Weighted Nonnegative Matrix Co-Tri-Factorization for Collaborative Prediction

Jiho Yoo and Seungjin Choi

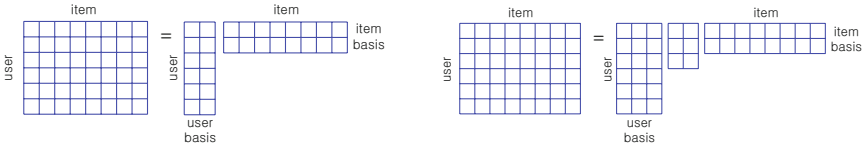
Department of Computer Science
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Korea
{zentasis,seungjin}@postech.ac.kr

Abstract. Collaborative prediction refers to the task of predicting user preferences on the basis of ratings by other users. Collaborative prediction suffers from the *cold start problem* where predictions of ratings for new items or predictions of new users' preferences are required. Various methods have been developed to overcome this limitation, exploiting side information such as content information and demographic user data. In this paper we present a matrix factorization method for incorporating side information into collaborative prediction. We develop Weighted Nonnegative Matrix Co-Tri-Factorization (WNMCTF) where we jointly minimize weighted residuals, each of which involves a nonnegative 3-factor decomposition of target or side information matrix. Numerical experiments on MovieLens data confirm the useful behavior of WNMCTF when operating from a cold start.

1 Introduction

Weighted nonnegative matrix factorization (NMF) is a method for low-rank approximation of nonnegative data, where the target matrix is approximated by a product of two nonnegative factor matrices (2-factor decomposition) [1,2]. Various extensions and modifications of NMF have been studied in machine learning, data mining, and pattern recognition communities. Orthogonality constraints on factor matrices were additionally considered to improve the clustering performance of NMF, leading to orthogonal NMF [3,4,5]. A variety of divergences were employed as an error function to yield different characteristics of learning machines, including α-divergence [6], Bregman divergence [7], and generalized divergences [8]. The 3-factor decomposition was incorporated into NMF, where a nonnegative data matrix is decomposed into a product of 3 nonnegative factor matrices, referred to as *nonnegative matrix tri-factorization* [3]. Probabilistic matrix tri-factorization [9] was recently developed, which is closely related to nonnegative matrix tri-factorization. Nonnegative Tucker decomposition (NTD) [10,11] is a multiway generalization of NMF, where nonnegativity constraints on mode matrices and a core tensor are incorporated into Tucker decomposition. In fact, nonnegative matrix tri-factorization or nonnegative 3-factor decomposition

Z.-H. Zhou and T. Washio (Eds.): ACML 2009, LNAI 5828, pp. 396–411, 2009.



(a) (b)

Fig. 1. A user-item matrix is factored into a product of user-related and item-related factor matrices: (a) 2-factor decomposition; (b) 3-factor decomposition. In the 2-factor decomposition, the target matrix $\boldsymbol{X}$ is decomposed into a product of two factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ such that $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are referred to as left and right factor matrices, respectively. In 2-factor decomposition, the number of columns in both $\boldsymbol{U}$ and $\boldsymbol{V}$ should be equal so that the target matrix is represented by a sum of outer products between corresponding columns in $\boldsymbol{U}$ and $\boldsymbol{V}$ which are related by one-to-one mapping. On the other hand, the 3-factor decomposition assumes $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ where a non-square factor matrix is introduced in the center, allowing different numbers of columns in $\boldsymbol{U}$ and $\boldsymbol{V}$ so that columns in $\boldsymbol{U}$ and $\boldsymbol{V}$ are interacted to represent $\boldsymbol{X}$. In this sense, the 3-factor decomposition captures more complex relations, compared to the 2-factor decomposition. In addition, in the 3-factor decomposition, the factor matrix in the center absorbs scaling effect when the other two factor matrices are normalized.

is a special case of NTD, where only 2-way is considered. Pictorial illustration for 2-factor and 3-factor decomposition is shown in Fig. 1.

A large body of past work on NMF has focused on the case of complete data matrix where all entries are observed without missing values. In practice, however, the data matrix is often incomplete with some of entries are missing or unobserved. For instance, most of the entries in user-rating matrix are zeros (unobserved), so that matrix completion is necessary to predict unobserved ratings, which becomes a popular approach to *collaborative prediction* [12,13,14]. Weighted nonnegative matrix factorization (WNMF) seeks a nonnegative 2-factor decomposition of the target matrix by minimizing weighted residuals [15,14,16]. With zero/one weights, only observed entries in the target matrix are considered in the decomposition and unobserved entries are estimated by learned factor matrices. While WNMF was successfully applied to collaborative prediction [14,16], its performance is degraded when the number of rated items is very small and it even fails when operating from a *cold start* where corresponding items are rated by none of users in the data.

Side information such as demographic user data and content information helps bridge the gap between existing items (or users) and new (cold start) items (or users). Various methods for blending pure collaborative filtering with content-based filtering have been developed. For example, these include content-boosted collaborative filtering [17], probabilistic models [18,19], pairwise kernel methods [20], and filterbots-based method [21] to name a few. A method based on the predictive discrete latent factor models was proposed [22], which makes use of the additional information in the form of pre-specified covariates. Recent advances in

