A Matrix Factorization & Clustering Based Approach for Transfer Learning

V. Sowmini Devi, Vineet Padmanabhan(✉), and Arun K. Pujari

School of Computer and Information Sciences, Univeristy of Hyderabad, Hyderabad, India
sowmiveeramachaneni@gmail.com, {vineetcs,akpcs}@uohyd.ernet.in

Abstract. Recommender systems that make use of *collaborative filtering* tend to suffer from data sparsity as the number of items rated by the users are very small as compared to the very large item space. In order to alleviate it, recently *transfer learning* (TL) methods have seen a growing interest wherein data is considered from multiple domains so that ratings from the first (source) domain can be used to improve the prediction accuracy in the second (target) domain. In this paper, we propose a model for transfer learning in collaborative filtering wherein the latent factor model for the source domain is obtained through Matrix Factorization (MF). User and Item matrices are combined in a novel way to generate cluster level rating pattern and a Code Book Transfer (CBT) is used for transfer of information from source to the target domain. Results from experiments using benchmark datasets show that our model approximates the target matrix well.

1 Introduction

Recommender systems provide recommendations on products or services so that users get to know about items that match their interests. In order to learn user profiles, predict users' intensions and recommend items of interest, recommender systems usually employ techniques like Collaborative Filtering (CF) where recommendation for a user (target user) is done by utilizing the observed preferences of other users with similar tastes as that of the target user. Popular methods include MMMF [1,2] and PMF [3]. However, these methods can only utilize the data from a single domain and cannot take into account user-item interaction from other domains. Moreover, most CF-based recommender systems perform poorly when there are very few ratings. To address this data sparsity, transfer learning methods have emerged.

The idea behind transfer learning [4] is to extract and transfer common knowledge across the source and the target domain so as to built a predictive model across different domains. In the case of recommender systems, for successful knowledge transfer, TL has to address two critical problems (1) Knowledge transfer when two domains have aligned users or items and (2) Knowledge transfer when the domains have no aligned users or items. The second problem is very difficult and in this paper we use a representative method to solve this issue using

© Springer International Publishing AG 2017
B.U. Shankar et al. (Eds.): PReMI 2017, LNCS 10597, pp. 77–83, 2017.
https://doi.org/10.1007/978-3-319-69900-4_10

CBT (CodeBook Transfer) [5]. We propose a model for transfer learning in collaborative filtering in which the latent factor model for the source domain is obtained through matrix factorization techniques like MMMF (Maximum Margin Matrix Factorization) and PMF (Probabilistic Matrix factorization) and the cluster level patterns are generated via clustering techniques like *Spectral Clustering* and *k-means Clustering*. Thereafter, we use a tri-factorization method with the help of CBT that exploits matrix tri-factorization for transfer of information from the source to the target domain.

One work that comes close to ours is that of [6] where matrix approximation is combined with cluster-level factor vectors. However, their approach is limited to a single domain only. In [7] a coordinate system transfer method is proposed in which the latent features of users and items of source domain are learnt and adapted to a target domain. However, they require either common users or items between the two domains. In [5], co-clustering is applied on a separate *auxiliary* rating matrix to directly get cluster level rating pattern(B), which is then used in matrix tri-factorization. Our approach differs from theirs as we do not use a separate dense *auxiliary* rating matrix. The rest of the paper is organized as follows: Sect. 2 gives a brief description about Matrix Factorization. The proposed approach is given in Sect. 3. Finally experimental results are shown in Sect. 4, and we conclude our work in Sect. 5.

2 Matrix Factorization

Matrix factorization (MF) [2,8,9] techniques are a family of algorithms in collaborative filtering which try to approximate a low dimensional representation of the data. The users and items are projected to a lower dimensional embedding which are modelled as latent variables or hidden factors. The idea is that inference on these hidden factors lead to accurate predicton for ratings.

Formally, given a user-item rating matrix $Y \in \mathbb{R}^{m \times n}$ where m is the number of users and n is the number of items. Assuming that k is the number of latent factors, we need to find two matrices, $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{n \times k}$ such that their product is approximately equal to Y, i.e., $U \times V^T = \hat{Y} \approx Y$. Since we need to use only the observed ratings $\mathcal{O}$, the objective then reduces to find $\hat{Y} = UV^T$ by minimizing

$$\mathcal{J} = \sum_{(i,j)\epsilon\mathcal{O}} (y_{ij} - u_i v_j)^2 \tag{1}$$

Of the different matrix factorization techniques proposed we have chosen MMMF and PMF to be used in this paper.

