Constrained Collective Matrix Factorization

Yu-Jia Huang  
Department of Computer Science  
Sun Yat-sen University  
Guangzhou, China  
huangyj9@mail2.sysu.edu.cn

Evan Wei Xiang  
Department of Computer Science and Engineering  
The Hong Kong University of Science and Technology  
Hong Kong, China  
wxiang@cse.ust.hk

Rong Pan  
Department of Computer Science  
Sun Yat-sen University  
Guangzhou, China  
panr@mail.sysu.edu.cn

ABSTRACT

Transfer learning for collaborative filtering (TLCF) aims to solve the sparsity problem by transferring rating knowledge across multiple domains. Taking domain difference into account, one of the issues in cross-domain collaborative filtering is to selectively transfer knowledge from source/auxiliary domains. In particular, this paper addresses the problem of inconstant users (users with changeable preferences across different domains) when transferring knowledge about users from another auxiliary domain. We first formulate the problem of inconstant users caused by domain difference and then propose a new model that performs constrained collective matrix factorization (CCMF). Our experiments on simulated and real data show that CCMF has superior performance than other methods.

Categories and Subject Descriptors
H.3 [INFORMATION STORAGE AND RETRIEVAL]:  
Information Search and Retrieval

Keywords  
Collaborative Filtering, Transfer Learning, Collective Matrix Factorization, Inconstant Users.

1. INTRODUCTION

Collaborative filtering [9] in recommender systems aims to predict users' ratings in the future on a set of items based on a collection of similar users' rating history. In real-life recommender systems, the rating matrix may be extremely sparse. As reported in [11], the density of the available ratings in commercial recommender systems is often less than 1%. To alleviate the sparsity problem in collaborative filtering, one common approach is to pool together the rating data from multiple rating matrices in related domains for knowledge transfer and sharing. Methods like [2, 3] assume that both users and items in an auxiliary domain are related to the target domain, while in practice it is often much easier to find an auxiliary data source with either related users or related items but not both. In this paper, we address the problem of user-sided transfer learning, although the method proposed in this paper can also be applied to item-sided transfer learning. One well-known approach to such one-side-related problem is Collective matrix factorization (CMF) [12]. CMF is proposed for jointly factorizing two matrices with the constraint of sharing one-side (user or item) latent features. CMF improves the prediction in target domain by increasing the rating records for each user/item. When the domain difference is small between source domain and target domain, e.g. transferring knowledge from a movie recommender system to another, this approach can easily improve prediction performance by simply jointing data from two systems.

CMF assumes that all users' features are constant. However, in real life, some users may change their features/preferences across different domains. For example, users having a preference over light music may like horror movies instead of ones about love stories because horror movies entertain audience in much more ways (pictures, stories and so on) than horror music (sounds). Even though the target domain and auxiliary domain are of same type (e.g. movie), users may also behave a little differently when rating different item sets. We refer “inconstant users” to those with changeable preferences across different domains. For these users, it would not hold that the source domain and target domain share a user latent feature matrix. To improve recommendation for inconstant users, we need a new method that takes the change of user feature into account.

To model the change of user feature caused by domain difference, we make two assumptions:

- Bias of latent features are user-dependent. The bias between user features across different domains varies from one user to another. Therefore, to recommend items for both constant and inconstant users, our model should allow user-dependent bias of latent features.

- The change lies in history. What items a user has rated implies what the user was looking for, thus we can infer whether and how much a user has inconstant hobbies through his/her rating history. More specifically, if the items rated by a user are similar to those
in source/auxiliary domain, we assume that the user tends to have constant preferences and vice versa.

To model the assumptions above, we extend CMF to a new matrix factorization model named Constrained CMF, denoted as CCMF in this paper. Unlike CMF, CCMF does not assume that the users share a same feature matrix in source domain and target domain. The remainder of this paper is organized as follows. In section 2, we first introduce the problem setting. In section 3, we formulate the model. Then we experimentally validate the effectiveness of the proposed models in section 4. Related work is introduced in section 5. Finally we conclude this paper in section 6.

2. PROBLEM SETTING

In our problem setting, we are given one source domain $D^{src}$, say music, and one target domain $D^{tgt}$ (movie in our experiments). $u_1,u_2,...,u_n$ denote $n$ users having rating records in both $D^{src}$ and $D^{tgt}$. In rating matrix $R^{src}$ of $D^{src}$, users make ratings on $m_{src}$ items $\{v_1^{src},v_2^{src},...,v_{m_{src}}^{src}\}$, the rating made by $u_i$ on $v_{jk}^{src}$ is denoted as $R_{ijk}^{src}$. Similarly, in rating matrix $R^{tgt}$ of $D^{tgt}$, users make ratings on $m_{tgt}$ items $\{v_1^{tgt},v_2^{tgt},...,v_{m_{tgt}}^{tgt}\}$, and the rating made by $u_i$ on $v_{jk}^{tgt}$ is denoted as $R_{ijk}^{tgt}$. Both $R^{src}$ and $R^{tgt}$ are observed and another $n \times m_{tgt}$ rating matrix $R^{test}$ is for testing and is unknown while training the model. All the elements from $R^{src}$, $R^{tgt}$ and $R^{test}$ are at the same scale (e.g. 1-5 or 0-1 [5]). Our goal is to make use of $R^{src}$ and $R^{tgt}$ to help predict the missing values in $R^{test}$. Figure 1 shows a toy example of our problem.

