Learning Dual Preferences with Non-negative Matrix Tri-Factorization for Top-N Recommender System

Xiangsheng Li[1], Yanghui Rao[1](✉), Haoran Xie[2], Yufu Chen[1],
Raymond Y. K. Lau[3], Fu Lee Wang[4], and Jian Yin[5]

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
raoyangh@mail.sysu.edu.cn
[2] Department of Mathematics and Information Technology,
The Education University of Hong Kong, Tai Po, Hong Kong
[3] Department of Information Systems, City University of Hong Kong,
Kowloon Tong, Hong Kong
[4] Caritas Institute of Higher Education, Tseung Kwan O, Hong Kong
[5] Guangdong Key Laboratory of Big Data Analysis and Processing,
Guangzhou 510006, People's Republic of China

Abstract. In recommender systems, personal characteristic is possessed by not only users but also displaying products. Users have their personal rating patterns while products have different characteristics that attract users. This information can be explicitly exploited from the review text. However, most existing methods only model the review text as a topic preference of products, without considering the perspectives of users and products simultaneously. In this paper, we propose a user-product topic model to capture both user preferences and attractive characteristics of products. Different from conventional collaborative filtering in conjunction with topic models, we use non-negative matrix tri-factorization to jointly reveal the characteristic of users and products. Experiments on two real-world data sets validate the effectiveness of our method in Top-N recommendations.

Keywords: Top-N recommender system · Topic model
Matrix tri-factorization

1 Introduction

The emergence of e-commerce facilitates the development of recommender systems. In recent years, an increasing number of companies have applied recommender systems to automatically suggest products or services to their

The first two authors contributed equally to this work which was finished when Xiangsheng Li was an undergraduate student of his final year.

J. Pei et al. (Eds.): DASFAA 2018, LNCS 10827, pp. 133–149, 2018.
https://doi.org/10.1007/978-3-319-91452-7_9

customers [1]. Top-N recommendation is a personalized information filtering strategy which aims to identify a set of items that best fit interests and needs of users [2]. As one of the classical approaches for Top-N recommender systems, Collaborative Filtering (CF) via matrix factorization [3] assumed that users who exhibited similar preferences tend to have similar rating patterns for each product. Since the incomplete user-product rating data is leveraged only, traditional CF-based methods suffer from the issue of the sparsity of rating vectors [4]. Thus, the topic information of items have been extracted and then adopted in the collaborative topic regression (CTR) [5] for recommending scientific articles. CTR captures the semantic information from the item contents by latent Dirichlet allocation (LDA) [6], which can effectively identify the attractive characteristics of items. For example, the food quality and the environment may be a restaurant's topic preferences, but the price may be the attractive characteristics of electronic products. However, we argue that users not only are attracted by item characteristics (topic preferences), but also have their personal rating patterns. Taking ratings of a restaurant as an example, we assume that the restaurant's environment is important to user A, while user B focuses more on the food quality. Although both the environment and the food quality are critical aspects for restaurants, user A is more likely to give a lower rating than user B if the restaurant's environment is poor but has a good food quality. Thus, jointly exploiting both user preferences and product characteristics can obtain the dominant aspects in ratings to users and the dominant attributes (or aspects) of items, which is a key motivation of our research.

In this paper, we propose an approach named user-product topic model (UPTM). The key idea is to find the user preferences and attractive characteristics of products. Specifically, these preferences and characteristics are considered as a topic distribution, in which each topic value represents the level that a user prefers or a product is attracted. Then, these topic preferences are incorporated into matrix tri-factorization to model the ratings. The major process is as below: Firstly, the topic preferences of users and products are jointly extracted from the review text. Secondly, the topic information is incorporated into a non-negative matrix tri-factorization to facilitate rating prediction. Compared to the traditional bi-factorization, tri-factorization can better reveal the latent structures among products (samples) and attributes (features) [7]. Matrix tri-factorization is significant only when it cannot be transformed into bi-factorization, and this happens when certain constraints are applied to the tri-factorization [8]. Thus, we incorporate the topic preferences of users and products into user and product latent vectors and add a mapping matrix. The predicted ratings are demonstrated to improve the performance of Top-N recommendations compared to the conventional CF-based approach combined with a topic model. The main contributions of our paper are summarized as follows:

- We propose a probabilistic matrix tri-factorization model that incorporates both user and product preferences. By taking both types of preferences into account, the model extends the features of recommender systems.

- Our model is the first research to consolidate the non-negative matrix tri-factorization with topic information for recommendations. Revealing the latent aspects among users, products and reviews, our model effectively improves the quality of Top-N recommendations.

Experimental results on two real-world datasets indicate that our model can outperform baselines consistently.

2 Related Work

2.1 Matrix Factorization

Matrix factorization is a basic model for CF-based recommendations, in which products are recommended to a user based on other users with similar preferences of products. To learn joint latent space for users and products, a preliminary study developed a probabilistic matrix factorization (PMF) [9] that combined matrix factorization with probabilistic models. Recently, nonnegative matrix factorization (NMF) has been shown to be useful in CF recommendations [10,11]. NMF aims to factor a matrix $\boldsymbol{X}$ into two lower-dimension matrices and minimizes the square error between $\boldsymbol{X}$ and the approximation of $\boldsymbol{X}$ using those lower-dimension matrices. NMF is applied when certain non-negativity constraints exist, which makes the result easier to explain since it is natural to consider that users have non-negative affinities for some user communities based on their interests [12]. Guillamet *et al.* [13] extended the NMF to a weighted non-negative matrix factorization (WNMF) to improve the capabilities of representations. Experimental results show that WNMF achieves a great improvement in the classification accuracy compared with NMF. Ding *et al.* [14] provided an analysis of the relationship between bi-factorization and tri-factorization, and proposed an orthogonal non-negative matrix tri-factorization for clustering. This model is demonstrated to better capture the latent features of products and reveal hidden aspects underlying products [7]. Kang *et al.* [15] recently proposed a matrix completion method based on a low-rank assumption, which shows the effectiveness of matrix factorization.

2.2 Topic-Based Recommendations

To learn how users prefer products, understanding the hidden preferences for a product is quite important, such as food quality for a restaurant or price for an electronic product. Modeling these hidden factors is key to obtaining state-of-the-art performance on product recommendation system [16]. Therefore, many recommender systems rely on users feedback, which typically comes in the form of a plain-text review and a numeric score. However, in spite of the wealth of research on utilizing numeric score, the plain-text review is not well exploited.

