{(2)} \end{bmatrix} \right)^2 \\ & + \left\| \mathbf{u}_c^{(1)\top} - \mathbf{u}_c^{(2)\top} \right\|^2 + \lambda \theta \end{aligned} \quad (2)$$

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

Let  $\mathbf{v}_j^{(01)} = \begin{bmatrix} \mathbf{v}_j^{(0)} \\ \mathbf{v}_j^{(1)} \end{bmatrix}$  and  $\mathbf{v}_j^{(02)} = \begin{bmatrix} \mathbf{v}_j^{(0)} \\ \mathbf{v}_j^{(2)} \end{bmatrix}$ , then (2) becomes

$$\mathcal{L} = \sum_{i,j}^{I,J} \left( \mathbf{X}_{i,j}^{(1)} - \mathbf{u}_i^{(1)\top} \mathbf{v}_j^{(01)} \right)^2 + \sum_{k,j}^{K,J} \left( \mathbf{X}_{k,j}^{(2)} - \mathbf{u}_k^{(2)\top} \mathbf{v}_j^{(02)} \right)^2 + \left\| \mathbf{u}_c^{(1)} - \mathbf{u}_c^{(2)} \right\|^2 + \lambda \theta$$

Optimal  $\mathbf{u}_i^{(1)\top}$  can be achieved by setting the derivative of  $\mathcal{L}$  with respect to  $\mathbf{u}_i^{(1)\top}$  to zero.

$$\begin{aligned} \frac{\delta \mathcal{L}}{\delta \mathbf{u}_i^{(1)\top}} = & -2 \sum_j^J (\mathbf{X}_{i,j}^{(1)} - \mathbf{u}_i^{(1)\top} \mathbf{v}_j^{(01)}) \mathbf{v}_j^{(01)\top} + 2\lambda \mathbf{u}_i^{(1)\top} + 2(\mathbf{u}_i^{(1)\top} - \mathbf{b}^\top) \\ = & -2 \mathbf{x}_{i,*}^{(1)\top} \mathbf{V}^{(01)} + 2 \mathbf{u}_i^{(1)\top} \mathbf{V}^{(01)\top} \mathbf{V}^{(01)} + 2\lambda \mathbf{u}_i^{(1)\top} + 2 \mathbf{u}_i^{(1)\top} - 2 \mathbf{b}^\top \end{aligned}$$

where  $\mathbf{b}^\top = \mathbf{u}_c^{(1)\top} + \mathbf{u}_c^{(2)\top} - \mathbf{u}_i^{(1)\top}$  and  $\mathbf{x}_{i,*}^{(1)\top}$  is a row vector of all observed  $\mathbf{x}_{i,j}^{(1)}$ ,  $\forall j \in J$ .

Let  $\frac{\delta \mathcal{L}}{\delta \mathbf{u}_i^{(1)\top}} = 0$ , we can achieve the update rule for  $\mathbf{u}_i^{(1)\top}$ :

$$\mathbf{u}_i^{(1)\top} = \left( \mathbf{V}^{(01)\top} \mathbf{V}^{(01)} + (\lambda + 1) \mathbf{I} \right)^{-1} \left( \mathbf{x}_{i,*}^{(1)\top} \mathbf{V}^{(01)} + \mathbf{b}^\top \right) \quad (3)$$

Similarly, optimal  $\mathbf{u}_k^{(2)\top}$  can be derived by:

$$\mathbf{u}_k^{(2)\top} = \left( \mathbf{V}^{(02)\top} \mathbf{V}^{(02)} + (\lambda + 1) \mathbf{I} \right)^{-1} \left( \mathbf{x}_{k,*}^{(2)\top} \mathbf{V}^{(02)} + \mathbf{b}^\top \right) \quad (4)$$

where  $\mathbf{b}^\top = \mathbf{u}_c^{(1)\top} + \mathbf{u}_c^{(2)\top} - \mathbf{u}_k^{(2)\top}$  and  $\mathbf{x}_{k,*}^{(2)\top}$  is a row vector of all observed  $\mathbf{x}_{k,j}^{(2)}$ ,  $\forall j \in \mathbf{J}$ .  $\mathbf{I}$  is the identity matrix.

