Research Article

Learning to Make Document Context-Aware Recommendation with Joint Convolutional Matrix Factorization

Lei Guo,1,2 Yu Han,1 Haoran Jiang,3 Xinxin Yang,4 Xinhua Wang,4 and Xiyu Liu1,2

1School of Business, Shandong Normal University, Jinan, China
2Postdoctoral Research Station of Management Science and Engineering, Shandong Normal University, Jinan, China
3Information Technology Bureau, Shandong Post Company, Jinan, China
4School of Information Science and Engineering, Shandong Normal University, Jinan, China

Correspondence should be addressed to Lei Guo; leiguo.cs@gmail.com and Xinhua Wang; wangxinhua@sdnu.edu.cn

Received 1 August 2019; Accepted 23 December 2019; Published 27 January 2020

1. Introduction

With the explosive growth of the number of users and items in e-commerce services, the user-to-item rating data often suffer from the data sparsity issue; that is, users can only interact with a small number of items, and items can only be visible to limited users. To overcome this challenge, several CR-based methods [1–8] have been developed to consider not only the rating matrix but also the context information (such as demography of users, social networks, and item description documents). For example, Wang and Blei [9] adopted topic modeling (LDA) techniques to discover latent aspects from reviews, and based on this, they proposed collaborative topic regression (CTR) to improve traditional collaborative filtering (CF). Similarly, in [5, 10], LDA was also used to discover potential aspects from the reviews of...
items or users. Bao et al. [2] employed a topic modeling technique to model the latent topics in review texts and utilized matrix factorization (MF) for rating prediction. Chen et al. [6] suggested adopting a novel context-aware hierarchical Bayesian method to take rating context and social relationships into consideration for prediction. These methods have achieved significant improvement compared with the methods that only utilize user-item ratings.

However, as these existing document-based modeling methods mainly adopt the bag-of-words model, they cannot fully capture the contextual information of documents, which will lead to suboptimal results. To address this issue, Kim et al. [11] resorted to the CNN model, which has achieved great success in computational graphs and natural language processing, to make a deeper understanding of the documents. Specifically, they integrated CNN into probability matrix factorization (PMF) and proposed a document context-aware recommendation method ConvMF, which not only can capture the contextual information of item description documents but also can improve the rating prediction accuracy. However, ConvMF only models the document context from an item view and ignores the following observations.

Firstly, in real-world scenario, items are not i.i.d, and the item relationships are important factors to impact the user’s decision. For example, when buying products on an e-commerce website, a user may tend to buy the products related to what he/she has bought in the recent past. To incorporate item relationships into the recommendation method, many previous studies have focused on learning similarities between items through the historical user-item interactions [12–16]. For example, Zhang et al. [17, 18] explored the temporal relationship between items from the user’s historical check-in data when recommending locations, and the experimental results demonstrated that incorporating item relationship into the recommendation algorithm can improve the recommendation accuracy. But these existing methods are not developed for content context-aware recommendations and cannot model the item relationship and document contextual information simultaneously.

Secondly, as we are social animals, we often turn to our friends for recommendations. The social relationship among users is another key factor that can help users make the right choice. For example, when we want to go to a restaurant or go to a cinema or hesitating between two T-shirts, we often ask friends for advice. Based on the social homogeneous theory [19] that friends often have similar preferences, Ma et al. [20] proposed a PMF-based Social Recommendation (SoRec) method, which not only compensates the data sparseness through social relationships but also improves the recommendation performance significantly. Besides local friends, users also tend to seek advice from users with high global reputations. Motivated by this, Tang et al. [21] developed a novel framework LOCABAL by taking advantage of both local and global social context for recommendation. As a user may trust different subsets of friends regarding different domains, Yang et al. [22] tried to develop a circle-based recommender system, which focuses on inferring category-specific social trust circles from available rating data combined with social network data. However, the above social recommendation methods mainly focus on modeling user’s social influence in the traditional user-to-item rating matrix, and user’s content information is ignored.

Thirdly, as users often express their interests by directly writing down their opinions in the posted reviews, user’s reviews provide us a meaningful way to infer their preferences to items. Recently, several recommendation methods have been proposed to leverage content information to enhance the rating prediction task. For example, Tan et al. [5] exploited textual review information, as well as ratings, to model user preferences and item features in a shared topic space and subsequently introduced them into a MF model for recommendation. To take into account the document contextual information, Kim et al. [11] introduced CNN into the PMF framework to model the document content and rating matrix simultaneously. Shen et al. [23] proposed an automatic learning resources recommendation algorithm based on CNN, where CNN is used to predict the latent factors from the text information. In [24], Zhang et al. developed a new hybrid model to jointly model content information implicit user feedback to make accurate recommendations. However, these existing methods pay more attention to the content of items rather than users. Modeling the content information of both users and items in a unified framework is still largely unexplored.