To exploit the textual information of products, a collaborative topic regression (CTR) [5] was proposed by integrating a topic model in the matrix factorization. CTR used LDA [6] to mine the topic information from the item's text and incorporated it into PMF [9]. Compared to LDA and PMF, CTR is an appealing method in that it produces promising and interpretable results. The aforementioned method, however, did not consider the topic preference from both user and product perspectives. More specifically, CTR utilized matrix bi-factorization to capture items' attractive characteristics only. Furthermore, McAuley and Leskovec [17] proposed a model called HFT, which combines latent dimensions in rating data with topics in review text based on matrix factorization and LDA, and obtain highly interpretable textual labels for latent rating dimensions. Although HFT further mined the information under the connection between rating and review text, it also ignored the dual preference between users and items.

3 User-Product Topic Model

3.1 Problem Definition

For the reader's convenience in understanding our description of the model, we define the following terms and notations: We consider a review text as a document, which describes the evaluation of a certain product from the aspect of a certain user. Thus, an online collection consists of T documents is denoted as $\{d_1, d_2, \ldots, d_T\}$, expressed by $|U|$ users $\{u_1, u_2, \ldots, u_{|U|}\}$ over $|P|$ products $\{p_1, p_2, \ldots, p_{|P|}\}$, together with the corresponding ratings. The number of documents authored by user u is denoted as D_u while the number of documents described about product p is denoted as D_p. In particular, a document d for each user-product pair contains a sequence of N words denoted by $\{w_1, w_2, \ldots, w_N\}$ and a given rating r. A user u can make comment on several products, and a product p may be reviewed by multiple users.

The key process of our method is to find the user preferences $\theta_{u_i} \in \mathbb{R}^K$ and the attractive characteristics of products $\theta_{p_j} \in \mathbb{R}^K$, i.e., the topic preference on the aspects of users and products. Typically, user i is represented by a latent vector $u_i \in \mathbb{R}^K$ and product j by a latent vector $p_j \in \mathbb{R}^K$. The rating prediction r_{ij} that describes whether user i will like product j with the inner product between their latent representations and a mapping matrix $\boldsymbol{H}$, i.e., $u_i \boldsymbol{H} v_j^T$.

3.2 Generative Process

Our proposed approach, designated a user-product topic model (UPTM), aims to jointly learn the users' and products' latent topic vector. The motivation of designing a UPTM is to incorporate user and product topic preferences into non-negative matrix tri-factorization to factorize ratings. Typically, users will reveal their own shopping preference in the review text, and products, being reviewed by a larger number of users, can also be discovered what they attract different users in the reviews over them. Based on this phenomenon, we can assume that the topic preference of a user or a product can be exploited from the collection of reviews that this user issued or this product received. Thus, our proposed

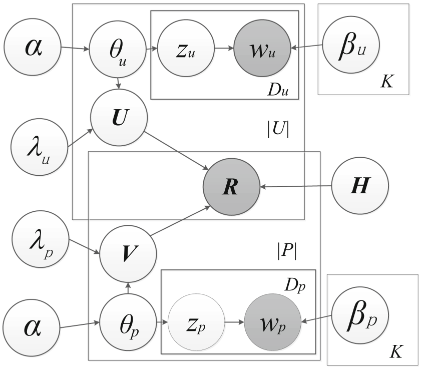

Fig. 1. Generative process of the proposed UPTM

UPTM is able to find topic preferences of users and products from review texts simultaneously. Given K topics, the generative process of UPTM, as shown in Fig. 1, is described as follows:

1. For the collection of documents expressed by user u_i:
   (a) Draw a topic distribution $\theta_{u_i} \sim$ Dirichlet (α);
   (b) For each word w_u in a document:
      i. Draw a topic assignment $z_u \sim$ Multinomial (θ_{u_i});
      ii. Draw the word $w_u \sim$ Multinomial (β_{z_u}).
2. For the collection of documents over product p_j:
   (a) Draw a topic distribution $\theta_{p_j} \sim$ Dirichlet (α);
   (b) For each word w_p in a document:
      i. Draw a topic assignment $z_p \sim$ Multinomial (θ_{p_j});
      ii. Draw the word $w_p \sim$ Multinomial (β_{p_j}).
3. For the i-th user, sample user latent vector $u_i \sim \mathcal{N}(\theta_{u_i}, \lambda_u^{-1} I_K)$.
4. For the j-th product, sample product latent vector $v_j \sim \mathcal{N}(\theta_{p_j}, \lambda_p^{-1} I_K)$.
5. For the (i, j)-th user-product pair, draw the rating

$$r_{ij} \sim \mathcal{N}(u_i \boldsymbol{H} v_j^T, c_{ij}^{-1}),$$

where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian distribution with a mean μ and a variance σ^2. Here, c_{ij} is the confidence parameter over user u_i and product p_j. If c_{ij} is larger, we trust r_{ij} more. The topic preferences on the perspective of users and products are obtained from the collection of reviews that this user issued or this product received. Typically, $\boldsymbol{H}$ is a matrix that maps dual (i.e., user and product) preferences to the rating space. The parameters λ_u and λ_p balance the proportion of users' topic preferences and ratings, and the proportion of products' topic preferences and ratings, respectively. Given that topic preferences are non-negative interests in a community [12], u_i, v_j and $\boldsymbol{H}$ are constrained to be non-negative. In this way, the objective can be considered non-negative matrix tri-factorization, which is a 3-factor decomposition of non-negative dyadic data $\boldsymbol{R} \in \mathbb{R}_+^{|U| \times |P|}$ that takes the form $\boldsymbol{R} \approx \boldsymbol{UHV}^T$, where $\boldsymbol{U} \in \mathbb{R}_+^{|U|\times K}$, $\boldsymbol{H} \in \mathbb{R}_+^{K\times K}$, and $\boldsymbol{V} \in \mathbb{R}_+^{|P|\times K}$ are constrained to be non-negative matrices.

