Collaborative Filtering via Co-Factorization of Individuals and Groups

Yihai Huang and James T. Kwok

***Abstract*—Matrix factorization is one of the most successful collaborative filtering methods for recommender systems. Traditionally, matrix factorization only uses the observed user-item feedback information, which makes predictions on cold users/items difficult. In many applications, user/item content information are also available and they have been successfully used in content-based methods. In recent years, there are attempts to incorporate content information into matrix factorization. In particular, the Factorization Machine (FM) is one of the most notable examples. However, FM is a general factorization model that models interactions between all features into a latent feature space. In this paper, we propose a novel combination of tree-based feature group learning and matrix co-factorization that extends FM to recommender systems. Experimental results on a number of benchmark data sets show that the proposed algorithm outperforms state-of-the-art methods, particularly for predictions on cold users and cold items.**

I. Introduction

With the rapid growing amount of information available on the Internet, recommender systems [1] are receiving more and more attention. In a rating-based recommender system, the users provide explicit ratings (such as a 5-star score) for items. The ratings expressed by users on items are stored in a "rating matrix" which is usually very sparse. Thus, an important task of recommender systems is the rating prediction problem, which aims to accurately predict the missing ratings in the rating matrix.

Based on the kind of information used, existing rating prediction algorithms can be mainly classified as content-based [2] and collaborative-filtering-based [3]. Content-based methods use content information of users (such as demographic information) and items (such as genre) to match users' interests to items. In contrast, collaborative filtering methods use the observed ratings to predict the missing ratings. In recent years, collaborative-filtering-based methods are more popular as they are usually more accurate. Matrix factorization [4], [5] is one of the most successful collaborative filtering methods for rating prediction. However, the number of observed ratings for each user/item is usually imbalanced and many users/items have few observed ratings. Content information plays an important role for these cold users/items in content-based methods [2]. In recent years, there are efforts on incorporating content information into matrix factorization [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. In particular, the Factorization Machine (FM) [15] is one of the most successful such models as it can handle all kinds of content information without domain knowledge. The FM models pairwise interactions between all features in a latent feature space. A latent vector of each feature is created, and the pairwise interactions are weighted by the inner product of the corresponding latent vectors. It has been successfully used in many areas, such as social network prediction and ads click through prediction in KDDCup 2012[1].

However, the FM is originally designed as a general factorization model. There are some issues on applying the FM directly to recommender systems. (i) It may not be desirable to put user-item and user-user/item-item latent pairwise interactions in the same latent feature space and user-user/item-item latent pairwise interactions may not be useful. (ii) Only pairwise interactions are used, and so the model cannot capture higher degrees of nonlinearity. (iii) Feature latent vectors can lead to overfitting, as there are many parameters without enough prior knowledge to constrain them.

In this paper, we extend the FM to recommender systems. First, we propose that the latent pairwise interactions are only used between user features and item features, as internal interactions of user features/item features are not suitable for collaborative filtering and also the internal interactions and user-item external interactions may not be desirable to put in the same latent feature space. Thus, we remove all the other pairwise factorizations, making the resultant model lighter. Second, to improve the ability of capturing nonlinearity, higher degrees of interactions (user/item groups) are learned by a tree-based method. The indicators of user/item groups are used as new features to replace all the raw content features of user/item. Finally, we propose to construct individual-groups matrices to describe the group preferences and co-factorize the individual-individual rating matrix and individual-group rating matrices to achieve further improvement. To our best knowledge, this is the first work to incorporate individual-group ratings into matrix factorization models. The proposed method achieves significantly reduced error, particularly for the cold users/items.

The rest of the paper is organized as follows: Section II gives the problem formulation and introduces related methods. The proposed method is described in Section III. Section IV reports and analyzes the experimental results on several benchmark data sets. Finally, conclusions are made in Section V.

Yihai Huang and James T. Kwok are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (email: yhuangal, jamesk@cse.ust.hk).

[1] https://www.kddcup2012.org/workshop

978-1-4799-1959-8/15/$31.00

II. Collaborative Filtering via Matrix Factorization

A. Probabilistic Matrix Factorization (PMF)

In rating-based collaborative filtering, we are given a set of n users $\mathbb{U} = \{\text{user}_1, \ldots, \text{user}_n\}$, a set of m items $\mathbb{I} = \{\text{item}_1, \ldots, \text{item}_n\}$, and a sparse $n \times m$ rating matrix $\mathbf{R} = [R_{ij}]$ for the ratings of user_i on item_j. The goal is to predict the missing ratings in $\mathbf{R}$.

In recent years, matrix factorization has been popularly used for collaborative filtering [9], [4], [5]. In the probabilistic matrix factorization (PMF) model [5], both users and items share a common d-dimensional latent feature space. Each user/item is represented by a latent feature vector, and the predicted rating on an (item, user) pair is given by the inner product of the corresponding latent vectors. Specifically, let $\mathbf{U} \in \mathbb{R}^{n \times d}$ (resp. $\mathbf{V} \in \mathbb{R}^{m \times d}$) be the latent user (resp. item) matrix, in which the ith row $\mathbf{U}_i$ (resp. $\mathbf{V}_j$) represents user_i (resp. item_j). Let $\mathbb{O}_R = \{(i_1, j_1), (i_2, j_2), \ldots, (i_{|\mathbb{O}_R|}, j_{|\mathbb{O}_R|})\}$ be the set of observed rating pairs. The probabilistic model for PMF is given by:

$$p(\mathbf{R}|\mathbf{U}, \mathbf{V}, \sigma_R^2) = \prod_{(i,j) \in \mathbb{O}_R} \mathcal{N}(R_{ij}|\mathbf{U}_i^T \mathbf{V}_j, \sigma_R^2), \quad (1)$$

$$p(\mathbf{U}|\sigma_U^2) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{U}_i|\mathbf{0}, \sigma_U^2 \mathbf{I}),$$

$$p(\mathbf{V}|\sigma_V^2) = \prod_{j=1}^{m} \mathcal{N}(\mathbf{V}_j|\mathbf{0}, \sigma_V^2 \mathbf{I}), \quad (2)$$

where $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, σ_R^2 is the noise variance on the ratings, and σ_U, σ_V are prior variances on $\mathbf{U}$ and $\mathbf{V}$, respectively.