1.1. Motivating Example. In context-aware recommendation scenario, there are four kinds of information that can be utilized, that is, the reviews of both users and items and the relationships of both users and items. Suppose a user wants to buy a T-shirt on the e-commerce website, as shown in Figure 1, he/she will first ask his/her friends for recommendations, and the friends with highest expertise levels in recommending clothes (they usually have many meaningful comments to the related products) will impact him/her the most. Then, the user will further filter out the items with low ratings and negative reviews. Moreover, if a T-shirt is related to the pants (e.g., they have the same band or style with the T-shirt) that he/she has bought recently, it will be recommended with high probability.

Motivated by the above observations, in this work, we focus on the CR problem and propose a JCMF method to consider the reviews of both users and items and the relationships of both users and items simultaneously. More specifically, to jointly model the document contextual information and the user-item rating matrix, we introduce ConvMF as our basic recommendation framework, which integrates CNN into PMF to enhance the rating prediction accuracy. To exploit the relationships among items, we propose an item relation-aware method CMF-I by sharing the item latent factor between item relation network and user-item matrix. To model the user’s social influence, similar to CMF-I, we incorporate the user’s social relationships into CMF-I by sharing the user latent factor between the social network and user-item matrix. To consider user’s reviews, we exploit another CNN to model...
the document contextual information of users and finally reach the JCMF model.

The main contributions of this study are summarized as follows:

(i) We systematically address CR in online systems and propose a joint convolutional matrix factorization (JCMF) to consider the reviews of both users and items and the relationships of both users and items simultaneously.

(ii) We propose an item relation-aware recommendation method CMF-I to exploit the relationships among items.

(iii) We incorporate the user’s social relationships into CMF-I by bridging the social network and user-item matrix space with a shared user latent factor.

(iv) We incorporate another CNN network to model the document contextual information of users.

(v) We conduct extensive experiments on the real-world dataset Yelp to demonstrate the effectiveness of our proposed method.

The remainder of this work is organized as follows. Section 2 describes the research problem and our proposed solution. Sections 3 and 4 present the experimental settings and results, respectively. Section 5 conducts model analysis to validate the effectiveness of different model components. Section 6 provides related works, and Section 7 concludes this work.

2. Recommendation Framework

In this section, we first formulate the recommendation problem and then describe our joint convolutional matrix factorization recommendation framework.

2.1. Task definition. In this work, we use bold capital letters (e.g., X) to represent matrices and graphs and use squiggle capital letters (e.g., \(\mathcal{X}\)) to denote sets. We employ bold capital letters with subscripts (e.g., \(X_j\)) to denote vectors and use normal lowercase or capital letters (e.g., x or X) to denote scalars. All vectors are in column forms if not clarified. The notations used in this work are summarized in Table 1.

In CR scenario, there are four kinds of information that can be utilized, that is, user-item rating matrix, users’ social network, the relationship among items, and the document reviews of both users and items. Let \(\mathcal{U} = \{u_1, u_2, \ldots, u_N\}\) be the set of users and \(\mathcal{V} = \{v_1, v_2, \ldots, v_M\}\) be the set of items, where \(N\) and \(M\) are the numbers of users and items, respectively. We use \(R = (r_{ij})_{N \times M}\) to denote the user-item rating matrix with each entry \(r_{ij}\) indicating the rating of user \(u_i\) on item \(v_j\). We define \(C = (c_{ij})_{M \times M}\) to represent the item-item relationship matrix with each entry \(c_{ij}\) denoting that there exists a direct relationship between item \(v_j\) and \(v_j\), and \(c_{ij} = 0\) otherwise. Similarly, we use \(S = (s_{ij})_{N \times N}\) to denote the adjacency matrix of user’s social network, where \(s_{ij} = 1\) means the user \(u_i\) has direct social connection with \(u_j\), and \(s_{ij} = 0\) otherwise. Suppose \(\mathcal{Y}\) and \(\mathcal{X}\) be the set of review texts from users and items, respectively. Then, the task of CR is defined as how to make accurate rating predictions in \(\mathcal{R}\) for \(\mathcal{U}\) and \(\mathcal{V}\) by exploring the relation matrices \(C, S, \mathcal{X}\) and \(\mathcal{Y}\).