Figure 1: A Toy Example

3. CONSTRAINED CMF

A well-known and effective approach to recommender systems is to factorize the user-item rating matrix, and utilize the latent user feature matrix and item feature matrix to make prediction for future ratings [1, 4, 8, 13, 14].

In order to learn the latent characteristics of the users and items in source and target domain, we employ probabilistic matrix factorization[10] to factorize the user-item matrix. The conditional distributions over the observed ratings in two domains are defined as:

\[
p(R^{tgt} | U^{tgt}, V^{tgt}, \sigma^2_R^{tgt}) = \prod_{i=1}^{n} \prod_{j=1}^{m_{tgt}} \left[ N \left( R_{ijk}^{tgt} | U_{ik}^{tgt}, (V^{tgt})_j^{T}, \sigma^2_R^{tgt} \right) \right] I_{ijk}^{tgt},
\]

and

\[
p(R^{src} | U^{src}, V^{src}, \sigma^2_R^{src}) = \prod_{i=1}^{n} \prod_{j=1}^{m_{src}} \left[ N \left( R_{ijk}^{src} | U_{ik}^{src}, (V^{src})_j^{T}, \sigma^2_R^{src} \right) \right] I_{ijk}^{src},
\]

where $N(x | \mu, \sigma^2)$ is the probability density function of the Gaussian distribution with mean $\mu$ and variance $\sigma^2$. $I_{ijk}^{tgt}$ and $I_{ijk}^{src}$ are the indicator functions that are equal to 1 if user rated item and equal to 0 otherwise.

The zero-mean spherical Gaussian priors are also placed on user and item feature vectors:

\[
p(U^{src} | \sigma^2_U^{src}) = \prod_{i=1}^{n} N(U_{ik}^{src} | 0, \sigma^2_U^{src} I),
\]

\[
p(V^{src} | \sigma^2_V^{src}) = \prod_{j=1}^{m_{src}} N(V_j^{src} | 0, \sigma^2_V I),
\]

and

\[
p(V^{tgt} | \sigma^2_V^{tgt}) = \prod_{j=1}^{m_{tgt}} N(V_j^{tgt} | 0, \sigma^2_V I).\]

To transfer knowledge about users from auxiliary domain, [12] factorizes two matrices with the constraint of sharing one-side (user or item) latent features. Instead of jointly factorizing rating matrices in source and target domain with a same user feature matrix $(U^{src} = U^{tgt})$, our model factorizes $R^{src}$ with $U^{src}$ and $V^{src}$ and factorizes $R^{tgt}$ with $U^{tgt}$ and $V^{tgt}$. We transfer the user feature matrix $U^{src}$ learned in source domain to help factorize the rating matrix in target domain by conducting a constraint on $U^{src}$ and $U^{tgt}$:

\[
u_{ik}^{tgt} = U_{ik}^{src} + \frac{1}{\sum_{k=1}^{M} I_{ik}^{tgt}} \sum_{k=1}^{M} I_{ik}^{tgt} Y_k.
\]

or

\[
u_{ik}^{tgt} = U_{ik}^{src} + \frac{1}{\sum_{k=1}^{M} R_{ik}^{tgt}} \sum_{k=1}^{M} R_{ik}^{tgt}. \]

Here $U_{ik}^{src}$ is the prior mean of $U_{ik}^{tgt}$ and the second term is the user feature bias caused by domain difference. $Y_k$ captures the feature bias for a user if the user rated item $v_k^{tgt}$. Intuitively, the more similar $v_k^{tgt}$ is to the items in source domain, the closer $Y_k$ is to zero. The Gaussian prior for $Y$ is given as follows:

\[
p(Y | \sigma_Y^2) = \prod_{j=1}^{m_{tgt}} N(Y_j | 0, \sigma_Y^2 I).
\]