3.3 Parameter Estimation

Since it is intractable to compute the full posterior of u_i, v_j, θ_{u_i} and θ_{p_j}, we develop an expectation maximization (EM)-style algorithm for learning parameters. Maximizing the posteriors with fixed hyper-parameters is equivalent to maximizing the complete log likelihood of $U, V, \theta_d, \boldsymbol{H}$, and $\boldsymbol{R}$ given λ_u, λ_p, $\boldsymbol{\beta_u}$, and $\boldsymbol{\beta_p}$, as follows:

$$\begin{aligned}\mathcal{L} = &-\frac{\lambda_u}{2}\sum\nolimits_i (u_i - \theta_{u_i})^T(u_i - \theta_{u_i})\\ &-\frac{\lambda_p}{2}\sum\nolimits_j (v_j - \theta_{p_j})^T(v_j - \theta_{p_j})\\ &+\sum\nolimits_i\sum\nolimits_{n_u} log\sum\nolimits_k \theta_{ik}\beta_{k,w_{in_u}}\\ &+\sum\nolimits_j\sum\nolimits_{n_p} log\sum\nolimits_k \theta_{jk}\beta_{k,w_{jn_p}}\\ &-\sum\nolimits_{i,j}\frac{c_{ij}}{2}(r_{ij} - u_i\boldsymbol{H}v_j^T)^2,\end{aligned} \tag{1}$$

where n_u and n_p denote the n-th item in the word set of the collection of documents expressed by user u, and the collection of documents over product p, respectively. A confidence parameter c_{ij} is used to determine the weight of ratings in different cases, and the Dirichlet parameter α is set to 1 by following [5]. To maximize the above likelihood function, we propose an alternating approach by coordinate ascent, i.e., by iteratively optimizing one variable while fixing the others, and repeat the procedure until convergence. The update formula of $\mathcal{L}$ with respect to $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{H}$ is as follows:

$$\boldsymbol{U}_{ik} \leftarrow \boldsymbol{U}_{ik}\sqrt{\frac{[\boldsymbol{C}\odot\boldsymbol{RVH}^T]_{ik} + \lambda_u\theta_{u_i k}}{[\boldsymbol{C}\odot(\boldsymbol{UHV}^T)\boldsymbol{VH}^T]_{ik} + \lambda_u\boldsymbol{U}_{ik}}}, \tag{2}$$

$$\boldsymbol{V}_{jk} \leftarrow \boldsymbol{V}_{jk}\sqrt{\frac{[(\boldsymbol{C}\odot\boldsymbol{R})^T\boldsymbol{UH}]_{jk} + \lambda_p\theta_{p_j k}}{[(\boldsymbol{C}\odot(\boldsymbol{UHV}^T))\boldsymbol{UH}]_{jk} + \lambda_p\boldsymbol{V}_{jk}}}, \tag{3}$$

$$\boldsymbol{H}_{ij} \leftarrow \boldsymbol{H}_{ij}\sqrt{\frac{[\boldsymbol{U}^T(\boldsymbol{C}\odot\boldsymbol{R})\boldsymbol{V}]_{ij}}{[\boldsymbol{U}^T(\boldsymbol{C}\odot(\boldsymbol{UHV}^T))\boldsymbol{V}]_{ij}}}, \tag{4}$$

where $\boldsymbol{U}_{ik}$ is the k-th item of u_i, $\boldsymbol{V}_{jk}$ is the k-th item of v_j, $\boldsymbol{H}_{ij}$ is the i-th row and the j-th column item of $\boldsymbol{H}$, $\boldsymbol{C}$ is the confidence parameter matrix with each element c_{ij}, and $\odot$ is the element-wise product.

Equations 2 and 3 show how the parameters λ_u and λ_p affect the user and product latent factors. A larger λ_u gives rise to the influence of topic preference rather than rating information. Similarly, a larger λ_p corresponds to a larger proportion of the product topic preference compared to rating information. These update formulae are in good and consistent agreement with traditional matrix factorization methods, with two additional issues that require proofs: (1) The correctness of the converged solution, and (2) the convergence of the algorithm.

Correctness Analysis. We optimize $\boldsymbol{U}$ by fixing $\boldsymbol{V}$ and $\boldsymbol{H}$ in Eq. 1, as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{U}) = -\frac{1}{2}||\boldsymbol{C} \odot (\boldsymbol{R} - (\boldsymbol{U}\boldsymbol{H}\boldsymbol{V}^T))||_F^2 - \frac{\lambda_u}{2} tr(\boldsymbol{G}_{\boldsymbol{u}}^T \boldsymbol{G}_{\boldsymbol{u}}) \\ s.t. \boldsymbol{U} \geq 0,\end{aligned}$$

where $||\cdot||_F$ is the Frobenius norm and $\boldsymbol{G}_{\boldsymbol{u}}$ is the matrix $(\boldsymbol{U} - \boldsymbol{\theta}_{\boldsymbol{u}})$. The derivative of $\mathcal{L}(\boldsymbol{U})$ with respect to $\boldsymbol{U}$ is

$$\frac{\partial \mathcal{L}(\boldsymbol{U})}{\partial \boldsymbol{U}} = \boldsymbol{C} \odot \boldsymbol{R}\boldsymbol{V}\boldsymbol{H}^T - \boldsymbol{C} \odot (\boldsymbol{U}\boldsymbol{H}\boldsymbol{V}^T)\boldsymbol{V}\boldsymbol{H}^T - \lambda_u \boldsymbol{G}_{\boldsymbol{u}}.$$

According to the Karush-Kuhn-Tucker (KKT) complementarity condition [18] of the non-negativity of $\boldsymbol{U}$, we have

$$[\boldsymbol{C} \odot \boldsymbol{R}\boldsymbol{V}\boldsymbol{H}^T - \boldsymbol{C} \odot (\boldsymbol{U}\boldsymbol{H}\boldsymbol{V}^T)\boldsymbol{V}\boldsymbol{H}^T - \lambda_u \boldsymbol{G}_{\boldsymbol{u}}]_{ik} \boldsymbol{U}_{ik} = 0.$$

This is the fixed-point relation that local minima must hold, and it is true that at convergence, from Eq. 2, the solution will satisfy

$$[\boldsymbol{C} \odot \boldsymbol{R}\boldsymbol{V}\boldsymbol{H}^T - \boldsymbol{C} \odot (\boldsymbol{U}\boldsymbol{H}\boldsymbol{V}^T)\boldsymbol{V}\boldsymbol{H}^T - \lambda_u \boldsymbol{G}_{\boldsymbol{u}}]_{ik} \boldsymbol{U}_{ik}^2 = 0.$$

This is identical to the fixed-point condition because either $\boldsymbol{U}_{ik} = 0$ or the left-hand term being equal to zero will make the above equation true. The correctness analysis of updating rules for $\boldsymbol{V}$ and $\boldsymbol{H}$ are similar to that of $\boldsymbol{U}$, by separating Eq. 1 that contains $\boldsymbol{V}$ and $\boldsymbol{H}$, respectively.