Input. User set \(\mathcal{U}\), item set \(\mathcal{V}\), user-item rating matrix \(R\), the adjacency matrix of user’s social network \(S\), the item’s relation matrix \(C\), the user’s review text \(\mathcal{X}\), and the item’s review text \(\mathcal{Y}\).

Output. A personalized score function that maps an item to a rating score \(f: \mathcal{V} \rightarrow \mathbb{R}\).

2.2. The ConvMF Method. To capture the item’s document context, we introduce ConvMF [11] as our basic recommendation framework, which integrates convolutional neural network (CNN) into probability matrix factorization (PMF) to capture the contextual information of documents and further enhance the rating prediction accuracy. The graphical model of ConvMF is shown in Figure 2.

Suppose the observed rating matrix is denoted by \(R = (r_{ij})_{N \times M}\), where \(N\) and \(M\) are numbers of users and items, respectively. Let \(U \in \mathbb{R}^{N \times L}\) and \(V \in \mathbb{R}^{M \times L}\) be the latent user and item factor matrices, with the column vectors \(U_i\) and \(V_j\) denoting user-specific and item-specific latent feature factors, respectively. Then, the conditional distribution over the observed ratings can be defined as

\[
p(R | U, V, \sigma^2) = \prod_{i} \prod_{j} \mathcal{N}(r_{ij} | U_i^T V_j, \sigma^2)^{I_{ij}},
\]

where \(\mathcal{N}(x | \mu, \sigma^2)\) is the probability density function of the Gaussian normal distribution with mean \(\mu\) and variance \(\sigma^2\). \(I_{ij}\) is the indicator function that is equal to 1 if user \(i\) rated item \(j\) and 0 otherwise. We place zero-mean spherical Gaussian prior on the user feature vector:

\[
p(U | \sigma^2_U) = \prod_{i} \mathcal{N}(U_i | 0, \sigma^2_U I_1).
\]

For the item latent factor \(V\), ConvMF assumes it is generated from three variables: (1) internal weights \(W^I\) in CNN, (2) the document \(X_j\) of item \(j\), and (3) the Gaussian
noise (denoted by \( \epsilon \)) that is used to optimize the item’s latent factor for the rating. Then, we can obtain the item latent factor by the following equations:

\[
V_j = \text{cnn}(W^1, X_j) + \epsilon_j,
\]

\[
\epsilon_j \sim \mathcal{N}(0, \sigma^2_{\epsilon}I).
\]

(3)

Similar to user latent factor \( U \), we also place zero-mean spherical Gaussian prior on each weight \( w_k^1 \) in \( W^1 \):

\[
p(W^1 | \sigma^2_{W^1}) = \prod_k \mathcal{N}(w_k^1 | 0, \sigma^2_{W^1}).
\]

(4)

Thus, we can arrive at the conditional distribution over item latent factor \( V \):

\[
p(V | W^1, X, \sigma^2_V) = \prod_j \mathcal{N}(V_j | \text{cnn}(W^1, X_j), \sigma^2_VI).
\]

(5)

where \( \text{cnn}(W^1, X_j) \) is the item latent factor learned from the CNN model that is used as the mean of the Gaussian distribution (denoted in equation (4)). It plays a role to bridge CNN and PMF and can capture the document content features and the latent factors in the rating matrix simultaneously.

However, ConvMF assumes items are i.i.d and only models the document context from an item view. The relationships of both items and users and the document context of users are not considered, which might be beneficial for the CR task.

2.3. Recommendation with Item Relationships. ConvMF ignores the relationship between items, which is an important factor in many recommendation scenarios. For example, in product recommendation scenarios, when a user bought a fishing rod in the past, he/she will probably buy the fishing bait later. Items that are bought within a short time interval have a strong correlation with each other. With this intuition in mind, we construct the item relation network by the following data policy; that is, if two items are rated by the same user within a short time window (in experiments, we set \( \Delta T \) as 24 hours) (denoted by \( \Delta T \)), we assume these two items are correlated. For the items that are rated beyond the time window or not rated by the same user, we cannot infer any correlations. The definition of the item correlation network is given as follows.