### Solving common $\mathbf{V}^{(0)}$

Let  $\mathbf{u}_i^{(1)\top} = [\mathbf{u}_i^{(10)} | \mathbf{u}_i^{(11)}]^\top$  and  $\mathbf{u}_k^{(2)\top} = [\mathbf{u}_k^{(20)} | \mathbf{u}_k^{(22)}]^\top$  where  $\mathbf{u}_i^{(10)\top}, \mathbf{u}_k^{(20)\top} \in \mathbb{R}^{1 \times C}$  and  $\mathbf{u}_i^{(11)\top}, \mathbf{u}_k^{(22)\top} \in \mathbb{R}^{1 \times R-C}$ , then (1) can be rewritten as:

$$\begin{aligned} \mathcal{L} = & \sum_{i,j} \left( \mathbf{x}_{i,j}^{(1)} - \mathbf{u}_i^{(10)\top} \mathbf{v}_j^{(0)} - \mathbf{u}_i^{(11)\top} \mathbf{v}_j^{(1)} \right)^2 + \sum_{k,j} \left( \mathbf{x}_{k,j}^{(2)} - \mathbf{u}_k^{(20)\top} \mathbf{v}_j^{(0)} - \mathbf{u}_k^{(22)\top} \mathbf{v}_j^{(2)} \right)^2 \\ & + \|\mathbf{u}_c^{(1)} - \mathbf{u}_c^{(2)}\|^2 + \lambda \theta \end{aligned}$$

Analogy to solving  $\mathbf{U}^{(1)}$  and  $\mathbf{U}^{(2)}$ , optimal  $\mathbf{v}_j^{(0)}$  can be achieved by setting the derivative of  $\mathcal{L}$  with respect to  $\mathbf{v}_j^{(0)}$  to zero.

$$\begin{aligned} \frac{\delta \mathcal{L}}{\delta \mathbf{v}_j^{(0)}} = & -2 \sum_i \left( \mathbf{Y}_{i,j}^{(1)} - \mathbf{u}_i^{(10)\top} \mathbf{v}_j^{(0)} \right) \mathbf{u}_i^{(10)} - 2 \sum_k \left( \mathbf{Y}_{k,j}^{(2)} - \mathbf{u}_k^{(20)\top} \mathbf{v}_j^{(0)} \right) \mathbf{u}_k^{(20)} + 2\lambda \mathbf{v}_j^{(0)} \\ = & -2 \mathbf{U}^{(1)\top} \mathbf{y}_{*,j}^{(1)} + 2 \mathbf{U}^{(1)\top} \mathbf{U}^{(1)} \mathbf{v}_j^{(0)} - 2 \mathbf{U}^{(2)\top} \mathbf{y}_{*,j}^{(2)} + 2 \mathbf{U}^{(2)\top} \mathbf{U}^{(2)} \mathbf{v}_j^{(0)} + 2\lambda \mathbf{v}_j^{(0)} \end{aligned}$$

The update rule for  $\mathbf{v}_j^{(0)}$  can be derived as:

$$\mathbf{v}_j^{(0)} = \left( \mathbf{U}^{(1)\top} \mathbf{U}^{(1)} + \mathbf{U}^{(2)\top} \mathbf{U}^{(2)} + \lambda \mathbf{I} \right)^{-1} \left( \mathbf{U}^{(1)\top} \mathbf{y}_{*,j}^{(1)} + \mathbf{U}^{(2)\top} \mathbf{y}_{*,j}^{(2)} \right) \quad (5)$$

### Solving domain-specific $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$

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

Then the update rule for  $\mathbf{v}_j^{(1)}$  can be derived as:

$$\mathbf{v}_j^{(1)} = \left( \mathbf{U}^{(1)\top} \mathbf{U}^{(1)} + \lambda \mathbf{I} \right)^{-1} \mathbf{U}^{(1)\top} \mathbf{y}_{*,j}^{(1)} \quad (6)$$