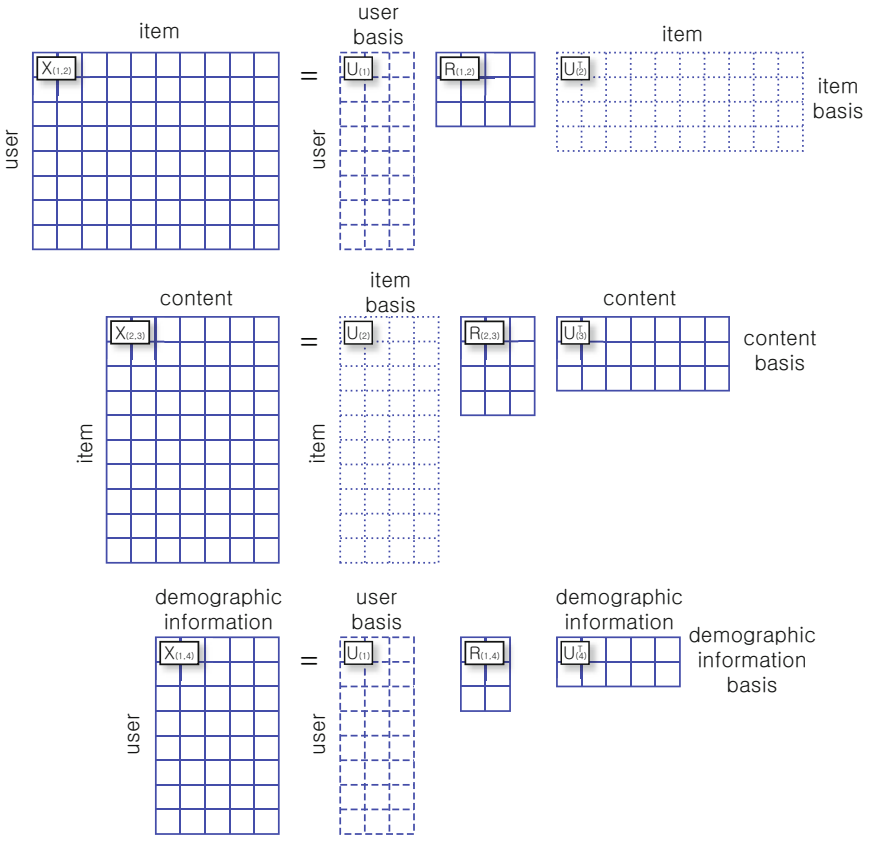


Fig. 2. Co-tri-factorizations of three matrices (user-item matrix, item-genre matrix, and user-demography matrix) are illustrated. User-related factor matrices are shared in the 1st and 3rd decompositions and item-related factor matrices are shared in the 1st and 2nd decompositions.

matrix factorization methods suggest *collective matrix factorization* or *matrix co-factorization* to incorporate side information, where several matrices (target and side information matrices) are simultaneously decomposed, sharing some factor matrices. Matrix co-factorization methods have been developed to incorporate label information [23], link information [24], and inter-subject variations [25]. Collective matrix factorization for relational learning was recently studied [26] Most of the existing methods are based on collective 2-factor decompositions. Some limitations of the 2-factor decomposition are described in Fig. 1 and they carry over to collective 2-factor decompositions.

Matrix tri-factorization or nonnegative 3-factor decomposition has been recently studied for co-clustering [3,27,9]. In this paper, we present a method for weighted nonnegative matrix co-tri-factorization (WNMCTF) where several

matrices are simultaneously factored by 3-factor decompositions through minimizing weighted residuals, sharing some factor matrices. Pictorial illustration for matrix co-tri-factorization is shown in Fig. 2. Some useful aspects of 3-factor decomposition (as illustrated in Fig. 1) carry over to WNMCTF. Moreover, we highlight why WNMCTF is preferred over existing collective matrix factorizations or weighted nonnegative matrix co-factorization (WNMCF).

- WNMCTF includes existing nonnegative matrix co-factorization methods as special cases.
- As in the 3-factor decompositions where many-to-many interactions between column vectors in factor matrices are allowed to approximate the target matrix, more complex relations are captured by WNMCTF, compared to WNMCF where only one-to-one interactions between column vectors in factor matrices are allowed.
- The non-square factor matrix in the middle allows different numbers of bases for the left and right factor matrices. It absorbs scaling effect so that the normalization for factor matrices in WNMCTF is much easier, compared to existing collective matrix factorizations.
- WNMCTF can easily handle multi-relational data. For example, it can handle cold-start users and cold-start items at the same time, without further careful manipulation. This is mainly due to the easiness of normalization for factor matrices which are shared in the decompositions.

The rest of this paper is organized as follows. We begin with weighted nonnegative matrix co-factorization which is collective 2-factor decomposition in Section 2. Section 3 presents WNMCTF and corresponding multiplicative updating algorithms. Consistency for shared factor matrices is considered in developing the algorithm for WNMCTF. Numerical experiments for cold-start users and cold-start users/items are provided in Section 4, confirming the useful behavior of WNMCTF over WNMCF. Conclusions are drawn in Section 5.

2 Weighted Nonnegative Matrix Co-Factorization

The classical *learning from data* have been focusing on a feature-based representation of data, which is usually given by a data-by-feature matrix. However, in many practical situations, the data is given by a set of matrices representing the pairwise relations between multiple objects in the domain. The natural way to model this kind of *relational data* is using the entity-relationship model, which represents the data as the relationships between the entities. For example, in the collaborative prediction of movie ratings, some examples of entity types are user, movie, genre, actor, user age and occupation. The exemplary relationships between the entity types are user's ratings of movies, actor's roles in movies, and user's employments in occupations. The set of all n number of entity types in consideration is denoted by $\mathcal{E} = \{e_1, e_2, ..., e_n\}$, where each element e_i represents the i-th entity type. We assume that the binary relationship between two entity types e_i and e_j is given by the form of a nonnegative matrix $\boldsymbol{X}_{(i,j)} \in \mathcal{X}$, where

$\mathcal{X}$ is the set of all given input matrices. Then, the set of all pairwise relations can be defined as $\mathcal{R} = \{(i,j)|\text{if } \boldsymbol{X}_{(i,j)} \text{ exists in } \mathcal{X}\}$.

Basically, WNMCF factors each nonnegative relation matrix $\boldsymbol{X}_{(i,j)}$ with the product of the nonnegative basis matrices $\boldsymbol{U}_{(i)}$ of entity type e_i and $\boldsymbol{U}_{(j)}$ of e_j. That is, we find the matrices $\boldsymbol{U}_{(i)}$ and $\boldsymbol{U}_{(j)}$ such that

$$\boldsymbol{X}_{(i,j)} = \boldsymbol{U}_{(i)}\boldsymbol{U}_{(j)}^{\top}, \text{ for all } (i,j) \in \mathcal{R}. \tag{1}$$

We share the same basis matrix $\boldsymbol{U}_{(i)}$ for all the pairwise relationships $(i,\cdot) \in \mathcal{R}$, that the corresponding entity type is involved in. If we denote the number of instances in the entity type e_i as $|e_i|$, the input matrix $\boldsymbol{X}_{(i,j)} \in \mathbb{R}_{+}^{|e_i|\times|e_j|}$. The dimensions of factor matrices are determined by the pre-specified number of basis $|b|$, such that $\boldsymbol{U}_{(i)} \in \mathbb{R}_{+}^{|e_i|\times|b|}$.