Definition 1 (Item Correlation Network). Let \( \mathcal{G}_V = (\mathcal{X}_V, \mathcal{Y}_V) \) be the item correlation network, where \( \mathcal{G}_V \) is the edge set. Given a time interval \( \Delta T \), for the item pair \( (v_i, v_j), (v_j, v_j) \) rated by the same user, where \( v_i \in \mathcal{X}_V, v_j \in \mathcal{Y}_V \), and \( t_i \) and \( t_j \) are time of being rated. If \( 0 < t_j - t_i \leq \Delta T \), there will be an edge from \( v_i \) to \( v_j \) \((v_i, v_j) \in \mathcal{E}_V\) and none otherwise.

Let \( C = (c_{ij})_{M \times M} \) denote the adjacency matrix of item correlation network \( \mathcal{G}_V \), with each entry \( c_{ij} \) denoting whether item \( v_i \) correlates with \( v_j \), that is \( c_{ij} = 1 \) if \( v_i \) directly connects to \( v_j \) and 0 otherwise. We factorize \( C \) to learn a low-dimensional representation of items’ correlation relationship. We use \( K \in \mathbb{R}^{K \times M} \) and \( V \in \mathbb{R}^{K \times M} \) to represent the latent item relation and item factor matrices, with column vectors \( K \) and \( V \), denoting relation-specific and item-specific latent vectors, respectively. Then, conditional distribution over the observed item correlation network can be defined as

\[
p(C | K, V, \sigma^2_C) = \prod_{i}^{\mathcal{M}} \prod_{j}^{\mathcal{M}} \mathcal{N}(c_{ij} | K^T V_j, \sigma^2_C I)_{ij}.
\]

(6)

where \( I_{ij} \) is the indicator function. For the relation latent factor \( K \), we place zero-mean spherical Gaussian prior on it:

\[
p(K | \sigma^2_K) = \prod_{i}^{\mathcal{M}} \mathcal{N}(K_i | 0, \sigma^2_K I).
\]

(7)

As we have shown in Figure 3, to integrate the item relation network into ConvMF, we share the item latent factor \( V \) between item relationships and rating matrix. Then, the item latent factor will not only be affected by the ratings and contents but also their correlations. After combining item relationship matrix, the joint probability distribution of our convolutional matrix factorization with item relations (CMF-I) method can be written as
2.4. Recommendation with Social Influence. CMF-I only considers the influence of relationships from an item view and does not explore the social influence between users [20]. But in real-world, as we often turn to our friends for recommendations, the social relationships are also the key factors to affect our decisions. To utilize the social influence to improve the recommendation accuracy, we follow the method of Wang et al. [25] and propose a social-enhanced CMF-I method ConvolutionalMatrixFactorizationwithSocialandItemRelations (CMF-SI), which exploits a shared user latent factor to bridge the latent space between user’s social network and rating matrix. Figure 4 shows the graphical model of CMF-SI.

Let $\mathbf{S} = (s_{ij})_{N,N}$ be the adjacency matrix of social network $\mathcal{G}_S$, with each entry $s_{ij}$ denoting whether user $u_i$ has social connection with user $u_j$; that is, 1 if $u_i$ directly connects to $u_j$ and 0 otherwise. As in CMF-I, we adopt a similar matrix factorization technique to factorize $\mathbf{S}$ to learn low-dimensional latent representation of user’s social relationship. Let $\mathbf{U} \in \mathbb{R}^{k \times N}$ and $\mathbf{Z} \in \mathbb{R}^{k \times N}$ be the latent user and social feature factor matrices, with columns $\mathbf{U}_i$ and $\mathbf{Z}_i$ representing user-specific and social-specific latent factors, respectively. Then, the conditional distribution over the observed social relation matrix $\mathbf{S}$ can be defined as

$$p(\mathbf{S} | \mathbf{U}, \mathbf{Z}, \sigma^2_S) = \prod_{i=1}^{N} \prod_{j=1}^{N} \mathcal{N}(s_{ij} | \mathbf{U}^T_i \mathbf{Z}_j, \sigma^2_S),$$

where $\sigma^2_S$ is the indicator function. Similar to the item relation factor $\mathbf{K}$, a zero-mean spherical Gaussian prior is also applied to the social latent factor $\mathbf{Z}$:

$$p(\mathbf{Z} | \sigma^2_Z) = \prod_{j=1}^{N} \mathcal{N}(\mathbf{Z}_j | 0, \sigma^2_Z \mathbf{I}).$$