With fixed σ_R, σ_U and σ_V, on using the Bayes rule, maximizing the log posterior $\log p(\mathbf{U}, \mathbf{V}|\mathbf{R}, \sigma_R^2, \sigma_V^2, \sigma_U^2)$ is the same as minimizing the following objective:

$$\mathcal{L} = \frac{1}{2} \sum_{(i,j) \in \mathbb{O}_R} (R_{ij} - \mathbf{U}_i^T \mathbf{V}_j)^2 + \frac{\lambda_U}{2} \|\mathbf{U}\|_F^2 + \frac{\lambda_V}{2} \|\mathbf{V}\|_F^2,$$

where $\lambda_U = \sigma_R^2/\sigma_U^2$, and $\lambda_V = \sigma_R^2/\sigma_V^2$. After solving for $\mathbf{U}$ and $\mathbf{V}$, the missing rating $\hat{R}_{ij}$ of user_i on item_j can be predicted as $\mathbf{U}_i^T \mathbf{V}_j$.

B. Factorization Machine (FM)

In some applications, additional content information on users and/or items are available to help recommendation. For example, in a movie recommender system, one may have access to the user's age, gender, and occupation, and also the item's genre. We may then expect young female users to give higher ratings to romantic movies than crime movies. In context-aware recommender systems [16], context information such as timestamp and location may also be included.

Let these additional information be $\mathbf{x} = [\mathbf{x}^u, \mathbf{x}^i] \in \mathbb{R}^l$, where $\mathbf{x}^u$ is the part for user u, $\mathbf{x}^i$ is that for item i, and l is the total number of content features. For simplicity, we assume that the user and item identifiers are always included as content features. They are numbered from 1 to n for users, and from 1 to m for items.

For a particular $(\text{userid}, \text{itemid})$ pair, its prediction by the $(\text{second-order})^2$ FM [15] is given by

$$\hat{y} = \hat{R}_{\text{userid,itemid}} = \mathbf{w}^T \mathbf{x} + \sum_{i=1}^{l} \sum_{j=i+1}^{l} \tilde{\mathbf{V}}_i^T \tilde{\mathbf{V}}_j x_i x_j, \quad (3)$$

where $\mathbf{w} \in \mathbb{R}^l$, and $\tilde{\mathbf{V}}_i \in \mathbb{R}^d$ is the latent feature vector of the ith feature. In (3), $\mathbf{w}^T \mathbf{x}$ considers the strengths of individual features, while $\tilde{\mathbf{V}}_i^T \tilde{\mathbf{V}}_j x_i x_j$ models the pairwise interaction between features. i and j. This is similar to the $\mathbf{U}_i^T \mathbf{V}_j$ term in matrix factorization. To obtain parameters $\mathbf{w}$ and $\tilde{\mathbf{V}} = [\tilde{\mathbf{V}}_i]$, we minimize the regularized square loss:

$$\min_{\mathbf{w}, \tilde{\mathbf{V}}} \frac{1}{2} \sum_{k=1}^{|\mathbb{O}_R|} (y_k - \hat{y}_k)^2 + \frac{\lambda_w}{2} \|\mathbf{w}\|^2 + \frac{\lambda_{\tilde{V}}}{2} \|\tilde{\mathbf{V}}\|_F^2,$$

where $\hat{y}_k$ is given by (3).

C. Other Related Methods

Gu [14] proposed a graph-regularized weighted nonnegative matrix factorization model. It uses a weighted graph for the similarity of content information between users/items, and incorporates graph regularization. The assumption is that if two users/items have similar content information, their latent features should also be similar to each other.

Fang [17] and Lippert [13] proposed a matrix co-factorization model which simultaneously factorizes the user-item rating matrix and item-content/user-content matrix. However, the construction of these matrices is sometimes non-trivial as different kinds of content features have different meanings, and how to normalize the different features require domain knowledge. Moreover, in comparison with FM, it does not model pairwise interactions between features.

Mirbakhsh [12] proposed a clustering-based matrix factorization model. It first performs clustering on users and items, and constructs a cluster-based rating matrix by computing the average rating in each cluster. It factorizes the cluster-based rating matrix and original rating matrix separately, and then combines their results. Thus, it is not an integrated approach.

Regression-based factor models (RBFM) [18] have also been successfully used in a variety of recommendation problems. The idea is to replace the zero mean Gaussian vectors in PMF with user/item specific regression-based means based on content information. RBFM also adds another layer of linear regression on top of PMF. However, its training is based on the Monte Carlo EM, and not very efficient.

[2] In this paper, we focus on the second-order FM. In general, a higher-order FM can also be used. For example, a third-order FM captures all the triplet-wise interactions, and contains $\sum_{i_1=1}^{l} \sum_{i_2=i_1+1}^{l} \sum_{i_3=i_2+1}^{l} x_{i_1} x_{i_2} x_{i_3} \langle \tilde{\mathbf{V}}_{i_1}, \tilde{\mathbf{V}}_{i_2}, \tilde{\mathbf{V}}_{i_3} \rangle$, where $\langle \tilde{\mathbf{V}}_{i_1}, \tilde{\mathbf{V}}_{i_2}, \tilde{\mathbf{V}}_{i_3} \rangle = \sum_{f=1}^{d} \tilde{\mathbf{V}}_{i_1 f} \tilde{\mathbf{V}}_{i_2 f} \tilde{\mathbf{V}}_{i_3 f}$, in the model. While potentially more powerful, this can be computationally expensive when there are a large number of features.