The probabilistic interpretation of the factorization (1) is based on the following parametrization of the joint probability $p(e_i, e_j)$ between the entity types,

$$p(e_i, e_j) = \sum_b p(e_i|b)p(e_j|b)p(b).$$

By applying appropriate normalization for the matrices in (1) [28], the matrix $\boldsymbol{X}_{(i,j)}$ corresponds to the joint probability $p(e_i, e_j)$ of two entity types, and each basis matrix $\boldsymbol{U}_{(i)}$ can be interpreted as the conditional probability $p(e_i|b)$.

The input matrix $\boldsymbol{X}_{(i,j)}$ might have missing (unobserved) values in it. To jointly factorize all such $\boldsymbol{X}_{(i,j)}$ into the form of (1), we use the objective function based on the weighted divergence, which can be formulated as

$$\mathcal{L} = \frac{1}{2} \sum_{(i,j)\in\mathcal{R}} \alpha_{(i,j)} \left\| \boldsymbol{W}_{(i,j)} \odot \left(\boldsymbol{X}_{(i,j)} - \boldsymbol{U}_{(i)}\boldsymbol{U}_{(j)}^{\top} \right) \right\|_F^2 \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\odot$ represents the Hadamard product (element-wise multiplication). $\boldsymbol{W}_{(i,j)}$ is the weight matrix consists of the value one for the observed input values in $\boldsymbol{X}_{(i,j)}$, and the value zero for the unobserved ones. $\alpha_{(i,j)}$'s are the scalar parameters adjusting different scales between the data matrices. The above objective function represents the difference between the observed input data and reconstructed data.

The derivative of the objective function with respect to the basis matrix $\boldsymbol{U}_{(i)}$ can be calculated as

$$\begin{aligned} \nabla_{U_{(i)}}\mathcal{L} = &- \sum_{(i,j)\in\mathcal{R}} \left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \boldsymbol{X}_{(i,j)} \right) \boldsymbol{U}_{(j)} \\ &+ \sum_{(i,j)\in\mathcal{R}} \left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \left(\boldsymbol{U}_{(i)}\boldsymbol{U}_{(j)}^{\top} \right) \right) \boldsymbol{U}_{(j)}, \end{aligned} \tag{3}$$

where $\widehat{\boldsymbol{W}}_{(i,j)} = \alpha_{(i,j)}\boldsymbol{W}_{(i,j)}$. This gradient of the objective function can be decomposed into the form

$$\nabla\mathcal{L} = \left[\nabla_{U_{(i)}}\mathcal{L}\right]^{+} - \left[\nabla_{U_{(i)}}\mathcal{L}\right]^{-}, \tag{4}$$

where $\left[\nabla_{U_{(i)}}\mathcal{L}\right]^+ > 0$ and $\left[\nabla_{U_{(i)}}\mathcal{L}\right]^- > 0$. From this decomposition, we can construct the multiplicative update for $\boldsymbol{U}_{(i)}$ as

$$\boldsymbol{U}_{(i)} \leftarrow \boldsymbol{U}_{(i)} \odot \frac{\left[\nabla_{U_{(i)}}\mathcal{L}\right]^-}{\left[\nabla_{U_{(i)}}\mathcal{L}\right]^+}. \tag{5}$$

It can be easily seen that the above multiplicative update preserves the nonnegativity of $\boldsymbol{U}_{(i)}$, while $\nabla_{U_{(i)}}\mathcal{L} = 0$ when the convergence is achieved. Therefore, the multiplicative update rules for the basis matrix $\boldsymbol{U}_{(i)}$ can be derived as follows,

$$\boldsymbol{U}_{(i)} \leftarrow \boldsymbol{U}_{(i)} \odot \frac{\sum_{(i,j)\in\mathcal{R}} \left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \boldsymbol{X}_{(i,j)}\right) \boldsymbol{U}_{(j)}}{\sum_{(i,j)\in\mathcal{R}} \left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \boldsymbol{U}_{(i)}\boldsymbol{U}_{(j)}^\top\right) \boldsymbol{U}_{(j)}}. \tag{6}$$

Update rule for $\boldsymbol{U}_{(j)}$ can be derived in the similar way, or we can transpose the relation into $\boldsymbol{X}_{(j,i)} = \boldsymbol{U}_{(j)}\boldsymbol{U}_{(i)}^\top$ and apply the above update rule.

3 Weighted Nonnegative Matrix Co-Tri-Factorization

Now we will introduce our method, weighted nonnegative matrix co-tri-factorization model and the iterative update algorithm with consistency constraints. As in the case of WNMCF, WNMCTF jointly factorizes given input matrices, but other than the two-factor decomposition (1), it applies three-factor decomposition model for each input matrix $\boldsymbol{X}_{(i,j)}$,

$$\boldsymbol{X}_{(i,j)} = \boldsymbol{U}_{(i)}\boldsymbol{S}_{(i,j)}\boldsymbol{U}_{(j)}^\top, \text{ for all } (i,j) \in \mathcal{R}, \tag{7}$$

where the full matrix $\boldsymbol{S}_{(i,j)}$ is put between the two-factor decomposition model. The new matrix $\boldsymbol{S}_{(i,j)}$ represents the many-to-many relationships between the bases in the matrices $\boldsymbol{U}_{(i)}$ and $\boldsymbol{U}_{(j)}$, that is, a basis in the matrix $\boldsymbol{U}_{(i)}$ can be multiplied to any basis in $\boldsymbol{U}_{(j)}$ through the matrix $\boldsymbol{S}_{(i,j)}$. In the two-factor decomposition model, an i-th basis in $\boldsymbol{U}_{(i)}$ can be multiplied only to the corresponding i-th basis in $\boldsymbol{U}_{(j)}$, so only one-to-one relationship between the basis can be modeled. Moreover, the matrix $\boldsymbol{S}_{(i,j)}$ is not restricted to be a square matrix, therefore, we can set the number of basis for entity types differently. If we denote the number of basis for the entity type e_i as $|b_i|$, then $\boldsymbol{U}_{(i)} \in \mathbb{R}_+^{|e_i|\times|b_i|}$, $\boldsymbol{U}_{(j)} \in \mathbb{R}_+^{|e_j|\times|b_j|}$ and $\boldsymbol{S}_{(i,j)} \in \mathbb{R}_+^{|b_i|\times|b_j|}$. Note that the two-factor model (1) introduced in the previous section can be considered as a special case of this model, where the number of bases $|b_i|$ are restricted to be the same ($|b|$) for all entities, and all matrices $S_{(i,j)}$ are restricted to be square and diagonal.