As we have shown in Figure 4, we incorporate the social relation into CMF-I by sharing the user latent factor $\mathbf{U}$ between user’s social network and the user-item rating matrix; that is, the user latent factor $\mathbf{U}$ is not only affected by social network $\mathbf{S}$ but also the rating matrix $\mathbf{R}$. The joint probability distribution of CMF-SI can be written as

$$p(\mathbf{U}, \mathbf{V}, \mathbf{K}, \mathbf{W}^1, \mathbf{C}, \mathbf{S}, \mathbf{X}, \sigma^2_U, \sigma^2_C, \sigma^2_V, \sigma^2_U, \sigma^2_K, \sigma^2_W) \propto p(\mathbf{R} | \mathbf{U}, \mathbf{V}, \sigma^2) p(\mathbf{C} | \mathbf{K}, \mathbf{V}, \sigma^2_C) p(\mathbf{U} | \sigma^2_U) p(\mathbf{V}^1 | \sigma^2_V) p(\mathbf{W}^1 | \sigma^2_W),$$

$$p(\mathbf{V}^1, \mathbf{X}, \sigma^2_V) p(\mathbf{K} | \sigma^2_K) p(\mathbf{W}^1 | \sigma^2_W).$$

2.5. Recommendation with User Reviews. In CMF-SI, the relationships of both users and items and item review texts are modeled jointly in the ConvMF framework. However, it only considers the document contextual information of items, and the user’s document context is not investigated, which is independent of the item’s context information and should be treated as different recommendation factors. To model user’s review texts, we follow the work in [11, 25] and further incorporate another CNN module into CMF-SI, which arrives at our final recommendation method joint convolutional matrix factorization (ICMF).

The graphical model of ICMF is shown in Figure 5, from which we can observe that a user latent factor is generated from three variables (similar to the item latent factor): (1) internal weights $\mathbf{W}^2$ in CNN, (2) the review information (denoted by $\mathbf{Y}_i$) of user $i$, and (3) the Gaussian noise variable $\epsilon_i$. Thus, the user latent factor $\mathbf{U}_i$ can be obtained by the following equations:

$$\mathbf{U}_i = \text{cnn}(\mathbf{W}^2, \mathbf{Y}_i) + \epsilon_i,$$

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2_E).$$

For the internal weight $\mathbf{W}^2$ in CNN, we also place zero-mean spherical Gaussian prior on it:

$$p(\mathbf{W}^2 | \sigma^2_W) = \prod_k \mathcal{N}(\mathbf{w}^2_k | 0, \sigma^2_W).$$
Accordingly, the conditional distribution of user latent factor can be modified as

\[
p(U \mid W^2, Y, \sigma_v^2) = \prod_i^N \mathcal{N}(U_i \mid \text{cnn}(W^2, Y_i), \sigma_U^2 I), \quad (14)
\]

where \(\text{cnn}(W^2, Y_i)\) is the user latent factor learned from CNN module, which try to capture the user latent features from user's reviews and user-item rating matrix simultaneously. Then, the joint probability distribution of JCMF can be written as

\[
p(U, V, K, Z, W^1, W^2 \mid R, C, S, \mathcal{X}, Y, \sigma, \sigma^2_C, \sigma^2_U, \sigma^2_V, \sigma^2_W) = p(R \mid U, \sigma^2)p(C \mid K, V, \sigma_C^2)p(S \mid U, Z, \sigma_Z^2) \cdot p(U \mid W^2, Y, \sigma_v^2)p(V \mid W^1, \mathcal{X}, \sigma_v^2)p(K \mid \sigma_K^2)p(Z \mid \sigma_Z^2)p(W^1 \mid \sigma_{W^1}^2)p(W^2 \mid \sigma_{W^2}^2).
\]

\[
\max_{U,V,K,Z,W^1,W^2} p(U, V, K, Z, W^1, W^2 \mid R, C, S, \mathcal{X}, Y, \sigma, \sigma^2_C, \sigma^2_U, \sigma^2_V, \sigma^2_W) = \max_{U,V,K,Z,W^1,W^2} \left[ p(R \mid U, V, \sigma^2)p(C \mid K, V, \sigma_C^2)p(S \mid U, Z, \sigma_Z^2) p(U \mid W^2, Y, \sigma_v^2)p(V \mid W^1, \mathcal{X}, \sigma_v^2) p(K \mid \sigma_K^2)p(Z \mid \sigma_Z^2)p(W^1 \mid \sigma_{W^1}^2)p(W^2 \mid \sigma_{W^2}^2) \right].
\]