III. Proposed Method

A. Constrained Probabilistic Matrix Factorization (CPMF)

In PMF [5], the prior means for the user/item feature vectors are zero. In this section, we propose to incorporate content information into these priors. Specifically, for item j, we decompose its latent feature $\mathbf{V}_j$ into two parts, as

$$\mathbf{V}_j = \mathbf{V}_j^c + \mathbf{V}_j^i, \tag{4}$$

where $\mathbf{V}_j^c$ is the component for the content information, and $\mathbf{V}_j^i$ is the part for the (intrinsic) non-content information. For example, if item j is a cartoon movie, we can set

$$\mathbf{V}_j^c = \mathbf{w}_{\text{genre=cartoon}}, \tag{5}$$
$$\mathbf{V}_j^i = \mathbf{w}_{\text{itemid}=j}, \tag{6}$$

where $\mathbf{w}_{\text{genre=cartoon}}$ and $\mathbf{w}_{\text{itemid}=j}$ are the latent feature vectors for genre and item identifier, respectively. Analogous to (2), for $\mathbf{V}^c = [\mathbf{V}_j^c]$ and $\mathbf{V}^i = [\mathbf{V}_j^i]$, we define

$$p(\mathbf{V}^c|\sigma_{V^c}^2) = \prod_{j=1}^{m} \mathcal{N}(\mathbf{V}_j^c|\mathbf{0}, \sigma_{V^c}^2\mathbf{I}), \tag{7}$$

$$p(\mathbf{V}^i|\sigma_{V^i}^2) = \prod_{j=1}^{m} \mathcal{N}(\mathbf{V}_j^i|\mathbf{0}, \sigma_{V^i}^2\mathbf{I}), \tag{8}$$

where $\sigma_{V^c}^2$ and $\sigma_{V^i}^2$ are the corresponding variances. Similarly, for user i, we decompose its latent feature as

$$\mathbf{U}_i = \mathbf{U}_i^c + \mathbf{U}_i^i. \tag{9}$$

If user i is 25 years old, female and a nurse, we can set

$$\mathbf{U}_i^c = \mathbf{w}_{\text{age=25}} + \mathbf{w}_{\text{sex=female}} + \mathbf{w}_{\text{occup=nurse}}, \tag{10}$$
$$\mathbf{U}_i^i = \mathbf{w}_{\text{userid}=i}, \tag{11}$$

where $\mathbf{w}_{\text{age=25}}, \mathbf{w}_{\text{sex=female}}, \mathbf{w}_{\text{occup=nurse}}$ are the latent feature vectors for her age, gender and occupation information, respectively. The corresponding probability distributions for $\mathbf{U}^c = [\mathbf{U}_i^c]$ and $\mathbf{U}^i = [\mathbf{U}_i^i]$ are

$$p(\mathbf{U}^c|\sigma_{U^c}^2) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{U}_i^c|\mathbf{0}, \sigma_{U^c}^2\mathbf{I}), \tag{12}$$

$$p(\mathbf{U}^i|\sigma_{U^i}^2) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{U}_i^i|\mathbf{0}, \sigma_{U^i}^2\mathbf{I}). \tag{13}$$

The proposed model will be called *constrained PMF* (CPMF), and its graphical model representation is shown in Figure 1.

The parameters $(\mathbf{U}^c, \mathbf{U}^i, \mathbf{V}^c, \mathbf{V}^i)$ can be learned by maximizing the posterior. It can be easily seen that it is the same as minimizing the following objective:

$$\mathcal{L} = \frac{1}{2}\sum_{(i,j)\in\mathbb{O}_R} [R_{ij} - (\mathbf{U}_i^c + \mathbf{U}_i^i)^T(\mathbf{V}_j^c + \mathbf{V}_j^i)]^2 + \frac{\lambda_{U^i}}{2}\|\mathbf{U}^i\|_F^2 + \frac{\lambda_{V^i}}{2}\|\mathbf{V}^i\|_F^2 + \frac{\lambda_{U^c}}{2}\|\mathbf{U}^c\|_F^2 + \frac{\lambda_{V^c}}{2}\|\mathbf{V}^c\|_F^2,$$

where $\lambda_{U^i}, \lambda_{V^i}, \lambda_{U^c}, \lambda_{V^c}$ are user-defined regularization hyperparameters.

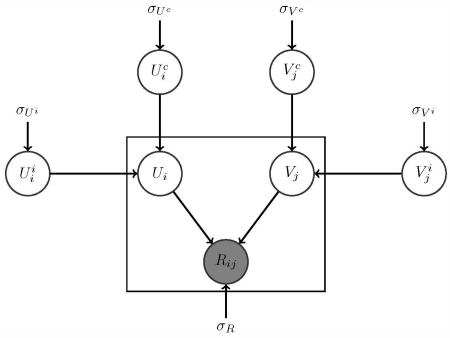


Fig. 1. Graphical model representation of the proposed constrained PMF (CPMF) model. $\mathbf{U}_i^c$ and $\mathbf{V}_j^c$ are the content-based priors of $\mathbf{U}_i$ and $\mathbf{V}_j$, while $\mathbf{U}_i^i$ and $\mathbf{V}_j^i$ are for the non-content information.

After training, the predicted rating of user i on item j can be obtained as

$$\hat{R}_{ij} = \mathbf{U}_i^T\mathbf{V}_j = (\mathbf{U}_i^c + \mathbf{U}_i^i)^T(\mathbf{V}_j^c + \mathbf{V}_j^i).$$

Using our previous example in (5), (6), (10), and (11), this can be written as

$$\begin{aligned}\hat{R}_{ij} = {} & \mathbf{w}_{\text{userid}=i}^T\mathbf{w}_{\text{itemid}=j} + \mathbf{w}_{\text{userid}=i}^T\mathbf{w}_{\text{genre=cartoon}} \\ & + \mathbf{w}_{\text{age=25}}^T\mathbf{w}_{\text{itemid}=j} + \mathbf{w}_{\text{age=25}}^T\mathbf{w}_{\text{genre=cartoon}} \\ & + \mathbf{w}_{\text{occup=nurse}}^T\mathbf{w}_{\text{itemid}=j} + \mathbf{w}_{\text{occup=nurse}}^T\mathbf{w}_{\text{genre=cartoon}} \\ & + \mathbf{w}_{\text{sex=female}}^T\mathbf{w}_{\text{itemid}=j} + \mathbf{w}_{\text{sex=female}}^T\mathbf{W}_{\text{genre=cartoon}}.\end{aligned}$$

Note that CPMF is a special case of FM, as it only models the pairwise interactions between user features and item features but not those between two different user features or (such as $\mathbf{w}_{\text{age=25}}^T\mathbf{w}_{\text{sex=female}}$) between two item features.

B. Constrained Probabilistic Matrix Co-Factorization (CPMCF)

Representations of a content-based prior like the one in (10) have several limitations. First, a linear combination cannot capture nonlinear interactions (for example, the latent feature $\mathbf{w}_{\text{age=25 and sex=female}}$ is typically not equal to sum of latent features $\mathbf{w}_{\text{age=25}} + \mathbf{w}_{\text{sex=female}}$). Second, as one content feature leads to one feature latent vector, the computational cost can be high when there are a lot of content features. Moreover, continuous features are clumsy to represent as they have an infinite number of possible values. To alleviate these limitations, one can bin each continuous feature and use the bin index as a categorical feature. This, however, requires an appropriate setting of the number of bins.