In a similar way, the update rule for  $\mathbf{v}_j^{(2)}$  can be achieved by:

$$\mathbf{v}_j^{(2)} = \left( \mathbf{U}^{(2)\top} \mathbf{U}^{(2)} + \lambda \mathbf{I} \right)^{-1} \mathbf{U}^{(2)\top} \mathbf{y}_{*,j}^{(2)} \quad (7)$$

---

**Algorithm 1:** CDRec

---

**Input** :  $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathcal{E}$   
**Output**:  $\mathbf{U}^{(1)}, \mathbf{V}^{(0)}, \mathbf{V}^{(1)}, \mathbf{U}^{(2)}, \mathbf{V}^{(2)}$

- 1 Randomly initialize all factors
- 2 Initialize  $\mathcal{L}$  by a small number
- 3 **repeat**
- 4      $\text{Pre}\mathcal{L} = \mathcal{L}$
- 5     Solve  $\mathbf{U}^{(1)}$  while fixing all other factor by minimizing (3)
- 6     Solve  $\mathbf{U}^{(2)}$  while fixing all other factor by minimizing (4)
- 7     Solve common  $\mathbf{V}^{(0)}$  while fixing all other factor by minimizing (5)
- 8     Solve domain-specific  $\mathbf{V}^{(1)}$  while fixing all other factor by minimizing (6)
- 9     Solve domain-specific  $\mathbf{V}^{(2)}$  while fixing all other factor by minimizing (7)
- 10    Compute  $\mathcal{L}$  following (1)
- 11 **until** ( $\frac{\text{Pre}\mathcal{L} - \mathcal{L}}{\text{Pre}\mathcal{L}} < \mathcal{E}$ )

---

## 5 Performance Evaluation

To evaluate our proposed idea, we compare CDRec<sup>3</sup> with existing algorithms on how well they utilize observed ratings to recommend unknown information. This section summaries our experiments’ settings and their results.

### 5.1 Data for the experiments

Two publicly available datasets: census data from Australian Bureau of Statistics (ABS) on different states<sup>4</sup> and crime statistics from Bureau of Crime Statistics and Research (BOCSAR)<sup>5</sup> are used for our evaluation. Table 2 summarizes these datasets’ distribution.

Australian Bureau of Statistics (ABS) publishes a comprehensive data about people and families for all Australia geographic areas. This ABS dataset has population and family profile within 154 New South Wales (NSW) state’ areas, so-called “local government areas” (LGA), and within 81 Victoria (VIC) state’ LGAs. We form these into a matrix  $\mathbf{X}^{(1)}$  of (LGA, population and family profile) of 154 by 7889 for NSW and a matrix  $\mathbf{X}^{(2)}$  of 81 by 7889 for VIC. We randomly select 10% of the data in our experiment and use about its 80% for training and 20% for testing.

BOCSAR’s crime data reports criminal incidents. There are 62 specific offences within 154 LGAs of New South Wales. The counting unit is the rate of criminal incidents per 100 population. We randomly select 10% of the data and

<sup>3</sup> CDRec’s source code is available at <https://github.com/quanie/CDRec>

<sup>4</sup> ABS: <http://www.abs.gov.au/websitedbs/censushome.nsf/home/datapacks>

<sup>5</sup> BOCSAR: [http://www.bocsar.nsw.gov.au/Pages/bocsar\\_crime\\_stats/bocsar\\_crime\\_stats.aspx](http://www.bocsar.nsw.gov.au/Pages/bocsar_crime_stats/bocsar_crime_stats.aspx)



**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 data of NSW ( $\mathbf{X}^{(3)}$ ).

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

include them in a matrix  $\mathbf{X}^{(3)}$  of (LGA, offences) of 154 by 62. Among this 10%, we use 80% for training and the rest for testing.

## 5.2 Baselines and metric for evaluation

We compare our proposed CDRec with existing cross-domain factorization algorithms: CMF [12], CBT [8] and CLFM [4] that leverage explicit similarities. Our goal is to assess how well these algorithms suggest unknown information based on the observed cross-domain ratings. For this purpose, we compare them with a popular Root Mean Squared Error (RMSE) metric.