\[
\mathcal{L}(U, V, K, Z, W^1, W^2) = \frac{1}{2} \sum_{i} \sum_{j} L_{ij}(r_{ij} - U_i^T V_j)^2 + \frac{\lambda_C}{2} \sum_{i} \sum_{j} f_i^C(c_{ij} - K_i^T V_j)^2 + \frac{\lambda_S}{2} \sum_{i} \sum_{j} f_i^S(s_{ij} - U_i^T Z_j)^2 + \frac{\lambda_U}{2} \sum_{i} \|U_i - \text{cnn}(W^2, Y_i)\|_F^2 + \frac{\lambda_V}{2} \sum_{j} \|V_j - \text{cnn}(W^1, X_j)\|_F^2 + \frac{\lambda_K}{2} \sum_{i} \|K_i\|_F^2 + \frac{\lambda_Z}{2} \sum_{j} \|Z_j\|_F^2 + \frac{\lambda_W^1}{2} \sum_{k} \|W_{ki}^1\|^2 + \frac{\lambda_W^2}{2} \sum_{k} \|W_{ki}^2\|^2,
\]

where \(\lambda_C = \sigma^2/\sigma_C^2, \lambda_S = \sigma^2/\sigma_S^2, \lambda_U = \sigma^2/\sigma_U^2, \lambda_V = \sigma^2/\sigma_V^2, \lambda_K = \sigma^2/\sigma_K^2, \lambda_Z = \sigma^2/\sigma_Z^2, \lambda_W^1 = \sigma^2/\sigma_W^1, \) and \(\lambda_W^2 = \sigma^2/\sigma_W^2\) are the regularization parameters. \(\|\cdot\|_F\) is the Frobenius norm.

Similar to ConvMF, we optimize the above loss function by adopting the coordinate descent; that is, we iteratively update a latent variable while fixing the remaining variables. For example, when updating \(U\), we temporarily assume \(V, K, Z, W^1, \) and \(W^2\) are constant. Then, the loss function \(\mathcal{L}\) (denoted by equation (17)) becomes a quadratic function w.r.t \(U\). The analytical optimal solution of \(U\) can be computed in a closed form. The analytical solutions of \(U_j, V_j, K_j, \) and \(Z_j\) are shown as follows:

\[
U_i \leftarrow \left( VI_i V^T + \lambda_S Z_i I_i Z_i^T + \lambda_U I_i \right)^{-1} \cdot VR_i + \lambda_U \text{cnn}(W^2, Y_i) + \lambda_S Z_i S_i,
\]

\[
V_j \leftarrow \left( UI_j U^T + \lambda_C K_j K_j^T + \lambda_V I_j \right)^{-1} \cdot UR_j + \lambda_V \text{cnn}(W^1, X_j) + \lambda_C K_j C_j,
\]

\[
K_i \leftarrow \left( \lambda_C VI_i V^T + \lambda_K I_i \right)^{-1} \lambda_C C_j,
\]

\[
Z_j \leftarrow \left( \lambda_S UI_j U^T + \lambda_Z I_j \right)^{-1} \lambda_S S_j.
\]
where $I_i$, $I_i^C$, and $I_i^F$ are diagonal matrices and $R$, $S$, and $C$ are column vectors of matrices $R$, $S$, and $C$, respectively.

However, because $W^1$ and $W^2$ are related to the CNN architecture such as convolution layer and max-pooling layer, they cannot be updated by analytical solutions as we do for $U$, $V$, $K$, and $Z$. Nonetheless, when $U$, $V$, $K$, and $Z$ are temporarily constant, we find that $L$ can be interpreted as a squared error function with $L_j$ regularization terms. The loss function of $W^1$ and $W^2$ is reached as follows:

$$\epsilon(W^1) = \frac{\lambda}{2} \sum_j \left\| V_j - \text{cnn}(W^1, X_j) \right\|^2_F + \lambda \sum_k \left\| u^1_k \right\|^2 + \text{constant},$$

$$\epsilon(W^2) = \frac{\lambda}{2} \sum_i \left\| U_i - \text{cnn}(W^2, Y_i) \right\|^2_F + \lambda \sum_k \left\| w^2_k \right\|^2 + \text{constant}. \tag{22}$$

To optimize $W^1$ and $W^2$, we use back propagation on $\epsilon(W^1), \epsilon(W^2)$. The learning algorithm of JCMF is shown in Algorithm 1.

3. Experimental Setup

This section first introduces the research questions that we answered in experiments and then describes the dataset and baseline methods that we utilized in this work.