1) Feature Groups for $\mathbf{U}^c$ and $\mathbf{V}^c$: In this paper, we partition the users/items into groups based on the content information, and then use the group index as a new categorical feature. The user (resp. item) groups should be informative in that each group should reflect preferences for a local group of users (resp. items), rather than simply some global averaged properties. Moreover, each group should be easily interpreted so as to facilitate further data analysis.

Here, we borrow the idea of decision trees to divide users/items to groups. In the following, we focus on the

construction of user groups. The construction of item groups is analogous. Let $\mathbf{X}^u = [\mathbf{x}_1^u, \mathbf{x}_2^u, \ldots, , \mathbf{x}_{|\mathbb{O}_{\mathbf{R}}|}^u] \in \mathbb{R}^{|\mathbb{O}_{\mathbf{R}}| \times l^u}$ be the matrix containing the content information of users (where l^u is the number of user-content features), and $\mathbf{y} \in \mathbb{R}^{|\mathbb{O}_{\mathbf{R}}|}$ be the rating vector. Algorithm 1 shows the procedure that partitions users into groups (the partitioning of items is analogous). It is similar to the standard decision tree algorithm except that we use the function *NodeError* for splitting. For a given item j in a given node, if its observed ratings in this node is fewer than a certain threshold θ_r, the predicted rating value of item j is the average value of all the observed ratings in this node; otherwise, it is the average value of the observed ratings for it in this node. Moreover, node splitting stops when this node has fewer users than another threshold θ_u. An example is shown in Figure 2.

After the user and item groups are formed, for a user i belonging to user group k_1, we define its content latent feature and variance as

$$\mathbf{U}_i^c = \mathbf{w}_{k_1}^{usr}, \;\; \sigma_{U^c}^2 = \sigma_{w^{usr}}^2, \tag{14}$$

where $\mathbf{w}_{k_1}^{usr}$ is the latent feature vector of user group k_1. Similarly, for an item j belonging to item group k_2, its content latent feature and variance are

$$\mathbf{V}_j^c = \mathbf{w}_{k_2}^{itm}, \;\; \sigma_{V^c}^2 = \sigma_{w^{itm}}^2, \tag{15}$$

where $\mathbf{w}_{k_2}^{itm}$ is the latent feature vector of item group k_2.

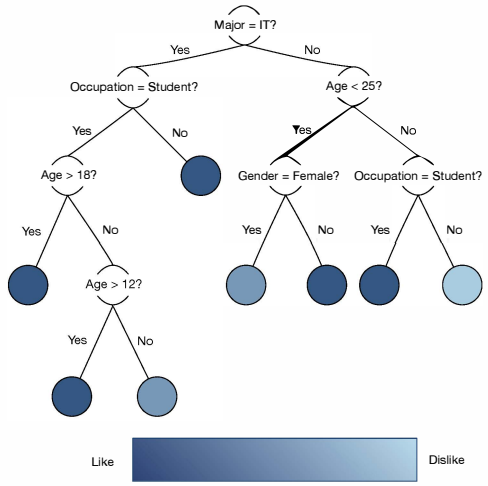


Fig. 2. An example decision tree that splits users into groups based on the user's content (in this case, demographic) information. For simplicity, the ratings are on one particular movie.

2) Ratings on Groups: From the learned item groups, we construct an *item group-based rating matrix* $\mathbf{T}^{itm} = [T_{ik}^{itm}] \in \mathbb{R}^{n \times c^{itm}}$, where c^{itm} is the number of item groups, and T_{ik}^{itm} is the average observed rating of user i on items in item group k:

$$T_{ik}^{itm} = \sum_{j \in \mathbb{O}_{T_{ik}^{itm}}} \frac{1}{|\mathbb{O}_{T_{ik}^{itm}}|} R_{ij},$$

where $\mathbb{O}_{T_{ik}^{itm}}$ is the set of items in item group k that have been rated by user i.

Algorithm 1 Algorithm for learning user groups.

Input: user feature matrix $\mathbf{X}^u$, rating vector $\mathbf{y}$.
function GENERATETREE($\mathbf{X}^u, \mathbf{y}$) /* main function */
  (bestf, bestv) ← SplitFeature($\mathbf{X}^u, \mathbf{y}$);
  if bestf $\neq -1$ **then** /* check stopping condition */
    for each branch i of split **do**
      find $\mathbf{X}_i^u$ and $\mathbf{y}_i$ falling in that branch according to (bestf, best);
      GenerateTree($\mathbf{X}_i^u, \mathbf{y}_i$);
    end for
  end if
end function
function SPLITFEATURE($\mathbf{X}^u, \mathbf{y}$)
  min_error ← MAX, bestf ← -1;
  for each feature i **do**
    for each possible binary split v **do**
      split $\mathbf{X}^u$ into $\mathbf{X}_1^u, \mathbf{X}_2^u$ according to the relation between v and feature i's value, and similarly $\mathbf{y}$ into $\mathbf{y}_1, \mathbf{y}_2$;
      e ← NodeError($\mathbf{X}_1^u, \mathbf{y}_1$)+NodeError($\mathbf{X}_2^u, \mathbf{y}_2$);
      if $e <$ min_error **then**
        min_error ← e; bestf ← i; bestv ← v;
      end if
    end for
  end for
  return (bestf,bestv);
end function
function NODEERROR($\mathbf{X}^u, \mathbf{y}$)
  error←0;
  if (#users in $\mathbf{X}^u$)$< \theta_u$ **then**
    return MAX;
  end if
  for each sample $(\mathbf{X}_i^u, y_i)$ **do**
    j ← itemid of the sample;
    if (#observed ratings in $\mathbf{X}^u$ for item j)$\geq \theta_r$ **then**
      $\hat{y}$ ← average observed rating in $\mathbf{X}^u$ for item j;
    else
      $\hat{y}$ ← average of all observed ratings in $\mathbf{X}^u$;
    end if
    error = error + $(\hat{y} - y_i)^2$;
  end for
  return error;
end function

Similarly, we can define the *user group-based rating matrix* $\mathbf{T}^{usr} = [T_{jk}^{usr}] \in \mathbb{R}^{m \times c^{usr}}$, where c^{usr} is the number of user groups, and T_{jk}^{usr} is the average observed rating on item j by users of user group k:

$$T_{jk}^{usr} = \sum_{i \in \mathbb{O}_{T_{jk}^{usr}}} \frac{1}{|\mathbb{O}_{T_{jk}^{usr}}|} R_{ij},$$

where $\mathbb{O}_{T_{jk}^{usr}}$ is the set of users in user group k that have rated item j.