Maximum Margin MF (MMMF)- When predicting discrete values such as ratings in recommender systems, a loss function other than the sum-squared error is more appropriate. In MMMF [1,10] sum-squared error is replaced with hinge loss. MMMF constrains the norms of U and V (trace norm) instead of their dimensionality and the predicted matrix contains only discrete values in $\{1, 2, ...r\}$. In order to output only the discrete values in MMMF we have to learn $r - 1$ thresholds θ_{ia} $(1 \leq a \leq r - 1)$ for every user i in addition to the

latent feature matrices U and V. For that, we need to minimize the following objective function:

$$\mathcal{J}(U,V,\theta) = \sum_{(i,j)\in\mathcal{O}} \sum_{a=1}^{r-1} h(\mathcal{T}_{ij}^{a}(\theta_{ia} - u_i v_j^T)) + \lambda(||U||_F^2 + ||V||_F^2) \tag{2}$$

where $\mathcal{T}_{ij}^{a} = \begin{cases} +1 & \text{if } a \geq y_{ij} \\ -1 & \text{if } a < y_{ij} \end{cases}$ $h(.)$ is a smoothed hinge loss function defined as $h(z) = (1-z)$, if $z < 1$ and $= 0$, otherwise, $\lambda > 0$ is regularization parameter.

Probabilstic MF- Probabilstic MF (PMF) is a generative model which presupposes a Gaussian distribution for the data. In this, ratings (Y) are modeled as draws from a Gaussian distribution with mean for Y_{ij} as $U_i V_j^T$. Zero-mean spherical gaussian priors are placed on U and V. i.e., Each row of U and V are drawn from a multi variate gaussian distribution with mean as 0 and precision is multiple of identity matrix I, as shown in equations below (3) and (4).

$$P(U|\sigma_U^2) = \prod_{i=1}^{m} \mathcal{N}(U_i|0, \sigma_U^2 I) \tag{3}$$

$$P(V|\sigma_V^2) = \prod_{j=1}^{n} \mathcal{N}(V_j|0, \sigma_V^2 I) \tag{4}$$

Given the user feature vectors and movie feature vectors, the distribution for the corresponding rating is given by Eq. (5),

$$P(Y|U,V,\sigma^2) = \prod_{i=1}^{m} \prod_{j=1}^{n} [\mathcal{N}(Y_{ij}|U_i V_j^T, \sigma^2)]^{I_{ij}} \tag{5}$$

Goal of PMF is to maximize the log-posterior of (5) over U and V. Maximizing the log posterior of (5) is equivalent to minimizing (6).

$$\mathcal{J} = \frac{1}{2}(\sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(Y_{ij} - U_i V_j^T)^2 + \lambda_U \sum_{i}^{m} ||U||_F^2 + \lambda_V \sum_{j}^{n} ||V||_F^2) \tag{6}$$

where, I_{ij} is the indicator matrix which equals 1 if item j is rated by user i otherwise 0, $\lambda_U = \frac{\sigma^2}{\sigma_U^2}$ and $\lambda_V = \frac{\sigma^2}{\sigma_V^2}$. One can solve the optimization functions given in Eqs. (2) and (6) using gradient descent.

3 Proposed Approach

For a target matrix (Y') of size $m' \times n'$ denoting users rating of items, our goal is to recommend the items in target domain using the source domain data. Initially, we apply MMMF (2) and PMF (6) individually on source domain to get

latent feature vectors U_s, V_s. Then we apply k-means clustering [11] or Spectral Clustering [12] on row vectors of U_s and V_s to get user-cluster latent matrix and item-cluster latent matrix. Following that we multiply them to get cluster level rating pattern (C). Once the rating pattern is formed, we try to minimize the objective function (7) which is a tri-factorization method so as to get the user and item membership matrices U_t, V_t of the target domain. After which predicted matrix can be obtained using Eq. (8) as outlined in Algorithm 1.