3.1. Research Questions. We conduct extensive experiments on the real-world data Yelp to answer the following research questions:

- RQ1: how does our proposed method JCMF perform compared with other state-of-the-art recommendation algorithms?
- RQ2: is the item relation network useful for improving the recommendation performance?
- RQ3: is the user social network useful for improving the recommendation results?
- RQ4: how does the document contextual information of users contribute to the performance of CR recommendation?
- RQ5: can our model JCMF converge quickly compared to the ConvMF method?

3.2. Dataset. In experiments, we use the real-world dataset Yelp as our data source (https://www.yelp.com/dataset). Yelp is an online review website that allows users to express their opinions by posting reviews to items (such as restaurants and businesses). Each user can also create social connections with the users that have similar interests with him/her. The dataset we used consists of users, reviews, and businesses across 11 metropolitan areas in 4 countries. We use this dataset for business recommendation. As the original dataset is too large, we only select the businesses that located in North Carolina (NC), Wisconsin (WI), and Ohio (OH) to conduct experiments. To reduce the data sparsity, we only keep users that have more than 3 ratings and businesses that at least have one review. The resulting NC dataset consists of 22,737 users who have rated and reviewed a total of 12,502 different items. The WI dataset consists of 8,386 users who have rated and reviewed a total of 4,593 different items. The OH dataset consists of 29,918 users who have rated and reviewed a total of 12,361 different items. The statistics of these three subdatasets is shown in Table 2.

3.3. Evaluation Protocols. In experiments, we randomly select 80% of the user-item ratings as the training set ($D_{train}$), 10% of the data as the validation set ($D_{valid}$), and use the remaining for testing ($D_{test}$). The optimal hyperparameters are determined on the validation set.

We employ two popular metrics [26] root mean squared error (RMSE) and mean average error (MAE) as the evaluation methods, where MAE measures the average magnitude of the errors in a set of predictions while RMSE tends to disproportionately penalize large errors. This also means RMSE is more prone to be affected by outliers or bad predictions. The formal definition of RMSE is written as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i,j} D_{test}(r_{ij} - \bar{r}_{ij})^2}{|D_{test}|}}, \tag{24}$$

where $r_{ij}$ is the observed rating value from user $u_i$ to item $v_j$, $\bar{r}_{ij}$ is the corresponding prediction, and $|D_{test}|$ is the number of the samples in the test set. The formal definition of MAE is written as

$$\text{MAE} = \frac{\sum_{i,j} D_{test}|r_{ij} - \bar{r}_{ij}|}{|D_{test}|}. \tag{25}$$

3.4. Baseline Methods. To evaluate the effectiveness of our proposed JCMF method, we compare it with the following methods.

- (i) PMF [27]: this is a matrix factorization-based recommendation method, which only utilizes the user-item rating matrix to learn user’s latent feature factors.
- (ii) SoRec [20]: this is a social trust-aware recommendation method that factorizes the user-item rating matrix and user’s social trust network jointly.
- (iii) RSTE [28]: learning to recommend with social trust ensemble is a classical algorithm which predicts user’s ratings by combining their own tastes and their trusted friends’ favors.
SocialMF [29]: this is a state-of-the-art social recommendation method that exploits social influence to enhance the recommendation process.

TrustMF [30]: this is another state-of-the-art social recommendation method, which integrates the user’s trust network into the matrix factorization.

TrustSVD [31]: this method is also a typical social trust-based recommendation method which considers both explicit and implicit influence of user trust.

SocialFD [32]: the method combines the matrix factorization model and distance metric learning. The positions of users and items are jointly determined by ratings and social relations.

DeepCoNN [33]: this is a state-of-the-art context-aware recommendation method that utilizes two parallel CNNs to process the views and integrates the user feature and item feature into the top-level factorization machine to predict ratings.

ConvMF [11]: this is another state-of-the-art context-aware recommendation method that integrates a CNN into PMF to learn a deeper understanding of the item’s document contextual information.

To evaluate the importance of the item relationship, social network, and user’s reviews, we further compare JCMF with the following methods:

CMF-I: this is a variant version of JCMF, which only considers the correlation network of items to make recommendations. This is to demonstrate the importance of the item relationships.

CMF-SI: this is another variant version of JCMF that jointly considers the item correlation network and user social network. Compared with JCMF, it does not model the user’s review information.