Hence, through a Bayesian inference, we have the log of the posterior distribution over the user and item feature and $Y$:

\[
\log p \left( \begin{array}{c} Y, U^{src}, V^{src}, T \end{array} | R^{src}, R^{tgt}, U^{src}, V^{src} \right) = -\frac{1}{2\sigma_R^2} \sum_{i=1}^{n} \sum_{j=1}^{m_{tgt}} I_{ij} \left( \left( R_{ij}^{src} - U_{ik}^{src}, (V^{src})_j^{T} \right)^2 \right)
\]

\[
-\frac{1}{2\sigma_R^2} \sum_{i=1}^{n} \sum_{j=1}^{m_{tgt}} I_{ij} \left( \left( R_{ij}^{tgt} - U_{ik}^{src}, (V^{src})_j^{T} + \sum_{k=1}^{M} I_{ik}^{tgt} Y_k \right)^2 \right)
\]

\[
-\frac{1}{2\sigma_V^2} \sum_{j=1}^{m_{tgt}} Y_j \cdot Y_j^{T} - \frac{1}{2\sigma_V^{tgt}} \sum_{j=1}^{m_{tgt}} Y_j \cdot Y_j^{tgt}
\]

\[-\frac{1}{2} \left( \sum_{i=1}^{n} \sum_{j=1}^{m_{tgt}} I_{ij} \left( \ln \sigma_Y^2 \right)^2 + n \ln \sigma_Y^2 + n \ln \sigma_V^2 \right) + C,
\]

where $C$ is a constant that does not depend on the parameters and $l$ denotes the feature number.

Maximizing the log-posterior over latent features with hyperparameters (i.e., the observation noise variance and prior variances) kept fixed is equivalent to minimizing the following sum-of-squared-errors objective functions with quadratic
regularization terms:

\[
E = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m_{src}} I_{ij}^{src} \left( \left( R_{ij}^{src} - U_{i}^{src} \cdot V_{j}^{srcT} \right)^2 \right) + \frac{1}{2} \sum_{j=1}^{m_{tgt}} I_{ij}^{tgt} \left( \left( R_{ij}^{tgt} - U_{i}^{src} \cdot \sum_{k=1}^{M} I_{ik}^{tgt} V_{j}^{src} \right)^2 \right) + \frac{\lambda_{U}}{2} \sum_{i=1}^{n} \left\| U_{i}^{src} \right\|_F^2 + \frac{\lambda_{V}}{2} \sum_{j=1}^{m_{tgt}} \left\| V_{j}^{tgt} \right\|_F^2 + \frac{\lambda_{Y}}{2} \sum_{j=1}^{m_{tgt}} \left\| Y_{j}^{tgt} \right\|_F^2 ,
\]

where \( \lambda_{U} = \sigma_{U}^2 / \sigma_{V}^2 \), \( \lambda_{V} = \sigma_{V}^2 / \sigma_{V}^2 \), \( \lambda_{Y} = \sigma_{Y}^2 / \sigma_{Y}^2 \), and \( \left\| \cdot \right\|_F^2 \) denotes the Frobenius norm. The graphical model is shown in Figure 2.

\[ \text{Figure 2: Graphical model of CCMF} \]

4. EMPIRICAL ANALYSIS

4.1 Experimental Setting

To check whether CCMF can fit with different settings, we evaluate CCMF on two data sets.

One is a simulated dataset sampled from the Netflix dataset. The Netflix data set used in the experiments is constructed as follows. We first randomly extracted a 10,000 \( \times \) 16,000 dense rating matrix \( R \) from the Netflix data, and take the submatrices \( R_{tgt} = R_{1,0000,1,0000} \) as the target rating matrix, and \( R_{src} = R_{1,8001,16000} \) as the user side source data, so that \( R_{tgt} \) and \( R_{src} \) share only common users.

The other is a real dataset crawled from Douban\(^1\), which is launched in 2005 and is a Chinese SNS website allowing registered users to record ratings and reviews related to movies, books, and music. We crawled 290,633 rating records rated by 5,000 users on 3,000 musical items and 10,000 movies. In the experiments, we used the musical rating matrix as auxiliary training data. Then we sampled randomly from the movie rating matrix to generate training data and testing data. The final datasets are summarized in Table 1. We adopt two evaluation metrics: RMSE (Root Mean Square Error) and MAE (Mean Absolute Error).

\[ \text{Table 1: Description of Douban data and Netflix data} \]

<table>
<thead>
<tr>
<th>dataset</th>
<th>type</th>
<th>sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netflix</td>
<td>movie</td>
<td>&lt;=0.60%</td>
</tr>
<tr>
<td>target(training)</td>
<td>movie</td>
<td>0.50%</td>
</tr>
<tr>
<td>target(testing)</td>
<td>movie</td>
<td>2.50%</td>
</tr>
<tr>
<td>auxiliary</td>
<td>movie</td>
<td>0.60%</td>
</tr>
<tr>
<td>Douban</td>
<td>music</td>
<td>1.50%</td>
</tr>
</tbody>
</table>

We compare our CCMF methods with two non-transfer learning methods: the UserMean and PMF\([10]\), as well as CMF\([12]\).

