Neurocomputing 136 (2014) 1–6

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Letters

Feature subspace transfer for collaborative filtering ☆

Jing Wang *, Liangwen Ke

The School of Computer Science and Technology, Huaqiao University, Xiamen City 361021, PR China

ARTICLE INFO

Article history:
Received 2 September 2013
Received in revised form
22 December 2013
Accepted 27 January 2014
Communicated by M. Wang
Available online 16 February 2014

Keywords:
Transfer learning
Wiberg algorithm
Collaborative filtering

ABSTRACT

The sparsity problem is a major bottleneck for the collaborative filtering. Recently, transfer learning methods are introduced in collaborative filtering to alleviate the sparsity problem which aim to use the shared knowledge in related domains to help improve the prediction performance. However, most of the transfer learning methods assume that the user features or item features learned from different data matrices have the same dimensions which is often not met in practice. In this paper, we propose a transfer learning method for collaborative filtering, called Feature Subspace Transfer (FST) to overcome this limitation. In our model, the user feature subspace learned from the auxiliary data is transferred to the target domain. An iterative algorithm is also proposed for solving the optimization problem. Numerical experiments on real-world data show the improvement of our method on alleviating the sparsity problem.

1. Introduction

In recent years, recommender systems play important roles in E-commerce which have been developed to help users identify the items that best fit their personal tastes. Collaborative filtering (CF) is one of the most effective recommendation approaches which aims at predicting the missing values in an user-rating matrix. See a typical user-rating matrix in Fig. 1(a). In the literature, the CF techniques can be classified into three main categories: memory-based, model-based, and hybrid CF algorithms [17]. Among these methods, matrix factorization (MF) may be the most popular one which represents the new trend in CF [8].

Given an original matrix $T \in R^{m\times n}$, matrix factorization tries to find a good approximation to T by the product of two matrix factors U and V. Considering the presence of noise in the data, the matrix factorization problem can be formulated as

$$\min_{U,V} \| T-UV^T \|_F^2, \tag{1}$$

with different constraints on U, V. The constraints on the matrix factors may be matrix structure constraints, orthogonal constraints [7], non-negative constraints [3] or label information constraints [5]. In some literatures, a regularization term is also introduced to (1) to prevent over-fitting, improve the robustness to the outliers, or incorporate the graph structure of the data matrix [2]. Although the matrix factorization methods have been successfully used in the areas of computer vision, pattern recognition, information retrieval and image processing [20–22], they cannot be applied to the recommendation systems directly.

In the recommendation systems, users may only rate a limited number of items which provides a user-rating matrix with missing entries. To indicate the existence of observed entries of the user-rating matrix $T \in R^{m\times n}$, a 0–1 weight matrix $W \in R^{m\times n}$ is provided, i.e., $W_{ij}=1$ if T_{ij} is observed or $W_{ij}=0$ otherwise. The optimization problem (1) is then modified as [1,8]

$$\min \sum_{(i,j)\in\Omega} \left(T_{ij}-(UV^T)_{ij}\right)^2 = \| W \odot (T-UV^T) \|_F^2, \quad \text{s.t.} \quad U \in R^{m\times d}, V \in R^{n\times d} \tag{2}$$

where Ω is the set of (i,j) for which the entry T_{ij} is observed, $\odot$ is the Hadamard product.[1]

In recent years, many approaches have been proposed for solving the optimization problem (2), but they have poor performance if the weight matrix W is extremely sparse. The sparsity problem has been a major bottleneck for most CF methods [17]. A new direction to alleviate the sparsity problem is to apply transfer learning to collaborative filtering. Although transfer learning methods have been widely used in many knowledge engineering areas including classification, regression, search and retrieval [13,18,19], the application of transfer learning to collaborative

☆The work of this author was supported in part by NSFC (61370006), Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University (ZQN-PY116), Program for New Century Excellent Talents in Fujian University (2012FJ-NCET-ZR01), and Outstanding Young Persons' Research Program for Higher Education of Fujian Province (11FJPY01).

* Corresponding author.
E-mail address: wroaring@yahoo.com.cn (J. Wang).

[1] Denote $A=W\odot T$, then $A_{ij}=W_{ij}T_{ij}$.

http://dx.doi.org/10.1016/j.neucom.2014.01.035

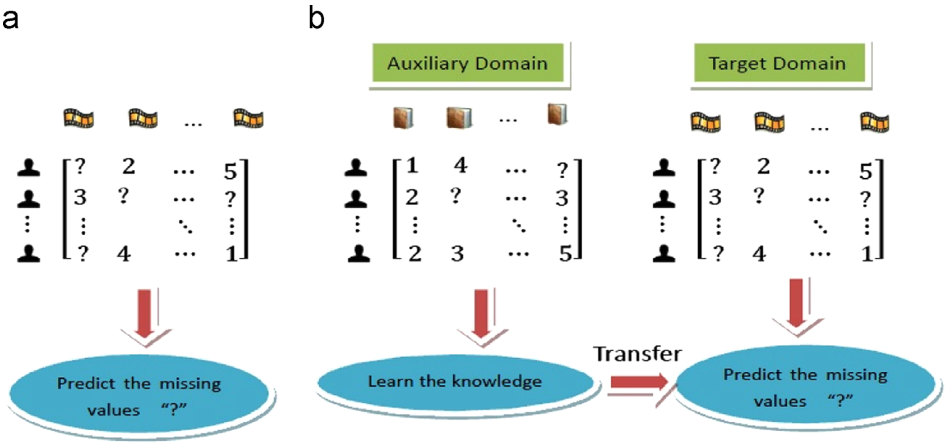

Fig. 1. Illustration of (a) collaborative filtering and (b) transfer learning in collaborative filtering.