## 5.3 Experimental setting

Two scenarios are thoroughly tested with the following settings:

### Case #1. States’ demographic similarities in latent sense can help to collaboratively suggest unknown information in these states

We use  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  which are from different LGAs in NSW and VIC states. Nevertheless, their demography may share some common characteristics. We would like to assess how well both explicit similarities in demography dimension and implicit ones in LGA dimension can help to collaboratively suggest unknown information in both NSW and VIC.

### Case #2. LGAs’ demographic similarities in latent sense can help to collaboratively suggest unknown crime information

In this case, we cross factorize NSW’s demography  $\mathbf{X}^{(1)}$  matrix and NSW’s crime  $\mathbf{X}^{(3)}$  matrix. Both  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(3)}$  explicitly have the same LGAs. Nevertheless, their hidden similarities in demography and crime rate can also help to improve recommendation performance. We want to assess how these explicit as well as implicit similarities can help to collaboratively suggest unknown crime rate.

Other than that, all algorithms stop when changes are less than  $10^{-5}$  which indicates convergence. In all algorithms, rank of the decomposition is set from 5 to 19. Each algorithm was run 5 times and we report their results’ mean and standard deviation in the next subsection.

**Table 3.** Tested RMSE on ABS NSW and ABS VIC data with different algorithms. Best results for each rank are in bold.

Dataset	Rank	CMF	CBT	CLFM	CDRec
ABS NSW	5	0.0226 $\pm$ 0.0026	0.0839 $\pm$ 0.0002	0.0838 $\pm$ 0.0002	<b>0.0132</b> $\pm$ 0.0002
	7	0.0222 $\pm$ 0.0009	0.0836 $\pm$ 0.0004	0.0842 $\pm$ 0.0006	<b>0.0131</b> $\pm$ 0.0003
	9	0.0241 $\pm$ 0.0011	0.0841 $\pm$ 0.0002	0.0848 $\pm$ 0.0009	<b>0.0143</b> $\pm$ 0.0004
	11	0.0265 $\pm$ 0.0026	0.0846 $\pm$ 0.0007	0.0841 $\pm$ 0.0007	<b>0.0143</b> $\pm$ 0.0003
	13	0.0237 $\pm$ 0.0024	0.0851 $\pm$ 0.0002	0.0850 $\pm$ 0.0005	<b>0.0151</b> $\pm$ 0.0004
	15	0.0229 $\pm$ 0.0029	0.0853 $\pm$ 0.0005	0.0847 $\pm$ 0.0005	<b>0.0150</b> $\pm$ 0.0000
ABS VIC	5	0.0364 $\pm$ 0.0031	0.0844 $\pm$ 0.0003	0.0845 $\pm$ 0.0004	<b>0.0266</b> $\pm$ 0.0030
	7	0.0428 $\pm$ 0.0020	0.0845 $\pm$ 0.0004	0.0849 $\pm$ 0.0004	<b>0.0239</b> $\pm$ 0.0025
	9	0.0476 $\pm$ 0.0040	0.0852 $\pm$ 0.0003	0.0848 $\pm$ 0.0003	<b>0.0221</b> $\pm$ 0.0019
	11	0.0501 $\pm$ 0.0029	0.0858 $\pm$ 0.0005	0.0851 $\pm$ 0.0007	<b>0.0242</b> $\pm$ 0.0015
	13	0.0489 $\pm$ 0.0032	0.0860 $\pm$ 0.0003	0.0852 $\pm$ 0.0003	<b>0.0227</b> $\pm$ 0.0015
	15	0.0514 $\pm$ 0.0041	0.0862 $\pm$ 0.0002	0.0854 $\pm$ 0.0006	<b>0.0215</b> $\pm$ 0.0005