4.2 Experimental Results

We denote the CCMF methods that utilize different constraints given in Eqs. (6) and (7) as CCMF1 and CCMF2, respectively. The best results of using different parameters as described in the previous section are reported in Table 2. We can make the following observations:

- UserMean is worse than all other methods.
- Matrix factorization methods (CMF and CCMF) that transfer knowledge from auxiliary domains perform much better than non-transfer method PMF.
- CCMF and CCMF2, which model the inconstant users outperform CMF at all sparsity levels.
- CCMF1 and CCMF2 achieve close results, but as the training data in target domain becomes very sparse (< 0.3%), the performance of CCMF1 deteriorates most slowly.

One challenge of the transfer learning for recommender systems is that it is difficult to recommend items to users who have very few ratings in the target domain. In order to further compare the above methods, we first group all the users based on the number of observed ratings in the target domain, and then evaluate prediction accuracies of different user groups. The experimental results are shown in ...

\[ ^1 \text{http://www.douban.com} \]
Table 2: Performance Comparisons. Numbers in boldface (i.e. 0.890) and in Italic (i.e. 0.895) are the best and second best results among all methods, respectively.

<table>
<thead>
<tr>
<th>dataset</th>
<th>sparsity of R</th>
<th>without transfer</th>
<th>with transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Douban</td>
<td>CMF</td>
<td>CCMF</td>
</tr>
<tr>
<td>0.20%</td>
<td>0.776</td>
<td>0.776</td>
<td>0.776</td>
</tr>
<tr>
<td>0.40%</td>
<td>0.775</td>
<td>0.759</td>
<td>0.749</td>
</tr>
<tr>
<td>0.60%</td>
<td>0.769</td>
<td>0.756</td>
<td>0.750</td>
</tr>
<tr>
<td>Netflix</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.20%</td>
<td>0.914</td>
<td>0.914</td>
<td>0.914</td>
</tr>
<tr>
<td>0.40%</td>
<td>0.917</td>
<td>0.904</td>
<td>0.904</td>
</tr>
<tr>
<td>0.60%</td>
<td>0.905</td>
<td>0.895</td>
<td>0.895</td>
</tr>
</tbody>
</table>

(a) Distribution of Testing Data
(b) RMSE Comparison on Different User Rating Scales

Figure 3: Performance Comparison on Different Users

in Figure 3. Users with less than 160 ratings are grouped into 6 classes: “1-10”, “11-20”, “21-40”, “41-80”, “81-160”, and “161-640”, denoting how many ratings users have rated in target domain.

Figure 3a summarizes the distributions of testing data according to groups in the training data (sparsity 0.4%). For example, there are a total 19820 user-item pairs to be predicted in the testing dataset in which the related users in the training dataset have rating numbers from 1 to 10. In Figure 3b, we observe that our CCMF algorithm consistently performs better than other methods when recommending items for all user groups.

5. RELATED WORK

PMF Probabilistic matrix factorization (PMF) [10] is a method for missing value prediction in a single matrix.

CMF Collective matrix factorization (CMF) [12] is proposed for jointly factorizing two matrices with the constraints of sharing one-side (user or item) latent features. However, CMF didn’t address the problem that users’ may have different interests in different domains.

CST Coordinate System Transfer (CST) was a method proposed by [7] CST addresses the data sparsity problem in a target domain by transferring knowledge about both users and items from auxiliary data sources. In our problem setting, only knowledge about one side (user) from auxiliary domain is available. Hence, CST are not applicable to the problem in this paper.

TCF [6] proposed a framework of Transfer by Collective Factorization that use the binary preference data expressed in the form of like/dislike to help reduce the impact of data sparsity of more expressive numerical ratings.

6. CONCLUSIONS AND FUTURE WORKS

In this paper, we formulated the problem of inconstant users caused by domain difference and presented a new model to address the problem in transfer learning for collaborative filtering. Our method iteratively factorizes the rating matrices in source/auxiliary domain and target domain with a constraint on the user feature matrices for target domain and auxiliary domain. Experimental results on both and simulated and real data show that CCMF performs better than CMF at various sparsity levels. The study in this paper clearly demonstrates (a) the necessity of taking domain difference and change of user features into account, and (b) the items rated by a user implies to whether and how much the user’s feature has changed.

In this paper, we assumed that the rating matrices in auxiliary and target domain are one-sided aligned. But in real life, some users may not have rating records in both domains. Hence, in order to model the domain difference more realistically, for future work, we will extend CCMF so that it can fit partial-users aligned setting. In the future, we will also extend CCMF in heterogeneous settings, e.g. for transferring like/dislike knowledge from books or music to target domain that involves rating.

7. REFERENCES