filtering is still a new topic [4,10–12,14]. In the real world, the rating data from multiple related domains may share some common properties. Based on this basic observation, the transfer learning methods transfer the shared knowledge among related domains to help improve the prediction performance. See Fig. 1 (b) for the illustration of transfer learning in collaborative filtering. And most current work on transfer learning focuses on two main research issue: (1) what to transfer and (2) how to transfer [13]. For example, Coordinate System Transfer (CST) learns two coordinate systems from the auxiliary data, and then adapts the discovered principle coordinates to the target domain via a regularization tri-factorization method [10]. Collective Matrix Factorization (CMF) is a multi-task learning method which jointly factors multiple matrices, sharing latent features of the rows and columns in different matrices when an entity participates in multiple relations [14]. However, these methods have a limitation due to certain assumption that may be not met in practice. They require the latent dimensions of the user features and item features that learned from different data matrices to be the same, while the rank of the rating matrices in different domains may be different in practice.

In this paper, we propose a novel approach named feature subspace transfer (FST) for transferring the user knowledge from an auxiliary domain to improve the prediction performance in the target domain. We assume that the user tastes on the related domains should be similar. Different with the other work in transfer learning, we do not require the dimensions of the user features learned from the auxiliary and target data to be the same. Our approach consists of three steps. First, we learn the user feature subspace from the rating data matrix in the auxiliary domain by solving a nuclear norm regularized least squares problem. Second, we transfer the learned user feature subspace to the target domain by introducing a penalty term. Finally, we solve the optimization problem via an iterative algorithm.

The rest of the paper is organized as follows. In Section 2, we propose our feature subspace transfer model for collaborative filtering. The iterative algorithm for solving the model is proposed in Section 3. In Section 4, we validate the effectiveness of the proposed algorithm by the experiments on the Movielens and EachMovie data sets. Some conclusion remarks are given in Section 5.

2. Feature subspace transfer model

In our problem setting, we have a target domain and an auxiliary domain which share common users. Denote $A \in R^{m\times s}$ and $T \in R^{m\times n}$ the rating matrix in the auxiliary domain and target domain, respectively. The purpose of our approach is to make use of the user features learned from the auxiliary data A to improve the predictions of the missing values in the target data T. And our algorithm consists of three major steps: (1) Construct user's preference structure; (2) Transfer user feature subspace to the target domain; and (3) Solving the optimization problem via an iterative algorithm. The details of the first two steps to construct the optimization model are described in the section.

Step 1: Construct users' preference structure

In the step, we learn the user's preference structure from the data matrix A in the auxiliary domain. We find the principle coordinates of the auxiliary data by solving the nuclear norm regularized least-squares (LS) problem [6]

$$\min_{Z} \tfrac{1}{2}\|W \odot (A-Z)\|^2+\lambda\|Z\|_*, \tag{3}$$

where $\|Z\|_*$ is the sum of the singular values of Z. Denote Z_A the solution to (3) and let $Z_A = U_A \Sigma_A V_A^T$ be the SVD of Z_A, where $U_A \in R^{m\times r}$, $V_A \in R^{s\times r}$ are orthogonal matrices and $\Sigma_A \in R^{r\times r}$ is the diagonal matrix of singular values. The user's preference structure can be constructed by the columns of U_A.

In the paper, we use the principle coordinates of the solution to (3) to construct user's preference structure due to the following reasons:

(1) The iterative algorithm for the nuclear norm regularized LS problem scales to large problems.
(2) The dimension of the user's preference structure is unknown generally. Different with some other matrix factorization models such as (2), the model (3) does not require the rank r as an input parameter. Details of the optimization model (3) and its' iterative algorithm can be founded in [6].

Step 2: Transfer user feature subspace for collaborative filtering.

For a extremely sparse matrix T in the target domain, it is clear that the solution of the optimization problem (2) may be not unique. To cope with this problem and prevent over-fitting, (2) is often modified as

$$\begin{aligned} \min \quad & \|W \odot (T-UV^T)\|_F^2+\lambda_1\|U\|_F^2+\lambda_2\|V\|_F^2 \\ \text{s.t.} \quad & U \in R^{m\times d}, V \in R^{n\times d} \end{aligned} \tag{4}$$

with regularization parameters λ_1 and λ_2.

After obtaining the principle coordinates U_A from the auxiliary data, the latent user tastes can be transferred to the target domain. For example, we can add a constraint $U = U_A$ to the optimization problem (4). Note that the user's preference structure in the