#### 5.4 Empirical results

Table 3 shows RMSE performance of all models on ABS data for New South Wales and Victoria states. Both CBT and CLFM that assume two states’ demography similarities in latent sense clearly perform the worst. The results demonstrate that explicit similarities in latent sense does not help both CBT and CLFM to improve performance. CMF applies another approach to take advantages of explicit correlations between NSW state’s population and family profile and those of VIC state. As a result, CMF’s assumption on the same population and family profile factor between NSW and VIC helps improve CMF’s performance over CBT’s and CLFM’s almost 4 times in NSW data and 2 times in VIC data. Nevertheless, the prediction accuracy can be improved even more as illustrated with our proposed idea of explicit and implicit similarities discovery. Utilizing them help our proposed CDRec to achieve about 2 times higher accuracy compared with CMF.

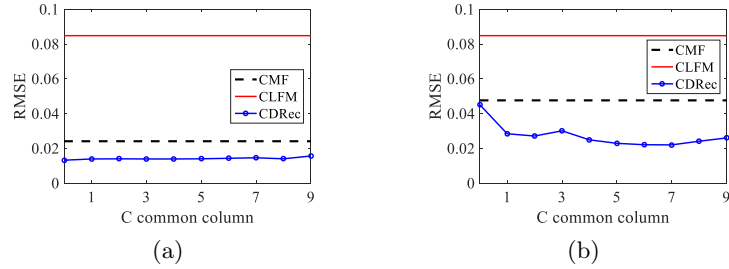
The advantages of both explicit and implicit similarities are further confirmed in Table 4. In this case, they are applied to other cross domains: ABS NSW demography and NSW Crime. These datasets have explicit similarities in their LGA latent factors. At the same time, implicit similarities in demography profile and crime behaviors are also utilized to collaborate between datasets. Our proposed CDRec leveraging both of the similarities outperforms existing algorithms by achieving the lowest RMSEs.

We also show how CDRec works with different number of common column C parameter in Figure 2 and 3. When there is no explicit similarities (C=0), the accuracy of VIC (Figure 2b) produced by our proposed method is almost the same as CMF’s performance whereas that of Crime (Figure 3a) is much worse than CMF’s one. Nevertheless, as C is larger, both explicit and implicit similarities help to further improve recommendation performance of them. Specifically, CDRec achieves the best result with C=6 in Figure 2b and C=9 in Figure 3a.

**Table 4.** Tested RMSE on ABS NSW demography and BOCSAR NSW crime data with different algorithms. Best results for each rank are in bold.

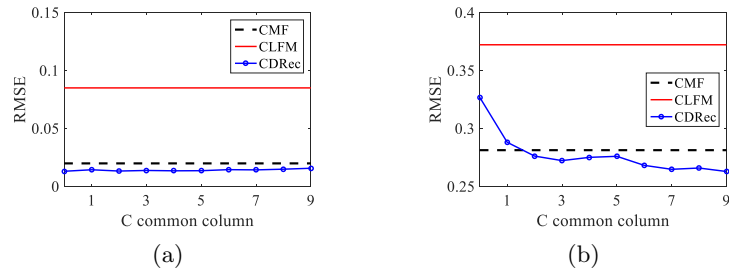
Dataset	Rank	CMF	CBT	CLFM	CDRec
Demography	5	0.0209 $\pm$ 0.0016	0.0840 $\pm$ 0.0001	0.0840 $\pm$ 0.0001	<b>0.0174</b> $\pm$ 0.0015
	7	0.0223 $\pm$ 0.0024	0.0840 $\pm$ 0.0002	0.0855 $\pm$ 0.0006	<b>0.0143</b> $\pm$ 0.0004
	9	0.0199 $\pm$ 0.0027	0.0838 $\pm$ 0.0002	0.0850 $\pm$ 0.0008	<b>0.0143</b> $\pm$ 0.0003
	11	0.0212 $\pm$ 0.0049	0.0839 $\pm$ 0.0001	0.0843 $\pm$ 0.0004	<b>0.0146</b> $\pm$ 0.0003
	13	0.0194 $\pm$ 0.0022	0.0837 $\pm$ 0.0001	0.0837 $\pm$ 0.0003	<b>0.0149</b> $\pm$ 0.0003
	15	0.0173 $\pm$ 0.0014	0.0835 $\pm$ 0.0001	0.0834 $\pm$ 0.0002	<b>0.0149</b> $\pm$ 0.0002
Crime	5	0.2796 $\pm$ 0.0204	0.3411 $\pm$ 0.0035	0.3422 $\pm$ 0.0071	<b>0.2697</b> $\pm$ 0.0073
	7	0.2907 $\pm$ 0.0265	0.3432 $\pm$ 0.0021	0.3912 $\pm$ 0.0188	<b>0.2716</b> $\pm$ 0.0029
	9	0.2813 $\pm$ 0.0261	0.3562 $\pm$ 0.0134	0.3722 $\pm$ 0.0249	<b>0.2648</b> $\pm$ 0.0058
	11	0.2689 $\pm$ 0.0143	0.3539 $\pm$ 0.0061	0.3712 $\pm$ 0.0199	<b>0.2618</b> $\pm$ 0.0012
	13	0.2700 $\pm$ 0.0150	0.3481 $\pm$ 0.0070	0.3500 $\pm$ 0.0135	<b>0.2623</b> $\pm$ 0.0024
	15	0.2647 $\pm$ 0.0031	0.3485 $\pm$ 0.0038	0.3580 $\pm$ 0.0099	<b>0.2625</b> $\pm$ 0.0015