When $|\mathbb{O}_{T_{jk}^{usr}}|$ is small, the T_{jk}^{usr} value obtained may not be reliable. Thus, we set a threshold on the minimum number

of observed ratings required. If the number of users in group k who have rated item j is below this threshold, we declare that T^{usr}_{jk} is missing. A large threshold value makes the T^{usr}_{jk} values more reliable, but will lead to more missing values in $\mathbf{T}^{usr}$. In general, the sparser the $\mathbf{R}$, a smaller threshold is used. In the experiments, a small threshold (around 10) is used.

3) Model: CPMF incorporates content information with our tree-based method to learn different prior latent feature vectors in different groups. However, these prior latent feature vectors are also parameters to be learned. Over-fitting can occur when there are too many learnable parameters but too few constraints. This can be alleviated by using the group rating information, since $\mathbf{V}^c$ captures the latent vectors of item groups, and $\mathbf{T}^{itm}$ captures the ratings of groups. Motivated by the factorization of $\mathbf{R}$, we factorize $\mathbf{T}^{itm}$ and $\mathbf{T}^{usr}$ as follows.

$$p(\mathbf{T}^{itm}|\mathbf{U},\mathbf{V}^c,\sigma^2_{T^{itm}}) = \prod_{i=1}^{n}\prod_{k=1}^{c^{itm}}\prod_{j\in\mathbb{O}_{T^{itm}_{ik}}} \mathcal{N}(T^{itm}_{ik}|\mathbf{U}_i^T\mathbf{V}^c_j,\sigma^2_{T^{itm}})^{1/\left|\mathbb{O}_{T^{itm}_{ik}}\right|} \quad (16)$$

$$p(\mathbf{T}^{usr}|\mathbf{U}^c,\mathbf{V},\sigma^2_{T^{usr}}) = \prod_{j=1}^{m}\prod_{k=1}^{c^{usr}}\prod_{i\in\mathbb{O}_{T^{usr}_{jk}}} \mathcal{N}(T^{usr}_{jk}|\mathbf{V}_j^T\mathbf{U}^c_i,\sigma^2_{T^{usr}})^{1/\left|\mathbb{O}_{T^{usr}_{jk}}\right|}, \quad (17)$$

where $\sigma^2_{T^{itm}}$ and $\sigma^2_{T^{usr}}$ are the noise variances of group based-ratings $\mathbf{T}^{itm}$ and $\mathbf{T}^{usr}$ The corresponding graphical model is shown in Figure 3. As we simultaneously factorize both $\mathbf{R}$ and $\mathbf{T}$s with shared parameters, it is called the constrained probabilistic matrix co-factorization (CPMCF) model.

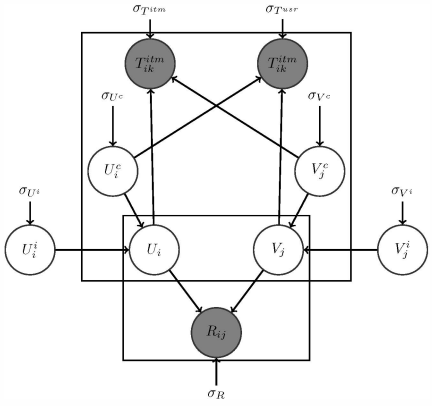


Fig. 3. Graphical model representation of the proposed CPMCF model.

As $\mathbf{T}^{itm}$ and $\mathbf{T}^{usr}$ are assumed to be independent, using the Bayes rule, the posterior of model parameters is given by

$$\begin{aligned} p(\mathbf{U}^i,\mathbf{V}^i,\mathbf{w}^{usr},\mathbf{w}^{itm}|\mathbf{R},\mathbf{T}^{usr},\mathbf{T}^{itm},\sigma^2_R & \\ \sigma^2_{U^i},\sigma^2_{V^i},\sigma^2_{w^{usr}},\sigma^2_{w^{itm}},\sigma^2_{T^{usr}},\sigma^2_{T^{itm}}) & \\ = p(\mathbf{R},\mathbf{T}^{usr},\mathbf{T}^{itm}|\mathbf{U}^i,\mathbf{V}^i,\mathbf{w}^{usr},\mathbf{w}^{itm},\sigma^2_R,\sigma^2_{T^{usr}},\sigma^2_{T^{itm}}) & \\ \cdot p(\mathbf{U}^i,\mathbf{V}^i,\mathbf{w}^{usr},\mathbf{w}^{itm}|\sigma^2_{U^i},\sigma^2_{V^i},\sigma^2_{w^{usr}},\sigma^2_{w^{itm}}) & \\ / p(\mathbf{R},\mathbf{T}^{usr},\mathbf{T}^{itm}|\sigma^2_R,\sigma^2_{U^i},\sigma^2_{V^i},\sigma^2_{w^{usr}},\sigma^2_{w^{itm}},\sigma^2_{T^{usr}},\sigma^2_{T^{itm}}) & \\ \propto p(\mathbf{R}|(\mathbf{U},\mathbf{V},\sigma^2_R) & \\ p(\mathbf{T}^{usr}|\mathbf{U}^c,\mathbf{V},\sigma^2_{T^{usr}})p(\mathbf{T}^{itm}|\mathbf{U},\mathbf{V}^c,\sigma^2_{T^{itm}}) & \\ p(\mathbf{U}^i|\sigma^2_{U^i})p(\mathbf{V}^i|\sigma^2_{V^i})p(\mathbf{w}^{usr}|\sigma^2_{w^{usr}})p(\mathbf{w}^{itm}|\sigma^2_{w^{itm}}). & \end{aligned}$$