The probabilistic interpretation of the tri-factorization (7) can be followed by the probability decomposition

$$p(e_i, e_j) = \sum_{b_i,b_j} p(e_i|b_i)p(e_j|b_j)p(b_i, b_j). \tag{8}$$

If appropriately normalized, the joint probability $p(e_i, e_j)$ corresponds to the matrix $\boldsymbol{X}_{(i,j)}$, and the conditional probabilities $p(e_i|b_i)$ and $p(e_j|b_j)$ to $\boldsymbol{U}_{(i)}$ and $\boldsymbol{U}_{(j)}$, respectively. The joint probability between the bases, $p(b_i, b_j)$ corresponds to $\boldsymbol{S}_{(i,j)}$.

To jointly factorize all $\boldsymbol{X}_{(i,j)}$ into the form of (7), we introduce the following objective function,

$$\begin{aligned}\mathcal{L} = \frac{1}{2} \sum_{(i,j)\in\mathcal{R}} \alpha_{(i,j)} \left\| \boldsymbol{W}_{(i,j)} \odot \left(\boldsymbol{X}_{(i,j)} - \boldsymbol{U}_{(i)} \boldsymbol{S}_{(i,j)} \boldsymbol{U}_{(j)}^\top \right) \right\|_F^2 \\ + \frac{1}{2} \sum_i \sum_{\substack{\{i,j\}\in\mathcal{R} \\ \{i,k\}\in\mathcal{R}\setminus\{i,j\}}} \gamma \left\| \frac{\boldsymbol{S}_{(i,j)}\mathbf{1}_{(j)}}{C_{(i,j)}} - \frac{\boldsymbol{S}_{(i,k)}\mathbf{1}_{(k)}}{C_{(i,k)}} \right\|^2, \end{aligned} \tag{9}$$

where $\mathbf{1}_{(i)}$ represents an $|e_i|$-dimensional vector of one's, $\|\cdot\|$ represents the vector 2-norm, and $C_{(i,j)} = \mathbf{1}_{(i)}^\top \boldsymbol{S}_{(i,j)} \mathbf{1}_{(j)}$. We used the braced pair $\{i, j\}$ to represent an un-ordered pair of i and j, that is, (i, j) or (j, i) for the notational brevity. The first term is defined in the same way as WNMCF objective function (2), only with different decomposition model. The second term is introduced to impose the consistency between the interpretation of the resulting marginal probability of the basis, $p(b_i)$. The probability $p(b_i)$ can be obtained implicitly from the resulting matrix $\boldsymbol{S}_{(i,j)}$ by applying appropriate normalization and summing with respect to b_j, that is,

$$\boldsymbol{p}(b_i) = \frac{\boldsymbol{S}_{(i,j)}\mathbf{1}_{(j)}}{\mathbf{1}_{(i)}^\top \boldsymbol{S}_{(i,j)}\mathbf{1}_{(j)}}, \tag{10}$$

in the vector form $\boldsymbol{p}(b_i)$ of $p(b_i)$. The marginal probability $p(b_i)$ can be calculated from several $\boldsymbol{S}_{(i,j)}$'s involved with the entity type e_i, but these probabilities are not guaranteed to be consistent. By minimizing the third term, we can reduce the difference between the marginal probabilities computed from different $\boldsymbol{S}_{(i,j)}$'s involved. The parameter γ determines the amount of this consistency constraint. In the objective function, we treat the denominator $\mathbf{1}_{(i)}^\top \boldsymbol{S}_{(i,j)} \mathbf{1}_{(j)}$ as a constant $C_{(i,j)}$, because it only depends on the scale of the input matrix $\boldsymbol{X}_{(i,j)}$ if we apply the normalization of the factor matrices which will be explained in the next section.

The derivative of the objective function with respect to the basis matrix $\boldsymbol{U}_{(i)}$ is

$$\begin{aligned}\nabla_{U_{(i)}} \mathcal{L} = &- \sum_{\{i,j\}\in\mathcal{R}} \left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \boldsymbol{X}_{(i,j)} \right) \boldsymbol{U}_{(j)} \boldsymbol{S}_{(i,j)}^\top \\ &+ \sum_{\{i,j\}\in\mathcal{R}} \left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \left(\boldsymbol{U}_{(i)} \boldsymbol{S}_{(i,j)} \boldsymbol{U}_{(j)}^\top \right) \right) \boldsymbol{U}_{(j)} \boldsymbol{S}_{(i,j)}^\top, \end{aligned} \tag{11}$$

where $\widehat{\boldsymbol{W}}_{(i,j)} = \alpha_{(i,j)} \boldsymbol{W}_{(i,j)}$. The derivative with respect to $\boldsymbol{U}_{(j)}$ can be computed in the similar way.

The derivative for $\boldsymbol{S}_{(i,j)}$ can be computed as follows,

$$\begin{aligned}\nabla_{S_{(i,j)}}\mathcal{L} = &-\boldsymbol{U}_{(i)}^{\top}\left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \boldsymbol{X}_{(i,j)}\right)\boldsymbol{U}_{(j)} \\ &+\boldsymbol{U}_{(i)}^{\top}\left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \left(\boldsymbol{U}_{(i)}\boldsymbol{S}_{(i,j)}\boldsymbol{U}_{(j)}^{\top}\right)\right)\boldsymbol{U}_{(j)} \\ &+\gamma'(m_{(i|j)} - \overline{m}_{(i)}) + \gamma'(m_{(j|i)} - \overline{m}_{(j)}) \end{aligned} \tag{12}$$

where the parameter $\gamma' = N\gamma$, $m_{(i|j)} = \left(\boldsymbol{S}_{(i,j)}\mathbf{1}_{(j)}\mathbf{1}_{(j)}^{\top}\right)/C_{(i,j)}$ and $\overline{m}_{(i)} = \frac{1}{N}\sum_{(i,k)\in\mathcal{R}\setminus(i,j)}\left((\boldsymbol{S}_{(i,k)}\mathbf{1}_{(k)}\mathbf{1}_{(j)}^{\top})/C_{(i,k)}\right)$ with number of relations N in which e_i is involved.