Convergence Analysis. In the following, we will demonstrate the deduction and the convergence of our updating formulas in Eqs. 2, 3, and 4. We apply the auxiliary function approach [19] and inequality Lemma [14] for the convergence analysis.

Definition 1. *$Z(h, h')$ is called an auxiliary function for $F(h)$ if the conditions*

$$Z(h, h') \geq F(h), Z(h, h) = F(h)$$

are satisfied [19].

Lemma 1. *If Z is an auxiliary function, then F is nonincreasing [19] under the update*

$$h^{(t+1)} = \arg\min_h Z(h, h^{(t)}).$$

Proof: $F(h^{(t+1)}) \leq Z(h^{(t+1)}, h^t) \leq Z(h^t, h^t) = F(h^t)$.
Note that if Z is lower bounded and we iteratively update until $F(h^{(t+1)}) = F(h^t)$, h^t becomes a local minimum of Z, which also implies the derivative $\nabla F(h^t) = 0$. The key is to find an appropriate $Z(h, h')$. □

Lemma 2. *For any matrices $\boldsymbol{A} \in \mathbb{R}_+^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}_+^{k\times k}$, $\boldsymbol{S} \in \mathbb{R}_+^{n\times k}$, $\boldsymbol{S'} \in \mathbb{R}_+^{n\times k}$, and $\boldsymbol{A}, \boldsymbol{B}$ being symmetric, the following inequality holds [14]:*

$$\sum_{i=1}^{n}\sum_{p=1}^{k} \frac{(\boldsymbol{AS'B})_{ip}\boldsymbol{S}_{ip}^2}{\boldsymbol{S}_{ip}^2} \geq tr(\boldsymbol{S}^T\boldsymbol{ASB}).$$

Proof: It can be referred to in [14]. □

Theorem 1. *Let*

$$\begin{aligned} \mathcal{J}(\boldsymbol{U}) &= -\mathcal{L}(\boldsymbol{U}) \\ &= \frac{1}{2}||\boldsymbol{C} \odot (\boldsymbol{R} - (\boldsymbol{UHV}^T))||_F^2 + \frac{\lambda_u}{2} tr(\boldsymbol{G}_{\boldsymbol{u}}^T\boldsymbol{G}_{\boldsymbol{u}}) \\ &\propto tr(\boldsymbol{G}_{\boldsymbol{u}}^T\boldsymbol{G}_{\boldsymbol{u}}) - tr(2\boldsymbol{C} \odot \boldsymbol{RVH}^T\boldsymbol{U}^T) \\ &\qquad + tr(\boldsymbol{C} \odot (\boldsymbol{UHV}^T)\boldsymbol{VH}^T\boldsymbol{U}^T). \end{aligned} \tag{5}$$

The auxiliary function of $\mathcal{J}(\boldsymbol{U})$ is then

$$\begin{aligned} &Z(\boldsymbol{U}, \boldsymbol{U'}) \\ =&\lambda_u \sum_{i,k} \boldsymbol{U}_{ik}^2 + \lambda_u \sum_{i,k} \boldsymbol{\theta_u}_{ik}^2 \\ &- 2\sum_{i,k} \boldsymbol{U}'_{ik}\boldsymbol{\theta_u}_{ik}(1 + \log\frac{\boldsymbol{U}_{ik}}{\boldsymbol{U}'_{ik}}) \\ &- 2\sum_{i,k} (\boldsymbol{C} \odot \boldsymbol{RVH}^T)_{ik}\boldsymbol{U}'_{ik}(1 + \log\frac{\boldsymbol{U}_{ik}}{\boldsymbol{U}'_{ik}}) \\ &+ \sum_{i,k} \frac{(\boldsymbol{C} \odot (\boldsymbol{U'HV}^T)\boldsymbol{VH}^T)_{ik}\boldsymbol{U}_{ik}^2}{\boldsymbol{U}'_{ik}}. \end{aligned} \tag{6}$$

Furthermore, this is a convex function with respect to $\boldsymbol{U}$ and its global minimum is

$$\boldsymbol{U}_{ik} = \boldsymbol{U}_{ik}\sqrt{\frac{[\boldsymbol{C} \odot \boldsymbol{RVH}^T]_{ik} + \lambda_u\boldsymbol{\theta_u}_{ik}}{[\boldsymbol{C} \odot (\boldsymbol{UHV}^T)\boldsymbol{VH}^T]_{ik} + \lambda_u\boldsymbol{U}_{ik}}}.$$

Proof: According to Lemma 1, it is obvious that when $\boldsymbol{U'} = \boldsymbol{U}$ the equality holds $Z(\boldsymbol{U}, \boldsymbol{U'}) = \mathcal{J}(\boldsymbol{U})$. Second, the inequality $Z(\boldsymbol{U}, \boldsymbol{U'}) \geq \mathcal{J}(\boldsymbol{U})$ also holds because the first four terms in Eq. 6 are larger than the first two terms in Eq. 5 since the inequality

$$z \geq 1 + \log(z), \forall z{>}0, \tag{7}$$

and we can set $z = \boldsymbol{U}_{ik}/\boldsymbol{U}'_{ik}$. Furthermore, the last term in Eq. 6 is larger than the last term in Eq. 5 in terms of Lemma 2. Therefore, according to Lemma 1, the minimum of $Z(\boldsymbol{U}, \boldsymbol{U}')$ fixing $\boldsymbol{U}'$ is given by

$$\begin{aligned} 0 &= \frac{\partial Z(\boldsymbol{U}, \boldsymbol{U}')}{\partial \boldsymbol{U}_{ik}} \\ &= 2\lambda_u \boldsymbol{U}_{ik} - 2\boldsymbol{\theta_u}_{ik} \frac{\boldsymbol{U}'_{ik}}{\boldsymbol{U}_{ik}} \\ &- 2(\boldsymbol{C} \odot \boldsymbol{RVH}^T)_{ik} \frac{\boldsymbol{U}'_{ik}}{\boldsymbol{U}_{ik}} \\ &+ 2\frac{(\boldsymbol{C} \odot (\boldsymbol{U'HV}^T)\boldsymbol{V})_{ik}\boldsymbol{U}_{ik}}{\boldsymbol{U}'_{ik}}. \end{aligned}$$

Then, to solve $\boldsymbol{U}_{ik}$, let $\boldsymbol{U} = \boldsymbol{U}^{(t+1)}$ and $\boldsymbol{U}' = \boldsymbol{U}^{(t)}$, we obtain the updating formula of $\boldsymbol{U}$, as shown in Eq. 2. □