Using (1), (4), (7), (8), (9), (12), (13), (14), (15), (16) and (17), for fixed observation noise variance and prior variances, maximizing the posterior is equivalent to minimizing the following objective:

$$\begin{aligned} \mathcal{L} = & \frac{1}{2}\sum_{(i,j)\in\mathbb{O}_{\mathbb{R}}}[R_{ij}-(\mathbf{U}^i_i+\mathbf{w}^{usr}_{g^u(i)})^T(\mathbf{V}^i_j+\mathbf{w}^{itm}_{g^i(j)})]^2 \\ & +\frac{\lambda_{T^{usr}}}{2}\sum_{(j,k)\in\mathbb{O}_{T^{usr}}}(T^{usr}_{jk}-\mathbf{V}_j^T\mathbf{w}^{usr}_k)^2 \\ & +\frac{\lambda_{T^{itm}}}{2}\sum_{(i,k)\in\mathbb{O}_{T^{itm}}}(T^{itm}_{ik}-\mathbf{U}_i^T\mathbf{w}^{itm}_k)^2 \\ & +\frac{\lambda_{U^i}}{2}\|\mathbf{U}^i\|^2_F+\frac{\lambda_{V^i}}{2}\|\mathbf{V}^i\|^2_F \\ & +\frac{\lambda_{w^{usr}}}{2}\|\mathbf{w}^{usr}\|^2_F+\frac{\lambda_{w^{itm}}}{2}\|\mathbf{w}^{itm}\|^2_F, \end{aligned} \quad (18)$$

where λs are regularization hyperparameters. $\mathbf{w}^{usr}$, $\mathbf{w}^{itm}$, $\mathbf{U}^i$ and $\mathbf{V}^i$ are model parameters need to be learned. g^u and g^i are the functions to map user/item id to user/item group id.

4) Advantages for the Cold-Start Problem: Recall from (9) that user i's latent features $\mathbf{U}_i$ has a content-based component $\mathbf{U}^c_i$ and a non-content-based component $\mathbf{U}^i_i$. If user i is a cold user (with few observed ratings), we can see from (18) that the regularizer term $\frac{\lambda_{U^i}}{2}\|\mathbf{U}^i\|^2_F$ may pull $\mathbf{U}^i_i$ to zero. With the group-based ratings in $\mathbf{T}^{usr}$, the $(T^{usr}_{jk}-\mathbf{U}^{cT}_i\mathbf{V}_j)^2$ term may help pull $\mathbf{U}^c_i$ to the more correct value.

5) Optimization: Stochastic gradient descent (SGD) has been commonly used for matrix factorization [9], [4]. Specifically, in each SGD iteration, an observed rating $(i,j)\in\mathbb{O}_R$ is randomly selected. Assume that user i belongs to user group k_1 and item j belongs to item group k_2. The corresponding sub-objective in (18) is

$$\begin{aligned} & \frac{1}{2}(R_{ij}-\mathbf{U}_i^T\mathbf{V}_j)^2+\frac{\lambda_{T^{usr}}}{2|\mathbb{O}_{T^{usr}_{jk_1}}|}(T^{usr}_{jk_1}-\mathbf{w}^{usrT}_{k_1}\mathbf{V}_j)^2 \\ & +\frac{\lambda_{T^{itm}}}{2|\mathbb{O}_{T^{itm}_{ik_2}}|}(T^{itm}_{ik_2}-\mathbf{U}_i^T\mathbf{w}^{itm}_{k_2})^2+\frac{\lambda_{U^i}}{2}\|\mathbf{U}^i\|^2_F \\ & +\frac{\lambda_{V^i}}{2}\|\mathbf{V}^i\|^2_F+\frac{\lambda_{w^{usr}}}{2}\|\mathbf{w}^{usr}_{k_1}\|^2_F+\frac{\lambda_{w^{itm}}}{2}\|\mathbf{w}^{itm}_{k_2}\|^2_F. \end{aligned}$$

This can be updated by gradient descent.

IV. EXPERIMENTS

In this section, we demonstrate the performance of the proposed CPMF and CPMCF models on a number of real-world recommender data sets.

A. Data Sets

The following data sets are used (Table I).

1) MovieLens:[3] The MovieLens-1M data set consists of 1M ratings (on a scale of 1 to 5) of 3,952 users on 6,040 movies. The user's demographic information and movie's genre information are provided. For demographics, there are two genders, seven age groups ("under 18", "18-24", "25-34", "35-44", "45"-49", "50-55", and "56+") and 21 occupations.
2) EachMovie:[4] The EachMovie data set contains 2,811,983 ratings (on a scale of 0 to 5) of 72,916 users on 1,628 movies. User's demographic information (without occupation) and the movie's genre information (10 genres) are provided. In this data set, sometimes the user's age/gender and/or the movie's genre are unknown. Hence, we also add "unknown age", "unknown gender" and "unknown genre" as new categorical features. Moreover, random age values are assigned to users with "unknown age".
3) Douban:[5] This data set is from [6] and contains about 16.8M ratings (on a scale of 1 to 5) of 129,490 users on 58,541 movies. User's friend list information are provided. For each user, we also add whether he is a friend of each other user as a binary content feature. The average and maximum number of friends per user are 13 and 986, respectively.
4) Netflix:[6] In 2006, the online DVD rental company Netflix announced a contest to improve the state-of-the-art recommender systems. This data set contains 100 million ratings of about 500,000 anonymous customers on more than 17,000 movies. Each movie is rated on a scale of 1 to 5 stars Each movie's release date is provided.

To be consistent with the PMF, the observed ratings on all data sets are adjusted to have zero mean.

Besides the content information provided in the data sets, we also extract some statistical features as additional content features. For each user, we add (i) the number of her observed ratings; (ii) the average value of her observed ratings; (iii) 100 Boolean features indicating if she has rated each of the Top-100 hot items. For each item, we add (i) the number of its observed ratings; (ii) the average value of its observed ratings; (iii) 100 Boolean features indicating if it has been rated by each of the Top-100 hot users.

[3]http://grouplens.org/datasets/movielens/
[4]http://grouplens.org/datasets/eachmovie/
[5]http://www.cse.cuhk.edu.hk/irwin.king/pub/data/douban
[6]http://www.netflixprize.com

B. Setup

Two different amounts (namely, 80% and 40%) of rating information are used for training[7]. Recall that hot users have a much larger number of observed ratings than cold users. To ensure all users are equally represented in the test set, we uniformly sample observed user-item ratings from the whole set into the test set.