The above derivatives can be decomposed as in (4), therefore we can build the multiplicative update rule for the matrices $\boldsymbol{U}_{(i)}$ and $\boldsymbol{S}_{(i,j)}$ by using the similar way to (5). Then the rules become,

$$\boldsymbol{U}_{(i)} \leftarrow \boldsymbol{U}_{(i)} \odot \frac{\sum_{(i,j)\in\mathcal{R}}\left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \boldsymbol{X}_{(i,j)}\right)\boldsymbol{U}_{(j)}\boldsymbol{S}_{(i,j)}^{\top}}{\sum_{(i,j)\in\mathcal{R}}\left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \left(\boldsymbol{U}_{(i)}\boldsymbol{S}_{(i,j)}\boldsymbol{U}_{(j)}^{\top}\right)\right)\boldsymbol{U}_{(j)}\boldsymbol{S}_{(i,j)}^{\top}} \tag{13}$$

$$\boldsymbol{S}_{(i,j)} \leftarrow \boldsymbol{S}_{(i,j)} \odot \frac{\boldsymbol{U}_{(i)}^{\top}\left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \boldsymbol{X}_{(i,j)}\right)\boldsymbol{U}_{(j)} + \gamma\left(\overline{m}_{(i)} + \overline{m}_{(j)}\right)}{\boldsymbol{U}_{(i)}^{\top}\left(\widehat{\boldsymbol{W}}_{(i,j)} \odot \left(\boldsymbol{U}_{(i)}\boldsymbol{S}_{(i,j)}\boldsymbol{U}_{(j)}^{\top}\right)\right)\boldsymbol{U}_{(j)} + \gamma\left(m_{(i|j)} + m_{(j|i)}\right)}, \tag{14}$$

where we denote the parameter γ' simply as γ for the notational simplicity.

We can prove the convergence of the above algorithm without consistency term by adapting the auxiliary function used in the convergence proof of NMF [2]. The critical part of the auxiliary function for WNMCTF which should be positive semi-definite can be formed as a sum of the positive semi-definite parts of the auxiliary function of NMF, hence becomes positive semi-definite. By vectorizing 3-factor decomposition model, the update rule for $\boldsymbol{S}$ can be formed in a 2-factor decomposition model, so the same auxiliary function method can be applied if we do not concern about the additional consistency term. Although the convergence of the algorithm with consistency term in (14) is not proved in this way, but at least the update rule (13) does not increase the objective function, and the main part of the update rule (14) also does.

3.1 Normalization of Factor Matrices

The two-factor decomposition model (1) aims to learn the basis vectors for the data and encodings for the vectors, from the input matrix consisting of data and corresponding feature values. In this case, we only have to normalize the basis matrix to have certain scale, and the encoding matrix is inversely scaled to maintain the scale of the input matrix. We assume that the input matrix is proportional to the joint probability $p(e_i, e_j)$ by the factor of $C_{(i,j)}$, that is, $\boldsymbol{X}_{(i,j)} = C_{(i,j)}\overline{\boldsymbol{X}}_{(i,j)}$, where $\overline{\boldsymbol{X}}_{(i,j)}$ consists of the probability $p(e_i, e_j)$. The scale

$C_{(i,j)}$ goes into the encoding matrix if we normalize the basis matrix to have the probabilistic scale. In other words, in the two-factor decomposition model, we cannot normalize basis and encoding matrices simultaneously in the same probabilistic scale.

The three-factor decomposition model (7) arose to learn from the dyadic data, that is, the input matrix represents the relations between two sets of objects (entities). Each factor matrix has the same semantic in the domain, and therefore should be normalized in the uniform scale. Uniform normalization of the factor matrices brings clear interpretation of the factor matrices, and is also useful to impose the additional constraints such as orthogonality on the factor matrices. If we apply the three-factor decomposition of the input matrix scaled by $C_{(i,j)}$ from its probabilistic scale, we can get the following normalized results,

$$\boldsymbol{X}_{(i,j)} = \boldsymbol{U}_{(i)} \boldsymbol{S}_{(i,j)} \boldsymbol{U}_{(j)}^{\top} \tag{15}$$

$$= \overline{\boldsymbol{U}}_{(i)} \left(C_{(i,j)} \overline{\boldsymbol{S}}_{(i,j)} \right) \overline{\boldsymbol{U}}_{(j)}^{\top}, \tag{16}$$

where $\overline{\boldsymbol{U}}_{(i)}$ is the normalized matrix from $\boldsymbol{U}_{(i)}$ having the values of $p(e_i|b_i)$, and $\overline{\boldsymbol{S}}_{(i,j)}$ is from $\boldsymbol{S}_{(i,j)}$ having $p(b_i, b_j)$, based on the following scaled probabilistic decomposition,

$$C_{(i,j)} p(e_i, e_j) = \sum_{b_i, b_j} C_{(i,j)} p(e_i|b_i) p(b_i, b_j) p(e_j|b_j) \tag{17}$$

$$= \sum_{b_i, b_j} p(e_i|b_i) \left(C_{(i,j)} p(b_i, b_j) \right) p(e_j|b_j). \tag{18}$$

Therefore, if we normalize the factor matrices to have probabilistic scale, the scale of input matrix can be absorbed in the center matrix $\boldsymbol{S}_{(i,j)}$ as $\boldsymbol{S}(i,j) = C_{(i,j)} \overline{\boldsymbol{S}}_{(i,j)}$. The matrix $\boldsymbol{S}_{(i,j)}$ is involved only in the input matrix $\boldsymbol{X}_{(i,j)}$, so this absorbtion of the scales in the matrix $\boldsymbol{S}_{(i,j)}$ does not conflict with factorizations of other input matrices.