Theorem 2. *Updating* $\boldsymbol{U}$ *under the update formula 2 will monotonically decrease the value in Eq. 5. The updating will finally converge.*

Proof: Since $\mathcal{J}(\boldsymbol{U})$ is lower bounded to zero, the only condition of convergence is that it is monotonically decreasing. Due to that $\mathcal{J}(\boldsymbol{U}^0) = Z(\boldsymbol{U}^0, \boldsymbol{U}^0) \geq Z(\boldsymbol{U}^1, \boldsymbol{U}^0) \geq \mathcal{J}(\boldsymbol{U}^1) \geq \cdots$, it converges. □

Theorem 3. *Let*

$$\begin{aligned} \mathcal{J}(\boldsymbol{V}) &= -\mathcal{L}(\boldsymbol{V}) \\ &= \frac{1}{2}||\boldsymbol{C} \odot (\boldsymbol{R} - (\boldsymbol{UHV}^T))||_F^2 + \frac{\lambda_p}{2} tr(\boldsymbol{G_p}^T \boldsymbol{G_p}) \\ &\propto tr(\boldsymbol{G_p}^T \boldsymbol{G_p} - tr(2\boldsymbol{C} \odot \boldsymbol{RVH}^T \boldsymbol{U}^T) \\ &\quad + tr(\boldsymbol{C} \odot (\boldsymbol{UHV}^T)\boldsymbol{VH}^T \boldsymbol{U}^T). \end{aligned} \tag{8}$$

Updating $\boldsymbol{V}$ *under the formula Eq. 3 will monotonically decrease the value* $\mathcal{J}(\boldsymbol{V})$*, and finally it converges.*

Proof: Since $\boldsymbol{V}$ is similar and symmetrical to $\boldsymbol{U}$ in Eq. 1, the proof of convergence is similar to that of $\boldsymbol{U}$. □

Theorem 4. *Let*

$$\begin{aligned} \mathcal{J}(\boldsymbol{H}) &= -\mathcal{L}(\boldsymbol{H}) \\ &= \frac{1}{2}||\boldsymbol{C} \odot (\boldsymbol{R} - (\boldsymbol{UHV}^T))||_F^2 \\ &\propto tr(-2\boldsymbol{U}^T(\boldsymbol{C} \odot \boldsymbol{R})\boldsymbol{VH}^T) \\ &\quad + tr(\boldsymbol{U}^T(\boldsymbol{C} \odot (\boldsymbol{UHV}^T))\boldsymbol{VH}^T). \end{aligned} \tag{9}$$

The auxiliary function of $\mathcal{J}(\boldsymbol{H})$ is then

$$\begin{aligned} &Z(\boldsymbol{H},\boldsymbol{H}') \\ =& -2\sum_{i,j}(\boldsymbol{U}^T(\boldsymbol{C}\odot\boldsymbol{R})\boldsymbol{V})_{ij}\boldsymbol{H}'_{ij}(1+\log\frac{\boldsymbol{H}_{ij}}{\boldsymbol{H}'_{ij}}) \\ &+\sum_{i,j}\frac{(\boldsymbol{U}^T(\boldsymbol{C}\odot(\boldsymbol{U}\boldsymbol{H}'\boldsymbol{V}^T))\boldsymbol{V})_{ij}\boldsymbol{H}^2_{ij}}{\boldsymbol{H}'_{ij}}. \end{aligned} \tag{10}$$

Furthermore, this is a convex function with respect to $\boldsymbol{H}$ and its global minimum is

$$\boldsymbol{H}_{ij}=\boldsymbol{H}_{ij}\sqrt{\frac{[\boldsymbol{U}^T(\boldsymbol{C}\odot\boldsymbol{R})\boldsymbol{V}]_{ij}}{[\boldsymbol{U}^T(\boldsymbol{C}\odot(\boldsymbol{U}\boldsymbol{H}\boldsymbol{V}^T))\boldsymbol{V}]_{ij}}}.$$

Proof: According to Lemma 1, it is obvious that when $\boldsymbol{H}'=\boldsymbol{H}$ the equality holds $Z(\boldsymbol{H},\boldsymbol{H}')=\mathcal{J}(\boldsymbol{H})$. Second, the inequality $Z(\boldsymbol{H},\boldsymbol{H}')\geq\mathcal{J}(\boldsymbol{H})$ also holds because the first term in Eq. 10 is larger than the first term in Eq. 9 since the inequality (Eq. 7) and we can set $z=\boldsymbol{H}_{ij}/\boldsymbol{H}'_{ij}$. Furthermore, the second term in Eq. 10 is larger than the second term in Eq. 9 in terms of Lemma 2.

Therefore, according to Lemma 1, the minimum of $Z(\boldsymbol{H},\boldsymbol{H}')$ fixing $\boldsymbol{H}'$ is given by

$$\begin{aligned} 0 &= \frac{\partial Z(\boldsymbol{H},\boldsymbol{H}')}{\partial\boldsymbol{H}_{ij}} \\ &= -2(\boldsymbol{U}^T(\boldsymbol{C}\odot\boldsymbol{R})\boldsymbol{V})_{ij}\frac{\boldsymbol{H}'_{ij}}{\boldsymbol{H}_{ij}} \\ &+2\frac{(\boldsymbol{U}^T(\boldsymbol{C}\odot(\boldsymbol{U}\boldsymbol{H}'\boldsymbol{V}^T))\boldsymbol{V})_{ij}\boldsymbol{H}_{ij}}{\boldsymbol{H}'_{ij}}. \end{aligned}$$

Then, to solve $\boldsymbol{H}_{ij}$, let $\boldsymbol{H}=\boldsymbol{H}^{(t+1)}$ and $\boldsymbol{H}'=\boldsymbol{H}^{(t)}$, and we obtain the updating formula of $\boldsymbol{H}$, as shown in Eq. 4. □

Theorem 5. *Updating $\boldsymbol{H}$ under the update formula 4 will monotonically decrease the value in Eq. 9. The updating will finally converge.*

Proof: Since $\mathcal{J}(\boldsymbol{H})$ is lower bounded to zero, the only condition of convergence is that it is monotonically decreasing. Due to that $\mathcal{J}(\boldsymbol{H}^0)=Z(\boldsymbol{H}^0,\boldsymbol{H}^0)\geq Z(\boldsymbol{H}^1,\boldsymbol{H}^0)\geq\mathcal{J}(\boldsymbol{H}^1)\geq\cdots$, it converges. □

Optimization of Other Parameters. Learning θ_{u_i} and θ_{p_j} is different because they are difficult to derive. We can, however, apply Jensen's inequality to solve this problem. For θ_{u_i}, it is constrained by a low bound with respect to θ_{u_i}. We now define $q(z_{in}=k)=\phi_{ink}$ and separate the items that contain θ_{u_i}.