The proposed CPMF and CPMCF will be compared with the following state-of-the-art algorithms:

1) probabilistic matrix factorization (PMF) [5];
2) matrix co-factorization (MCF) [13];
3) graph regularized matrix factorization (GRMF) [14];
4) factorization machine (FM) [15];
5) factorization machine with only using pairwise interactions between user features and item features (FM2).

To reduce statistical variability, results are averaged over 20 repetitions. Both hyperparameter[8] tuning and the early stopping of SGD are based on a validation set (of size 5% of the whole data set). Performance is evaluated based on the root mean squared error $RMSE = \sqrt{\frac{1}{T}\sum_{i,j}(R_{ij} - \hat{R}_{ij})^2}$, where R_{ij} is the rating of user i on item j, and $\hat{R}_{i,j}$ is the corresponding predicted rating.

C. Results

1) RMSE Comparison: Table II shows the RMSE with varying amounts of training data and latent feature dimensionalities. Overall, the proposed methods and FMs, which can capture nonlinear feature interactions, always outperform the others. In particular, the proposed CPMCF outperforms FM by around 0.003.

Moreover, as can be seen, FM2 and FM are very close to each other (most of their results differ by less than 0.0005). This shows that the pairwise latent interactions in user/item features are not very useful in improving overall performance. On the other hand, note that there are significant performance gaps between CPMF and FMs on the MovieLens and EachMovie data sets, and are larger than those on the Douban and Netflix data sets. We speculate that it is because the content information in Douban and Netflix are not very informative. Thus, it is not necessary to capture a high degree of nonlinearity using the proposed tree-based method.

Note that the RMSE improves more significantly when the latent feature dimensionality is increased from 5 to 20, than when it is increased from 20 to 100. In other words, using a latent feature dimensionality of 20 is usually sufficient, and will be adopted in the following experiments .

2) Cold Items/Users: In this section, we focus on the performance on cold items and cold users. Here, cold users (resp. items) are defined as users (resp. items) with fewer than 10 observed ratings. The latent feature dimensionality is set

[7]As the EachMovie data set is very sparse, 80% and 60% of the rating information are used instead.

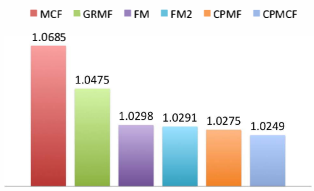
[8]For CPMCF, we simplify hyperparameter tuning by setting $\lambda_{T^{usr}} = \lambda_{T^{itm}}$ (which will be denoted as λ_T in the sequel).

TABLE I
SUMMARY OF THE DATA SETS.

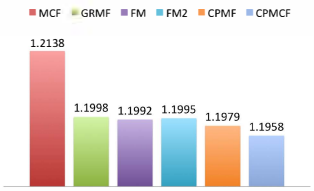
data set	#users	#items	#ratings	min #ratings per user	min #ratings per item	max #ratings per user	max #ratings per item	avg #ratings per user	avg #ratings per item	content information user	content information item
MovieLens	3,952	6,040	1M	20	1	2,314	3,428	253.1	165.6	age,gender,occupation	genre
EachMovie	72,916	1,628	2.8M	1	1	1,375	32,294	38.6	1,727.3	age,gender	genre
Douban	129,490	58,541	16.8M	1	1	1,960	7,082	12.2	7.6	friend list	/
Netflix	480,189	17,770	100M	1	3	17,653	232,944	209.3	5,654.5	/	release date

TABLE II
RMSES FOR DIFFERENT AMOUNTS OF TRAINING DATA AND LATENT FEATURE DIMENSIONALITIES. (THE STANDARD DEVIATION IS ALWAYS SMALLER THAN 0.0006, AND SO NOT REPORTED)

data set	amount of data for training	dim	PMF	MCF	GRMF	FM	FM2	CPMF	CPMCF
MovieLens	80%	5	0.9265	0.9185	0.9162	0.9135	0.9139	0.9124	**0.9112**
		20	0.9146	0.9122	0.9052	0.9022	0.9019	0.9013	**0.9003**
		100	0.9137	0.9115	0.9045	0.9020	0.9018	0.9012	**0.9001**
	40%	5	0.9801	0.9694	0.9689	0.9558	0.9559	0.9545	**0.9536**
		20	0.9762	0.9668	0.9675	0.9514	0.9517	0.9508	**0.9495**
		100	0.9757	0.9665	0.9669	0.9513	0.9510	0.9502	**0.9494**
EachMovie	80%	5	1.1176	1.1060	1.1058	1.1043	1.1045	1.1034	**1.1022**
		20	1.1100	1.0999	1.0995	1.0992	1.0998	1.0984	**1.0975**
		100	1.1085	1.0979	1.0992	1.0990	1.0992	1.0975	**1.0958**
	60%	5	1.1406	1.1220	1.1194	1.1157	1.1164	1.1137	**1.1121**
		20	1.1358	1.1178	1.1162	1.1117	1.1119	1.1108	**1.1093**
		100	1.1352	1.1170	1.1161	1.1112	1.1112	1.1105	**1.1087**
Douban	80%	5	0.7325	0.7268	0.7243	0.7194	0.7189	0.7187	**0.7162**
		20	0.7218	0.7175	0.7158	0.7122	0.7123	0.7118	**0.7102**
		100	0.7209	0.7173	0.7145	0.7112	0.7114	0.7110	**0.7091**
	40%	5	0.7552	0.7452	0.7421	0.7399	0.7395	0.7394	**0.7352**
		20	0.7503	0.7423	0.7405	0.7362	0.7352	0.7351	**0.7315**
		100	0.7499	0.7418	0.7404	0.7355	0.7350	0.7350	**0.7313**
Netflix	80%	5	0.9258	0.9240	0.9222	0.9199	0.9197	0.9195	**0.9181**
		20	0.9172	0.9162	0.9153	0.9134	0.9135	0.9130	**0.9115**
		100	0.9158	0.9151	0.9143	0.9117	0.9120	0.9114	**0.9096**
	40%	5	0.9531	0.9512	0.9502	0.9492	0.9488	0.9484	**0.9472**
		20	0.9467	0.9443	0.9440	0.9428	0.9430	0.9426	**0.9412**
		100	0.9462	0.9442	0.9437	0.9425	0.9425	0.9425	**0.9404**

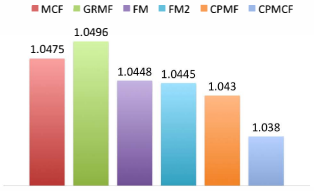


(a) MovieLens.