$$\min_{U_t \in \{0,1\}^{m' \times p}, V_t \in \{0,1\}^{n' \times q}} ||[Y' - U_t C V_t^T] \circ W||_F^2 \quad \text{s.t.}, U_t 1 = 1, V_t 1 = 1. \tag{7}$$

$$\tilde{Y'} = W \circ Y' + [1 - W] \circ [U_t C V_t^T], \tag{8}$$

where W is the indicator matrix of size $m' \times n'$ in which the value is 1 if the rating exists in original rating matrix, 0 otherwise. W ensures that the error is calculated only for the predicted ratings and, $\circ$ denotes element wise product. U_t and V_t are binary matrices, in which the value 1 (best cluster indicator) indicates whether a user or item belongs to a particular cluster and $U_t 1 = 1$, $V_t 1 = 1$ ensures that each user or item belongs to only one cluster. The solution to the optimization problem (Eq.-7) relates the source and target tasks and is NP-hard. Smaller value of Eq. (7) indicates that a better rating pattern between source and target while larger values indicate weak correspondence, which may result in negative transfer [13]. To get the minimum local solution, Alternating Least Squares (ALS) technique is used. ALS monotonically decreases Eq. (7), by updating U_t and V_t alternatively. This has been demonstrated in algorithm 2 of [5], where updating U_t is given in lines 7-10, and updating V_t is given in lines 11-14. Once we get U_t, V_t by solving the optimization function (7), we construct the predicted target matrix using Eq. (8), which is illustrated in Fig. 2. Consider Fig. 1, where source rating matrix (presented at level-1) is factorized into user latent factor matrix (U_s) and item latent factor matrix (V_s) as shown in level-2. Clustering technique is applied on U_s and V_s to get user and item cluster matrices (P, Q) which are at level-3. Finally, level-4 shows that these cluster matrices are multiplied to get cluster-level rating pattern (C) which is to be used in the target domain.

Algorithm 1. MF combined with clustering

1: **Input:** Source domain ratings
2: **Output:** Predicted target domain ratings
3: Find U_s, V_s by minimizing the optimization function of MMMF (2) or PMF (6).
4: Apply k-means clustering or spectral clustering on U_s, V_s to get user-cluster latent matrix(P) and item-cluster latent matrix(Q).
5: Calculate $C = P * Q'$ as cluster level rating pattern, which is assumed to be shared between two domains.
6: Use C, and find U_t, V_t of target domain by minimizing Eq. (7).
7: Using these U_t and V_t, find the predicted matrix using (8).

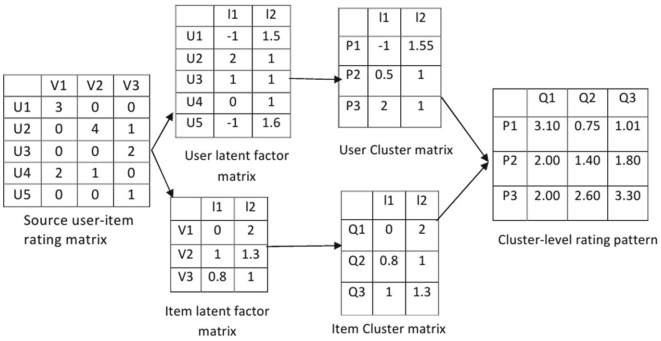

Fig. 1. Construction of cluster-level rating pattern using source rating data

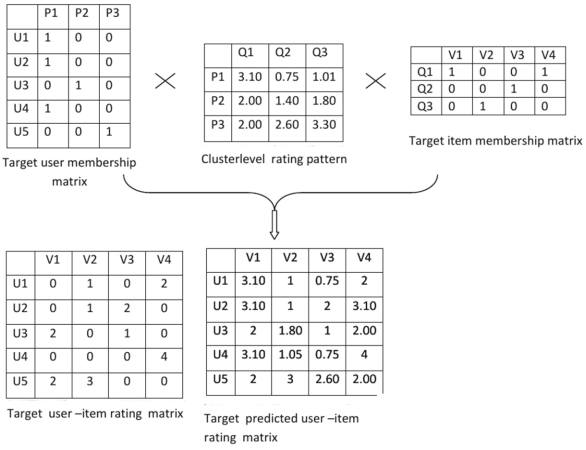

Fig. 2. Approximation of target rating matrix using cluster-level rating pattern.