We then apply Jensen's inequality as follows:

$$\begin{aligned}\mathcal{L}(\theta_{u_i}) &\geq -\frac{\lambda_u}{2}\sum\nolimits_i (u_i-\theta_{u_i})^T(u_i-\theta_{u_i}) \\ &+\sum\nolimits_{n_u}\sum\nolimits_k \phi_{ink}(log\theta_{ik}\beta_{k,w_{in_u}} - log\phi_{in_u k}) \\ &= \mathcal{L}(\theta_{u_i},\phi_i),\end{aligned}$$

where $\phi_i = (\phi_{ink})_{n=1,k=1}^{|D_u|\times K}$, $|D_u|$ is the number of words in the document group of this user. The optimal ϕ_{ink} satisfies $\phi_{ink} \propto \theta_{ik}\beta_{k,w_{in_u}}$.

Thus, $\mathcal{L}(\theta_{u_i},\phi_i)$ gives the tight lower bound of $\mathcal{L}(\theta_{u_i})$. The gradient projection [20] can be applied to optimize θ_{u_i}. We can then optimize $\boldsymbol{\beta_u}$ as follows:

$$\beta_{kw_i} \propto \sum_d \sum_{n_u} \phi_{in_u k}1[w_{in_u} = w].$$

This is consistent with the M-step of the EM-algorithm in LDA [6]. Moreover, for θ_{p_j}, $\phi_{jn_p k}$, and $\beta_k w_j$, it is similarly updated. Note, however, that in order to ensure topics θ_{u_i} and θ_{p_j} have the same semantic information, we make $\boldsymbol{\beta_u}$ equal to $\boldsymbol{\beta_p}$. After estimating u_i, v_j, θ_{u_i}, and θ_{p_j}, the rating is predicted by $r_{ij} \approx u_i \boldsymbol{H} v_j^T$.

Complexity Analysis. Our method applies an EM-style algorithm, so the parameter estimation algorithm is implemented in an iterative manner. The efficiency is determined by the convergence and time cost per iteration. The time cost mainly comes from two parts: topic modeling and matrix tri-factorization. For topic modeling, the time complexity is $\mathcal{O}(N_{iter}\cdot(|U|\cdot|\widetilde{D_u}|+|P|\cdot|\widetilde{D_p}|)\cdot K\cdot\tilde{l})$, where N_{iter} is the number of iterations, $\widetilde{D_u}$ and $\widetilde{D_p}$ are the average number of review text of users and products, respectively. $|U|$ and $|P|$ are the number of users and products, K is the number of topic and $\tilde{l}$ is the average length of each review text. For matrix tri-factorization, the time complexity is $\mathcal{O}(N_{iter}\cdot|U|^2\cdot|P|^2\cdot|K|^5)$. Thus, the total time complexity is

$$\varTheta = N_{iter}\cdot((|U|\cdot|\widetilde{D_u}|+|P|\cdot|\widetilde{D_p}|)\cdot K\cdot\tilde{l} + |U|^2\cdot|P|^2\cdot|K|^5).$$

4 Experiments

4.1 Datasets

We employ two datasets in our experiment: IMDB [21] and Yelp2013[1]. IMDB and Yelp2013 contain users' reviews and ratings on different aspects of movies, and on different restaurants, respectively. Table 1 presents the statistical information of these datasets.

In our experiment, we split each dataset into three parts: a training set (80%), a validation set (10%), and a testing set (10%). Each model is trained on the training set and obtains its optimal parameters on the validation set. The performance is then evaluated on the testing set.

[1] http://www.yelp.com/dataset_challenge.

Table 1. Statistical information of datasets.

Dataset	# users	# products	# reviews	Rating scale
IMDB	1,310	1,635	84,919	1–10
Yelp2013	1,631	1,633	78,966	1–5

4.2 Comparisons and Evaluation

The baseline models are listed as follows:

- **2NMF** [12]: Non-negative matrix bi-factorization is a type of probability matrix factorization in which a rating matrix $\boldsymbol{R}$ is factorized into two matrices $\boldsymbol{U}$ and $\boldsymbol{V}$.
- **3NMF**: Different from 2NMF, 3NMF is a non-negative matrix tri-factorization that factorizes a rating matrix into three matrices, i.e., $\boldsymbol{U}, \boldsymbol{V}$ and $\boldsymbol{H}$. Review text is not considered in this method.
- **CTR** [5]: Collaborative topic regression is a model used to perform topic modeling and collaborative filtering simultaneously.
- **RSMC** [15]: Recommender system via matrix completion is a method based on a low-rank assumption for Top-N recommendations.

We measure the performance of the proposed model and baselines by comparing *Precision* and *Recall*, as in [5]. For each user, *Precision* and *Recall* are defined as follows:

$$Precision@M = \frac{\#\ products\ the\ user\ likes\ in\ Top\ M}{M},$$

$$Recall@M = \frac{\#\ products\ the\ user\ likes\ in\ Top\ M}{Total\ number\ of\ products\ the\ user\ likes},$$

where M is the number of returned items sorted by their predicted ratings. Specifically, *Precision* evaluates the recommendation accuracy of the model while *Recall* evaluates which of the returned items were actually in each user's purchase records. The final result reported is the average precision and recall over all users.

4.3 Experimental Setting

As we mention earlier, we leveraged a validation dataset to find the optimal parameters of all models. For 2NMF and 3NMF, we employed multiplication update rules to avoid the learning rate setting, which is similar to [12]. CTR delivered good performance when $\lambda_u = 0.01$ and $\lambda_p = 0.01$, and when $a = 1$ and $b = 0.01$, where a and b are the confidence parameters c_{ij}. For RSMC, we set $\mu = 1.2 \times 10^{-3}$ and $\gamma = 1.3$ in IMDB and $\mu = 1.5 \times 10^{-3}$ and $\gamma = 1.8$ in Yelp2013. Except for RSMC, we set a common topic dimension $K = 20$, i.e., the rating matrix $\boldsymbol{R}$ is factorized into $\boldsymbol{U}^{|U| \times K}$ and $\boldsymbol{V}^{|P| \times K}$ in 2NMF and CTR, while

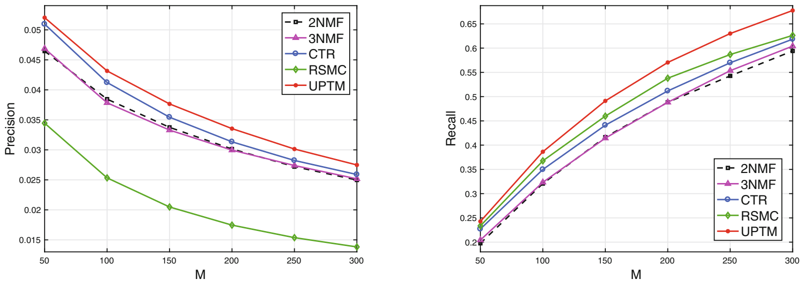

Fig. 2. Performance on IMDB with different M.