Algorithm outline: WNMCTF

---

1. Initialize $\boldsymbol{U}_{(i)}$'s and $\boldsymbol{S}_{(i,j)}$'s with random positive values.
2. Iterate until convergence
   For each factor matrix $\boldsymbol{U}_{(i)}$,
   (a) Update $\boldsymbol{U}_{(i)}$ using the related matrices (13)
   (b) Normalize the matrix $\boldsymbol{U}_{(i)}$ to the probabilistic scale

$$\boldsymbol{U}_{(i)} \leftarrow \boldsymbol{U}_{(i)} \left(\mathrm{diag} \left(\boldsymbol{U}_{(i)} \mathbf{1}_{(i)} \right) \right)^{-1}$$

   (c) Update all $\boldsymbol{S}_{\{i,j\}}$'s related to $\boldsymbol{U}_{(i)}$ (14)

---

4 Numerical Experiments

We tested WNMCTF algorithm for the collaborative prediction problems, especially with item and user cold-start cases. We used two different datasets, MovieLens-100K which consists of the movie ratings of 943 users for 1682 movies, and MovieLens-1M which consists of the ratings of 6040 users for 3952 movies. MovieLens dataset is one of the most suitable dataset to test the algorithms because the dataset is packed with additional user and movie information. The preference of the user for a specific movie is rated by the integer score from 1 to 5, and 0 value indicates that the movie is not rated by the user.

User information consists of the demographic information about the user, including age, gender, and occupation. For the MovieLens-100K dataset, we partitioned the age into five groups, under 20, 21 to 31, 31 to 40, 41 to 50, and over 51, and mark the value of 1 for the age group of the user in the 5-dimensional age group vector. The age information of MovieLens-1M dataset is coded in the similar way, but uses seven age groups which have more smaller ranges. The gender is represented in the two dimensional vector. There are 21 occupation categories in both datasets, so we used 21-dimensional indicator vector to represent the occupations. As a result, the demographic information of MovieLens-100K dataset is written in a 943-by-28 dimensional user-demographic information matrix, and MovieLens-1M dataset in a 6040-by-30 dimensional matrix. Movie genre information matrix is built in a similar way. There are 19 genre categories in the MovieLens-100K dataset and 18 categories in the MovieLens-1M dataset, so we construct 1682-by-19 movie-genre matrix for MovieLens-100K dataset and 3952-by-18 matrix for MovieLens-1M dataset.

We have four entity types (user, movie, demographic information, and genre information) represented in the three relationship matrices (user-movie matrix, user-demographic matrix, and movie-genre matrix). We want to predict the rating values for the un-rated values in the user-movie rating matrix, using all of these information. We can reconstruct the input rating matrix as (1) or (7), and the missing target ratings are predicted with the reconstructed values. To evaluate the performance of the algorithm, we used the Mean Absolute Error (MAE) defined as follows,

$$\mathrm{MAE} = \frac{1}{N} \sum_{i=1}^{N} |r_i - \bar{r}_i| \,, \tag{19}$$

where N is the number of held-out (test) data points, r_i is the predicted ranking value for the i-th test point, and $\bar{r}_i$ is the true value for the point.

We perform the test on the three different settings. First one compares the WNMCTF algorithm with or without the additional consistency constraints to show the necessity of the constraints in the algorithm. Second one compares the WNMCTF algorithm with several algorithms, on the different number of given user ratings in the test data set. Zero given data case corresponds to the user cold-start problem, which is no ranking information is given by the user.

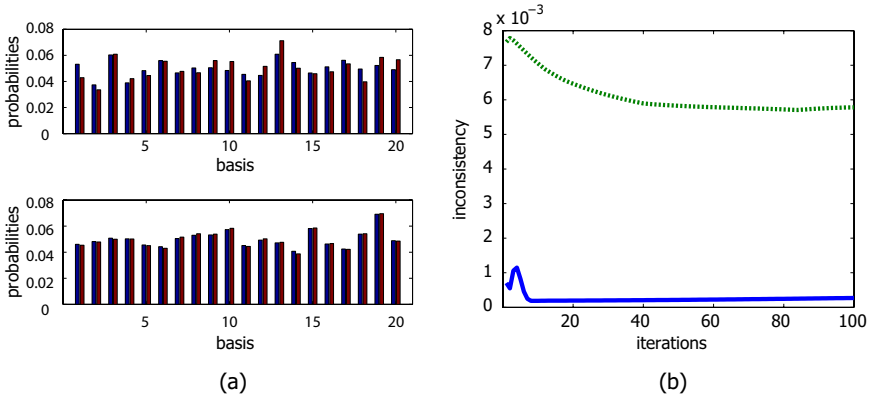
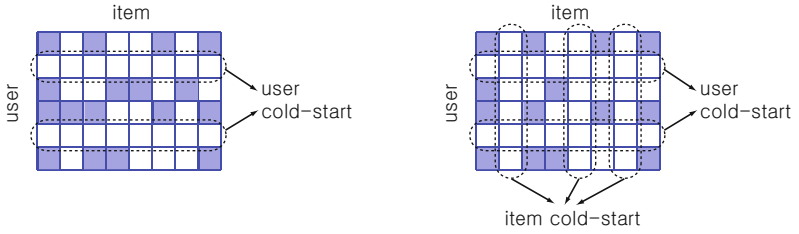


Fig. 3. The result of consistency test. (a) Comparison of two marginal probabilities computed from different relationship matrices for 20 basis components. Upper graph compares the results when the consistency constraints are not given by setting the parameter γ value to be zero, and lower graph compares the results when the consistency constraints with the parameter γ value of 10^{-5} are applied. We can obtain fairly consistent result by using the consistency constraints on WNMCTF. (b) The change of mean difference between the marginal probability elements, dotted line shows the case without consistency constraints, solid line shows the case with consistency constraints. After a few iterations, the WNMCTF algorithm with consistency constraints finds fairly consistent results.

In addition to this, the third one eliminates all given rankings for some items, to simulate the situation of item cold-start.

4.1 Consistency of the Results

First we tested the consistency of the results from WNMCTF. For the MovieLens-100K, we used the user-rating matrix and demographic information matrix in addition to the user-movie rating matrix. We set the entity types e_1 to represent user, e_2 to movie, and e_3 to demographic information. We obtained the relationships between the user bases and movie bases $\boldsymbol{S}_{(1,2)}$, and between user bases and demographic information bases $\boldsymbol{S}_{(1,3)}$. We check the difference between the marginal probabilities of the user bases $p(b_1)$ obtained from both matrices $\boldsymbol{S}_{(1,2)}$ and $\boldsymbol{S}_{(1,3)}$. By setting the parameter value of γ in (14) to be zero, we can simulate the case when the consistency constraints are not given. Fig. 3 compares the results of WNMCTF with consistency constraints and without consistency constraints. The average differences between the marginal probability values of each basis are used as a consistency measure. Without consistency constraints, the marginal probabilities appears to be much inconsistent than that with the consistency constraints.