Moreover, it is interesting to observe in both figures that RMSE of NSW demography reduces a bit to significantly improve that of VIC and NSW crime. This confirms both explicit and implicit similarities between cross-domain datasets can be collaborative used to improve both of their recommendation performance.

**Fig. 2.** Tested RMSEs under different number of common column  $C$  in CDRec. a) Results on ABS NSW dataset; b) Results on ABS VIC dataset.

## 6 Conclusion

We have discovered both explicit and implicit similarities between datasets across domains. In this paper we propose a method to preserving common and specific parts in coupled factor as well as aligning non-coupled factors' similarities. Moreover, an algorithm that solves the optimization problem of this method is also introduced. The advantages of our ideas, validated with real-world datasets, suggest combining both explicit and implicit similarities is the best way to improve cross-domain recommendation.



**Fig. 3.** Tested RMSEs under different  $C$  in CDRec. a) Results on ABS NSW demography; b) Results on BOCSAR NSW Crime. At  $C=9$ , RMSE of NSW demography reduces a little to further improve the accuracy of predicting crime information.

## References

- [1] Acar, E., Kolda, T.G., Dunlavy, D.M.: All-at-once optimization for coupled matrix and tensor factorizations. arXiv preprint arXiv:1105.3422 (2011)
- [2] Chen, W., Hsu, W., Lee, M.L.: Making recommendations from multiple domains. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) (2013)
- [3] Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix tri-factorizations for clustering. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) (2006)
- [4] Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., Guo, J.: Cross-domain recommendation via cluster-level latent factor model. In: Machine Learning and Knowledge Discovery in Databases: European Conference, (ECML PKDD) (2013)
- [5] Jiang, M., Cui, P., Chen, X., Wang, F., Zhu, W., Yang, S.: Social recommendation with cross-domain transferable knowledge. IEEE TKDE (2015)
- [6] Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Review (2009)
- [7] Koren, Y., Bell, R.: Advances in collaborative filtering. In: Recommender Systems Handbook (2011)
- [8] Li, B., Yang, Q., Xue, X.: Can movies and books collaborate?: Cross-domain collaborative filtering for sparsity reduction. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI) (2009)
- [9] Li, C.Y., Lin, S.D.: Matching users and items across domains to improve the recommendation quality. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) (2014)
- [10] Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Recommender Systems Handbook (2011)
- [11] Pan, W., Xiang, E., Liu, N., Yang, Q.: Transfer learning in collaborative filtering for sparsity reduction. In: AAAI Conference on Artificial Intelligence (2010)
- [12] Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) (2008)
- [13] Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R.: Large-scale parallel collaborative filtering for the netflix prize. In: Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management (AAIM) (2008)