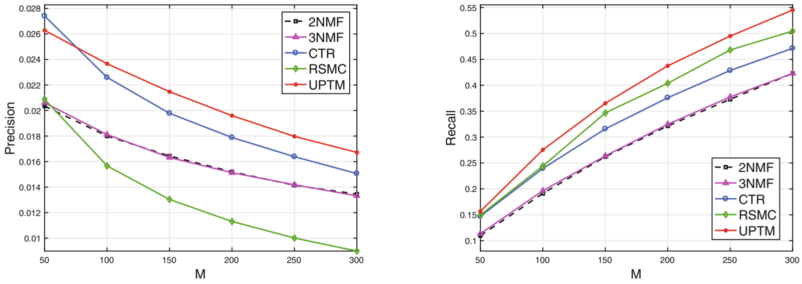

Fig. 3. Performance on Yelp2013 with different M.

it is factorized into $\boldsymbol{U}^{|U|\times K}, \boldsymbol{S}^{K\times K}$ and $\boldsymbol{V}^{|P|\times K}$ in 3NMF and our UPTM. The impact of topic number will be further discussed in Sect. 4.5. For our UPTM, we directly set $a = 1, b = 0.01$, and apply a grid search to obtain the best parameters λ_u and λ_p on the validation dataset. On IMDB, the optimal performance is achieved when $\lambda_u = 1000$ and $\lambda_p = 10$, while $\lambda_u = 100$ and $\lambda_p = 1$ obtains the best performance on Yelp2013. For evaluation, we set $M = 50, 100, 150, 200, 250,$ and 300 and fix the parameters of each approach to the best results.

4.4 Performance Comparison

The overall performance of each approach on IMDB and Yelp2013 are shown in Figs. 2 and 3, respectively. For non-negative matrix factorization, 3NMF sightly outperforms 2NMF for both metrics, which shows the effectiveness of non-negative matrix tri-factorization in recommender systems. However, they both do not perform as well as other models in *Recall*. RSMC performs better than most baseline models in *Recall*, while it is the worst in *Precision* in both datasets. Our proposed model, the UPTM, outperforms 2NMF, 3NMF, RSMC and CTR on IMDB in terms of different M consistently. On Yelp2013, CTR sightly outperforms the UPTM on *Precision* when $M = 50$. However, a zero entry in the rating matrix may be due to the fact that the user is not interested in the product, which indicates that *Recall* is a more important performance measure than *Precision* on Top-N recommender systems [1,5]. This slightly worse *Precision* is unconvincing on the condition that *Recall* of the proposed UPTM improves 6.51% and 15.04% compared to CTR. On average, UPTM improves 2NMF, 3NMF, CTR and RSMC by 18.03%, 16.60%, 10.06%, and 82.06%, respectively, in terms of precision, and by 18.03%, 16.60%, 10.06%, and 6.24% in terms of *Recall*, respectively, on the dataset of IMDB. On the other dataset Yelp2013, the UPTM improves 2NMF, 3NMF, CTR, and RSMC by 28.60%, 28.58%, 6.58% and 63.41%, respectively, in terms of *Precision*, and by 37.35%, 35.27%, 14.13% and 7.51%, respectively, in terms of *Recall*.

The performance comparison shows the effectiveness of our UPTM which captures both users' and products' topic preferences. Compared to conventional non-negative matrix factorization, our model incorporates the topic information between user and products, which effectively improves the recommendation performance. Compared to CTR, our model leverages topic information on both aspect of users and products and adopts matrix tri-factorization to better reveal the latent aspects among users, products and topic features [7], which significantly improves the recommendation performance. Compared to the state-of-the-art matrix completion method, RSMC, our model also performs better, achieving the best performance in Top-N recommendations among matrix factorization methods. Note that execution times of algorithms and the performance on a small M value are also important to test the effectiveness of a recommender system, we leave these kinds of evaluations to the future work due to the limit of space.

4.5 Influence of the Number of Topics

The number of topics K is an important parameter in topic-based recommendation. We tried different values of K and the result is shown in Fig. 4. We can observe that as the cross-validation we did, $K = 20$ delivers the best performance in both data sets. Furthermore, as K gets larger, the performance gets worse. This is because too many topics over-depict the review features. Topic number below 30 is enough to depict the review features and performs well in recommendation.

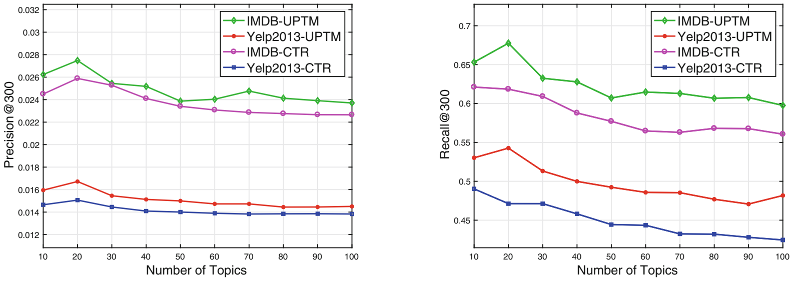

Fig. 4. *Precision* and *Recall* performance with different topic number K

We also investigate the effect of K on our model and the baseline CTR. Our model performs better over both of the evaluations. This demonstrates that considering two aspects of users and products simultaneously is superior to the one-side topic modeling. As the number of topics becomes larger, their performances get worse since the number of the documents under a specific topic come to the bottleneck. On the other hand, like LDA, if K is too small, the topics will be coarse and a lot of useful features are missing. If K is too large, some useless features may be incorporated and useful features may be confounded by those noise.