(a) (b)

Fig. 4. Pictorial illustration for: (a) cold-start users; (b) cold-start user/items

4.2 User Cold-Start Problem

We tested six algorithms, pure collaborative filtering (PCF), pure content-based prediction (PCB), content-boosted collaborative filtering (CBCF), WNMF, WNMCF, and WNMCTF for the collaborative prediction problem. PCF is the user neighborhood-based collaborative filtering whose neighborhood size is set to 30, PCB is the naive Bayes classifier for each user trained by the movie genre information, and CBCF is a collaborative filtering method based on the predicted ratings from PCB. The details of these algorithms can be found in [17]. WNMF is implemented as a special case of WNMCF, which uses a single input user-movie rating matrix only. We randomly picked 200 users as a test set, and set some of their ratings to be zero. We tested the algorithms for different number of remaining (given) ratings for each user in the test set. In the extremal setting, no ratings were given from users in the test dataset, which is called the user cold-start case (Fig. 4 (a)). In applying WNMCF and WNMCTF, we use the user demographic information with the typical user-movie rating data to deal with this case.

We measured the MAE for the different numbers of given ratings in test set (Table 1). For each case, we performed the test on the 30 different randomly chosen testsets. As a baseline of the performance, we measured the MAE value when we set the unknown rating values as the mean of the known rating values, and call it MEAN. For the NMF-based algorithms, we used the number of bases as 20, which follows the typical settings [14,29]. To determine the suitable parameter $\alpha_{(1)}$ for the user-movie rating matrix and $\alpha_{(2)}$ for the user-demographic information matrix, we tested the algorithm with several different ratios of α, which were $(\alpha_{(1)}, \alpha_{(2)}) = (1, 0), (0.9, 0.1), \cdots, (0, 1)$. The settings $(0.8, 0.2)$ and $(0.9, 0.1)$ showed better performance than using only a single input matrix, and we used $(0.9, 0.1)$ in our experiments. γ value was also tested on the several scales, and we set γ to be 10^{-5} based on the performances.

WNMCTF showed significantly better performance for all the cases. The complex relationship captured by WNMCTF model was helpful in collaborative prediction problems. PCF, PCB, CBCF and WNMF are failed to generate appropriate predictions in the cold-start cases. PCF requires the mean of the

Table 1. Average and standard deviations of MAE measured for the different number of given ratings per user in the test dataset. The symbol * is used to indicate the cases significantly worse than the best result, based on the Wilcoxon signed-rank test with p-value 0.01.

Dataset	# Given	MEAN	PCF	PCB	CBCF	WNMF	WNMCF	WNMCTF
MovieLens-100K	0	0.9382*	-	-	-	-	0.8472*	**0.8100**
	5	0.9396*	1.0116*	1.3729*	1.3709*	0.8355*	0.8329*	**0.7929**
	10	0.9417*	0.9766*	1.2724*	1.2744*	0.7941*	0.7944*	**0.7661**
	15	0.9447*	0.9614*	1.1608*	1.1748*	0.7685*	0.7682*	**0.7543**
	20	0.9504*	0.9548*	1.0653*	1.0958*	0.7560*	0.7569*	**0.7502**
MovieLens-1M	0	0.9427*	-	-	-	-	0.8099*	**0.7800**
	5	0.9349*	0.9330*	1.4284*	1.4258*	0.8101*	0.8097*	**0.7714**
	10	0.9348*	0.8872*	1.3409*	1.3406*	0.7724*	0.7713*	**0.7452**
	15	0.9348*	0.8697*	1.2601*	1.2649*	0.7556*	0.7544*	**0.7354**
	20	0.9309*	0.8618*	1.1993*	1.2117*	0.7457*	0.7457*	**0.7306**

given ratings, which is impossible to compute when the number of given ratings is zero. PCB fails to build a classifier for the users who have no ratings, and as a result, CBCF cannot predict well for the case. WNMF completely failed to predict the ratings when the given number of ratings is 0, and produces zero values for all the cold-start users. In other cases, PCB brings worse results because the algorithm only uses very coarse content data, which is 18(or 19)-dimensional genre indicator vectors. CBCF performance are degraded because of the poor PCB results. In our experimental settings, the content information itself does not have much information to predict appropriate ratings, but we can achieve good performance by using this information in WNMCTF.

4.3 User and Item Cold-Start Problem

We tested the algorithms on more realistic case, when some movies has no ratings on them, as well as some users (Fig. 4 (b)). To deal with this situation, we jointly used movie genre information with user-movie ratings and user demographic information in WNMCF and WNMCTF.

The MAE values of the results are shown in Table 2. MEAN is the rating scheme that uses the average rating value to predict all unknown ratings. PCF and WNMF used the user-item rating matrix only, PCB and CBCF exploited the genre information to build the content-based predictions, and the WNMCF and WNMCTF co-factorized additional matrices about users and movies. We randomly chose 200 movies in the user-movie rating matrix, and set the ratings for them to be zero. To determine the $\alpha_{(3)}$ value for the movie genre matrix, we performed the similar test with different values for the user-movie rating matrix and movie genre matrix. Based on the performance, we used $(\alpha_{(1)}, \alpha_{(2)}, \alpha_{(3)}) = (0.8, 0.1, 0.1)$ in the experiments. γ is set to be 10^{-5} as in the previous section.

Table 2. Average and standard deviations of MAE measured for the different number of given ratings for test users. We eliminate all ratings for 200 randomly chosen items to simulate item cold-start cases. The symbol * is used to indicate the cases significantly worse than the best result, based on the Wilcoxon signed-rank test with p-value 0.01.