4 Experimental Setup

The two datasets used in our experiments are `MovieLens` (https://grouplens.org//datasets/movielens/) as *source* dataset (6040 users and 3952 movies) and `Books` (https://grouplens.org/datasets/book-crossing/) as *target* dataset (2095 users and 4544 books). In movielens each user has ratings range of 1-5, whereas in books the range is 1-10, and we have scaled it to 1-5. In all experiments 80% of the total rating data is taken for training, and the rest 20% is used for testing. We evaluated our algorithm using Root Mean Squared Error (RMSE) Eq. (9) and Mean Absolute Error (MAE) Eq. (10), where smaller the values of these, better the performance. If we observe Table 1, we can see that MMMF or PMF, when combined with spectral clustering is giving better result (i.e., lesser RMSE and MAE) when compared with MMMF or PMF combined with k-means, which says that spectral clustering is more general and powerful compared to k-means

clustering technique. In some cases, even if the number of clusters is known, k-means clustering may fail to effectively cluster, because k-means is ideal to discover globular clusters, in which the members are in compact form but not connected.

$$RMSE = \sqrt{\sum_{(i,j)\epsilon\mathcal{O}} \frac{(y_{ij} - \hat{y}_{ij})^2}{|\mathcal{O}|}} \tag{9}$$

$$MAE = \sum_{(i,j)\epsilon\mathcal{O}} \frac{|(y_{ij} - \hat{y}_{ij})|}{|\mathcal{O}|} \tag{10}$$

where y_{ij} is the original rating and $\hat{y}_{ij}$ is the predicted rating.

Table 1. RMSE and MAE comparison of MMMF, PMF combined with k-means clustering and spectral clustering

	Number of clusters	*RMSE*		*MAE*	
		K-means	*Spectral*	*K-means*	*Spectral*
MMMF	40	0.9702	**0.9372**	0.6963	**0.6029**
PMF	40	0.8205	**0.8001**	0.6674	**0.6476**
MMMF	140	0.9690	**0.9171**	0.6986	**0.5864**
PMF	140	0.8282	**0.799**	0.799	**0.6867**
MMMF	200	1.0603	**0.9277**	0.777	**0.6044**
PMF	200	0.8535	**0.8362**	0.6778	**0.6473**
MMMF	300	1.0180	**0.9089**	0.7187	**0.5925**
PMF	300	0.8337	**0.8138**	0.6578	**0.6208**
MMMF	500	1.0927	**0.9247**	0.7813	**0.6105**
PMF	500	0.8452	**0.8123**	0.6508	**0.6222**

5 Conclusion and Future Work

We have proposed a novel model for cross-domain recommendation when multiple domains do not share a latent common rating pattern. We made use of Matrix Factorization techniques to get the initial latent hidden factor models and apply clustering techniques to find cluster-level rating pattern which is then used in a tri-factorization approximation. Experimental results using benchmark datasets shows that our model approximates the target matrix well. In the future we would like to vary the number of items in different domains which requires a special treatment and aslo investigate different techniques of tensor-based knowledge transfer learning.

References

1. Srebro, N., Rennie, J.D.M., Jaakkola, T.S.: Maximum-margin matrix factorization. In: NIPS, vol. 17, pp. 1329–1336 (2004)
2. Sowmini, V., Venkateswara Rao, K., Pujari, A.K., Padmanabhan, V.: Collaborative filtering by pso-based MMMF. In: Systems, Man and Cybernetics (SMC), pp. 569–574. IEEE (2014)
3. Ruslan, S., Mnih, A.: Probabilistic matrix factorization. In: NIPS, vol. 1 (2007)
4. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
5. Li, B., Yang, Q., Xue, X.: Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. IJCAI **9**, 2052–2057 (2009)
6. Ji, K., Sun, R., Li, X., Shu, W.: Improving matrix approximation for recommendation via a clustering-based reconstructive method. Neurocomputing **173**, 912–920 (2016)
7. Pan, W., Xiang, E.W., Nan Liu, N., Yang, Q.: Transfer learning in collaborative filtering for sparsity reduction. In: AAAI, vol. 10, pp. 230–235 (2010)
8. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8) (2009)
9. Wu, M.: Collaborative filtering via ensembles of matrix factorizations. In: Proceedings of KDD Cup and Workshop, vol. 2007 (2007)
10. Rennie, J.D.M., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: ICML, pp. 713–719 (2005)
11. Anil, K.J., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall Inc. (1988)
12. Ng, A.Y., Jordan, M.I., Weiss, Y., et al.: On spectral clustering: analysis and an algorithm. In: NIPS, vol. 14, pp. 849–856 (2001)
13. Rosenstein, M.T., Marx, Z., Kaelbling, L.P., Dietterich, T.G.: To transfer or not to transfer. In: NIPS 2005 Workshop on Transfer Learning, vol. 898 (2005)