4.6 Impact of Training Data Size

A good recommender system aims to perform well even when the data is quite sparse. We examined the impact of the size of training data on each model's performance by randomly selecting $x\%$ data from the original training corpus. The values of x varied from 20 to 80, with an interval of 20. In case of coincidence, we extracted the training data 10 times and calculated the average of performance each time.

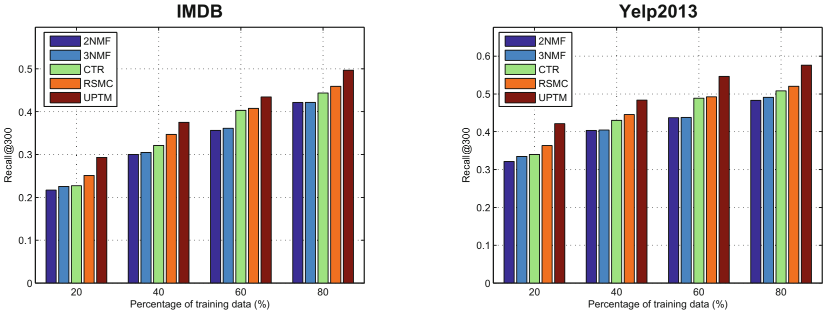

Fig. 5. The *Recall*@300 with different training data size.

To be consistent with existing evaluations [1], the *Recall*@300 performance is shown in Fig. 5. We observed that the performance of all methods increased as the size of the training data increased, and our model outperforms the baselines on both datasets.

4.7 Parameter Effect Analysis

Here, we study the effects of the parameters λ_u and λ_p on the proposed UPTM using *Recall*@300 (ref. Table 2). On IMDB, UPTM achieved the best performance when $\lambda_u = 1000$ and $\lambda_p = 10$, which indicates that product topic preference contributes more than user topic preference on this dataset. On Yelp2013, the optimal parameters for the best performance were $\lambda_u = 100$ and $\lambda_p = 1$; the contribution of product preference is consistently important in this dataset.

Table 2. The *Recall*@300 of UPTM with different λ_u and λ_p.

(a) IMDB

λ_p \ λ_u	0.01	0.1	1	10	100	1000
0.01	0.647	0.657	0.648	0.661	0.663	0.649
0.1	0.650	0.656	0.664	0.666	0.664	0.648
1	0.655	0.653	0.663	0.669	0.663	0.663
10	0.666	0.659	0.648	0.659	0.663	**0.678**
100	0.656	0.655	0.648	0.657	0.661	0.664
1000	0.657	0.656	0.659	0.666	0.647	0.668

(b) Yelp2013

λ_p \ λ_u	0.01	0.1	1	10	100	1000
0.01	0.527	0.526	0.534	0.533	0.532	0.528
0.1	0.531	0.530	0.531	0.537	0.535	0.532
1	0.527	0.538	0.532	0.537	**0.545**	0.531
10	0.531	0.529	0.532	0.529	0.541	0.531
100	0.528	0.530	0.532	0.533	0.537	0.535
1000	0.527	0.530	0.532	0.532	0.532	0.529

In addition, on both datasets, when λ_u and λ_p were both small (i.e., smaller than 1), the performance suffered, which means that both user and product topic preferences affect recommendation performance.

5 Conclusions

In this paper, we proposed a probabilistic matrix tri-factorization approach named UPTM, which applied LDA to mine the user and product topic preferences and incorporated them into the matrix factorization. We also leveraged non-negative matrix tri-factorization to factorize the rating matrix into a user latent matrix, a product latent matrix and a mapping matrix. The main conclusions of our paper are the following:

- By mining the topic preference not only from the product aspect but also from the user aspect, our UPTM was used to find a connection between a user's topic of interest and a product topic that was attractive to the user.
- The matrix factorization part of our model is based on non-negative matrix tri-factorization. By incorporating a third mapping matrix, the predicted rating was demonstrated to enhance the recommender performance.

In the future, we plan to explore the implementation of parallel calculating algorithms, which can make the proposed method scalable to large-scale datasets.

Acknowledgements. We are grateful to the anonymous reviewers for their valuable comments on this manuscript. The research has been supported by the National Natural Science Foundation of China (61502545, U1611264, U1711262), a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (UGC/FDS11/E03/16), and the Individual Research Scheme of the Dean's Research Fund 2017–2018 (FLASS/DRF/IRS-8) of The Education University of Hong Kong.

References

1. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1235–1244 (2015)
2. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)

3. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
4. Chen, C., Zheng, X., Wang, Y., Hong, F., Lin, Z.: Context-ware collaborative topic regression with social matrix factorization for recommender systems. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 9–15 (2014)
5. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 448–456 (2011)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
7. Yang, X., Huang, K., Zhang, R., Hussain, A.: Learning latent features with infinite non-negative binary matrix tri-factorization. In: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (eds.) ICONIP 2016. LNCS, vol. 9947, pp. 587–596. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46687-3_65
8. Li, T., Ding, C.: The relationships among various nonnegative matrix factorization methods for clustering. In: Proceedings of the 6th International Conference on Data Mining, pp. 362–371 (2006)
9. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Proceedings of Advances in Neural Information Processing Systems, pp. 1257–1264 (2007)
10. Luo, X., Zhou, M., Xia, Y., Zhu, Q.: An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Trans. Ind. Inform. **10**, 1273–1284 (2014)
11. Hernando, A., Bobadilla, J., Ortega, F.: A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. Knowl.-Based Syst. **97**, 188–202 (2016)
12. Zhang, S., Wang, W., Ford, J., Makedon, F.: Learning from incomplete ratings using non-negative matrix factorization. In: Proceedings of the 2006 SIAM International Conference on Data Mining, pp. 548–552 (2006)
13. Guillamet, D., Vitrià, J., Schiele, B.: Introducing a weighted non-negative matrix factorization for image classification. Pattern Recogn. Lett. **24**, 2447–2454 (2003)
14. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 126–135 (2006)
15. Kang, Z., Peng, C., Cheng, Q.: Top-N recommender system via matrix completion. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, pp. 179–185 (2016)
16. Bennett, J., Elkan, C., Liu, B., Smyth, P., Tikk, D.: KDD Cup and workshop 2007. SIGKDD Explor. **9**, 51–52 (2007)
17. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM Conference on Recommender Systems, pp. 165–172 (2013)
18. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
19. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Proceedings of Advances in Neural Information Processing Systems, pp. 556–562 (2001)
20. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont (1999)
21. Diao, Q., Qiu, M., Wu, C., Smola, A.J., Jiang, J., Wang, C.: Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 193–202 (2014)