Dataset	# Given	MEAN	PCF	PCB	CBCF	WNMF	WNMCF	WNMCTF
MovieLens-100K	0	0.9442*	-	-	-	-	0.8630*	**0.8273**
	5	0.9401*	0.9902*	1.3790*	1.3766*	0.9024*	0.8460*	**0.8080**
	10	0.9477*	0.9530*	1.2768*	1.2784*	0.8361*	0.8049*	**0.7802**
	15	0.9441*	0.9471*	1.1659*	1.1780*	0.7948*	0.7733*	**0.7607**
	20	0.9431*	0.9425*	1.0784*	1.1088*	0.7708*	0.7579*	**0.7548**
MovieLens-1M	0	0.9348*	-	-	-	-	0.8214*	**0.7916**
	5	0.9348*	0.9257*	1.4419*	1.4390*	0.8527*	0.8144*	**0.7796**
	10	0.9347*	0.8842*	1.3495*	1.3477*	0.8026*	0.7785*	**0.7508**
	15	0.9347*	0.8707*	1.2678*	1.2711*	0.7761*	0.7588*	**0.7392**
	20	0.9309*	0.8589*	1.1878*	1.2003*	0.7616*	0.7483*	**0.7341**

The usefulness of co-factorization in the cold-start cases of collaborative prediction was more clearly shown by this experiment. WNMCF and WNMCTF showed better performance than WNMF in all the cases, and WNMCTF was even better than WNMCF. If more than 15 ratings are given, performances of WNMCF and WNMCTF was not degraded much from the performance showed in the previous section, but WNMF performance was seriously degraded for all the cases. Also in these experiments, PCF, PCB and CBCF cannot produce a good predictions.

5 Conclusions

We have presented WNMCTF algorithm for learning from multiple data matrices, which jointly factorizes each data matrix into the three matrices. WNMCTF can be used as a general framework for the problem having several entities and complex relationships between them. The numerical experiments on the collaborative prediction problem, which uses additional information about users and items, shows the superior performance of WNMCTF, especially on the cold-starting cases.

Acknowledgments. This work was supported by Korea Research Foundation (Grant KRF-2008-313-D00939), Korea Ministry of Knowledge Economy under the ITRC support program supervised by the IITA (IITA-2009-C1090-0902-0045), KOSEF WCU Program (Project No. R31-2008-000-10100-0), and Microsoft Research Asia.

References

1. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
2. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems (NIPS). vol. 13. MIT Press, Cambridge (2001)
3. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix tri-factorizations for clustering. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Philadelphia, PA (2006)
4. Choi, S.: Algorithms for orthogonal nonnegative matrix factorization. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN), Hong Kong (2008)
5. Yoo, J., Choi, S.: Orthogonal nonnegative matrix factorization: Multiplicative updates on Stiefel manifolds. In: Fyfe, C., Kim, D., Lee, S.-Y., Yin, H. (eds.) IDEAL 2008. LNCS, vol. 5326, pp. 140–147. Springer, Heidelberg (2008)
6. Cichocki, A., Lee, H., Kim, Y.D., Choi, S.: Nonnegative matrix factorization with α-divergence. Pattern Recognition Letters 29(9), 1433–1440 (2008)
7. Dhillon, I.S., Sra, S.: Generalized nonnegative matrix approximations with Bregman divergences. In: Advances in Neural Information Processing Systems (NIPS), vol. 18. MIT Press, Cambridge (2006)
8. Kompass, R.: A generalized divergence measure for nonnegative matrix factorization. Neural Computation 19, 780–791 (2007)
9. Yoo, J., Choi, S.: Probabilistic matrix tri-factorization. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan (2009)
10. Kim, Y.D., Choi, S.: Nonnegative Tucker decomposition. In: Proceedings of the IEEE CVPR 2007 Workshop on Component Analysis Methods, Minneapolis, Minnesota (2007)
11. Kim, Y.D., Cichocki, A., Choi, S.: Nonnegative Tucker decomposition with alpha-divergence. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Las Vegas, Nevada (2008)
12. Srebro, N., Jaakkola, T.: Weighted low-rank approximation. In: Proceedings of the International Conference on Machine Learning (ICML), Washington DC (2003)
13. Rennie, J.D.M., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: Proceedings of the International Conference on Machine Learning (ICML), Bonn, Germany (2005)
14. Zhang, S., Wang, W., Ford, J., Makedon, F.: Learning from incomplete ratings using non-negative matrix factorization. In: Proceedings of the SIAM International Conference on Data Mining, SDM (2006)
15. Mao, Y., Saul, L.K.: Modeling distances in large-scale networks by matrix factorization. In: Proceedings of the ACM Internet Measurement Conference, Taormina, Sicily, Italy (2004)
16. Kim, Y.D., Choi, S.: Weighted nonnegative matrix factorization. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan (2009)
17. Melville, P., Mooney, R.J., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendations. In: Proceedings of the AAAI National Conference on Artificial Intelligence (AAAI), Edmonton, Canada, pp. 187–192 (2002)

18. Popescul, A., Ungar, L.H., Pennock, D.M., Lawrence, S.: Probabilistic models for unified collaborative and content-based recommendation in sparse-data environment. In: Proceedings of the International Conference on Uncertainty in Artificial Intelligence, UAI (2001)
19. Schein, A.I., Popescul, A., Pennock, D.M.: Generative models for cold-start recommendations. In: Proceedings of the SIGIR Workshop on Recommender Systems, New Orleans, LA (2001)
20. Basilico, J., Hofmann, T.: Unifying collaborative and content-based filtering. In: Proceedings of the International Conference on Machine Learning (ICML), Banff, Canada, pp. 65–72 (2004)
21. Park, S.T., Pennock, D., Madani, O., Good, N., DeCoste, D.: Naïve filterbots for robust cold-start recommendations. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Philadelphia, PA (2006)
22. Agarwal, D., Merugu, S.: Predictive discrete latent factor models for large scale dyadic data. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), San Jose, California, USA (2007)
23. Yu, K., Yu, S., Tresp, V.: Multi-label informed latent semantic indexing. In: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Salvador, Brazil (2005)
24. Zhu, S., Yu, K., Chi, Y., Gong, Y.: Combining content and link for classification using matrix factorization. In: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Amsterdam, The Netherlands (2007)
25. Lee, H., Choi, S.: Group nonnegative matrix factorization for EEG classification. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), Clearwater Beach, Florida (2009)
26. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Las Vegas, Nevada (2008)
27. Long, B., Zhang, Z., Yu, P.S.: Co-clustering by block value decomposition. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Chicago, IL (2005)
28. Li, T., Ding, C.: The relationships among various nonnegative matrix factorization methods for clustering. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), Hong Kong (2006)
29. Chen, G., Wang, F., Zhang, C.: Collaborative filtering using orthogonal nonnegative tri-factorization. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), Omaha, NE (2007)