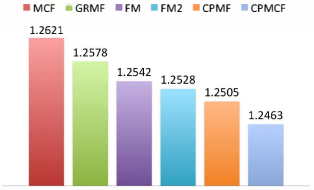
(b) EachMovie.

(c) Douban.

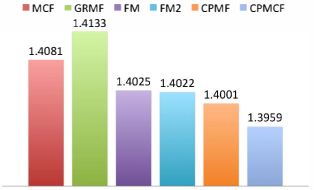


(d) Netflix.

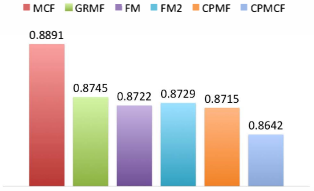
Fig. 4. RMSEs for the cold users.



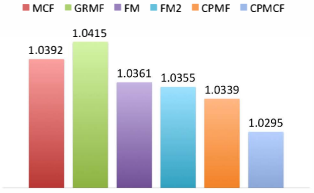
(a) MovieLens.



(b) EachMovie.



(c) Douban.

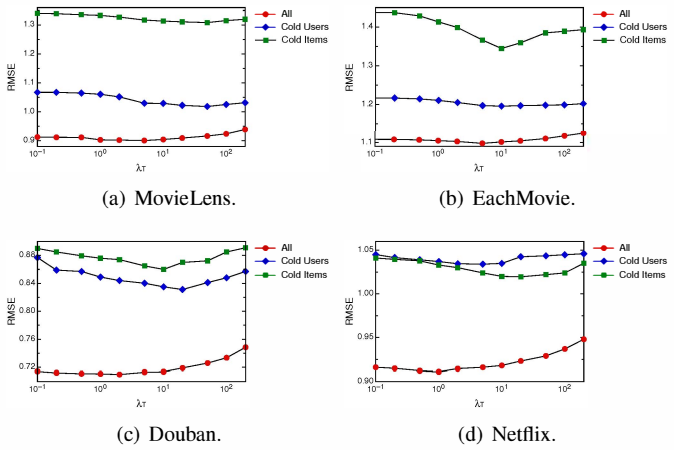


(d) Netflix.

Fig. 5. RMSEs for the cold items.

to 20, and 40% (resp. 60%) of the data are used for training for MovieLens, Douban, Netflix (resp. EachMovie).

Figures 4 and 5 show the RMSEs for the cold users and cold items, respectively. As can be seen, the RMSE improvements of CPMCF over FM (of around 0.05) for the subset of cold users/items are much larger than those on the full set This shows, as expected, that cold users/items can benefit more from the use of content information than users/items with suf-



(a) MovieLens. (b) EachMovie.

(c) Douban. (d) Netflix.

Fig. 6. RMSE at different λ_T's. Here, the latent feature dimensionality is 20, and 80% of the data are used for training.

ficient observed ratings. In particular, CPMCF is again better than CPMF. This is because not only can the cold users/items benefit from their group information via the regularization terms $\|\mathbf{V}^i\|_F^2$ and $\|\mathbf{U}^i\|_F^2$, but also the co-factorization of group-based rating matrices can also help.

3) Effect of Co-Factorization: In CPMCF, the λ_T parameter plays an important role as it controls how much the model should fit the group-based rating matrices $\mathbf{T}^{usr}$ and $\mathbf{T}^{itm}$. With a small λ_T, we only add the prior means $\mathbf{U}^c$ and $\mathbf{V}^c$ to the constrained PMF but does not require fitting $\mathbf{T}^{usr}$ and $\mathbf{T}^{itm}$; whereas with a large λ_T, the fitting of $\mathbf{T}$s dominate the learning process and the training loss of the observed rating information $\mathbf{R}$ is ignored.

Figure 6 shows how λ_T affects the prediction accuracy (with a latent feature dimensionality of 20). As can be seen, as λ_T increases, the RMSE decreases first, but then increases with an increase in λ_T. This confirms our intuition that co-factorizing the raw rating matrix and group-based rating matrices simultaneously achieve better performance than using either only $\mathbf{R}$ or $\mathbf{T}$s. For the cold users/items, the best λ_T is usually larger than that for the full set. We speculate that it is because the group-based rating information is more helpful for cold users/items.

V. Conclusion

In this paper, we proposed the combination of a novel tree-based feature group learning method and a novel constrained probabilistic matrix co-factorization model. This extends the state-of-the-art factorization machine on the rating prediction problem in recommender systems. We alleviated some key limitations of applying factorization machine. Experimental results show that the proposed method outperforms a number of baseline methods on several real-world data sets, with results particularly encouraging on the cold users and cold items.

Acknowledgment

This research was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region (Grant 614513).

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[2] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[3] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, p. 4, 2009.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[5] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances on Neural Information Processing Systems*, 2007, pp. 1257–1264.

[6] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, 2011, pp. 287–296.

[7] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *Proceedings of the 18th International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 1267–1275.

[8] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the 4th ACM Conference on Recommender Systems*, 2010, pp. 135–142.

[9] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 426–434.

[10] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th International Conference on Information and Knowledge Management*, 2008, pp. 931–940.

[11] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 650–658.

[12] N. Mirbakhsh and C. X. Ling, "Clustering-based matrix factorization," *arXiv preprint arXiv:1301.6659*, 2013.

[13] C. Lippert, S. H. Weber, Y. Huang, V. Tresp, M. Schubert, and H.-P. Kriegel, "Relation prediction in multi-relational domains using matrix factorization," in *Proceedings of the NIPS Workshop on Structured Input-Structured Output*, 2008.

[14] Q. Gu, J. Zhou, and C. Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs," in *Proceedings of the International Conference on Data Mining*, 2010, pp. 199–210.

[15] S. Rendle, "Factorization machines," in *Proceedings of the International Conference on Data Mining*, 2010, pp. 995–1000.

[16] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*. Springer, 2011, pp. 217–253.

[17] Y. Fang and L. Si, "Matrix co-factorization for recommendation with rich side information and implicit feedback," in *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, 2011, pp. 65–69.

[18] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 19–28.