Table 2: The statistics of NC, WI, and OH datasets.

| Statistics         | NC    | WI    | OH    |
|--------------------|-------|-------|-------|
| # users            | 22,737| 8,386 | 29,918|
| # items            | 12,502| 4,593 | 12,361|
| # ratings          | 225,580| 80,643| 201,540|
| # item relations   | 299,632| 118,351| 249,055|
| # social relations | 111,394| 34,099| 120,104|

Algorithm 1: The learning algorithm of JCMF.
3.5. Implementation Details. We implement JCMF using Python and Keras library with NVidia Tesla K80 GPU. The dimensions of the latent factors U, V, K, and Z are set as 50 for both of these two datasets. For the NC dataset, the regularization parameters are set as $\lambda_U = 15$, $\lambda_V = 10$, $\lambda_K = 70$, $\lambda_C = 0.1$, $\lambda_Z = 15$, and $\lambda_S = 150$. For the WI dataset, the regularization parameters are set as $\lambda_U = 8$, $\lambda_V = 5$, $\lambda_K = 60$, $\lambda_C = 0.0001$, $\lambda_Z = 70$, and $\lambda_S = 70$. For the OH dataset, the regularization parameters are set as $\lambda_U = 1$, $\lambda_V = 100$, $\lambda_K = 70$, $\lambda_C = 0.1$, $\lambda_Z = 500$, and $\lambda_S = 70$. Similar to [3, 11], we preprocess the reviews of both users and items by the following steps: (1) remove stop words, (2) calculate the tf-idf score for each word, (3) remove the words whose document frequency scores are higher than 0.5, (4) select 8,000 words with highest tf-idf scores as a vocabulary, and (5) remove all nonvocabulary words from the raw review documents.

For the CNN architectures of both users and items, (1) the maximum length of review documents is set to 300. (2) The word latent vectors are randomly initialized with dimension 200. (3) In the convolution layer, we use various window sizes (3, 4, and 5) for shared weights to consider various length of surrounding words and set the number of shared weights per window size as 100. (4) To prevent CNN from overfitting, we use dropout and set the dropout ratio to 0.2 for these three datasets. (5) To train the weights of CNN, we use minibatch-based RMSprop and set the minibatch size as 64. For all the hyper-parameters, we tune them on the validation set and repeat each setting for 5 times and report the average results. The details of tuning the hyperparameters are shown in Section 5. After publishing, we will release our dataset and source code.

4. Experimental Results (RQ1)

Figure 6 reports the experimental results on NC, WI, and OH in terms of RMSE and MAE, from which we have the following observations: (1) Our JCMF method achieves the best performance on three datasets (the improvement between JCMF and other baseline methods is significant with $p < 0.01$). This result demonstrates the effectiveness of our JCMF solution, that is, jointly considering the relationships of users and items and the review documents from both users and items are helpful for making recommendations in the CR task. (2) Neural network-based recommendations (i.e., DeepCoNN, ConvMF, CMF-I, CMF-SI, and JCMF) can perform better than traditional MF-based methods, showing the effectiveness of the neural networks in learning the user-item interactions, because of their ability of capturing the nonlinear and high-level latent features of users and items. (3) ConvMF achieves a better performance than traditional social recommendation methods (i.e., SoRec, SociaLFM, TrustMF, RSTE, and SocialFD). This indicates that the document contextual information is helpful for the CR task, and considering them jointly can make more accurate recommendations than utilizing the social network information only.

5. Model Analysis

This section investigates the importance of model components and hyperparameters.

5.1. Importance of Item Relations (RQ2). To understand the importance of item relations in the CR task, we first compare CMF-I with ConvMF and then tune the hyperparameter $\lambda_C$ in JCMF to see how JCMF depends on the item relations. The hyperparameter $\lambda_C$ in JCMF represents how much item relation information is utilized in JCMF. If $\lambda_C = 0$, JCMF will only utilize the rating matrix, item’s review document, and user’s social network to make recommendations. If $\lambda_C = \infty$, JCMF will only utilize the item correlation information to learn the items’ latent factors.

The experimental results are shown in Figure 6 and Table 3, from which we have the following observations: (1) CMF-I significantly outperforms ConvMF on both datasets ($p < 0.01$), showing that the item relationship is beneficial to context-aware recommendations, and only consider the item’s review documents cannot get better results than combining them together. (2) The performance of JCMF changes significantly with varying the value of $\lambda_C$. In both NC, WI, and OH, with the increase in $\lambda_C$, the performance of JCMF first increases (RMSE and MAE decrease). But when $\lambda_C$ surpasses a certain threshold (0.1 in NC and OH and 0.0001 in WI), the performance of JCMF will decrease with further increasing the value of $\lambda_C$. This result, again, demonstrates the importance of item relations; that is, we can further improve the recommendation performance by the shared item latent factor.

5.2. Importance of Social Relations (RQ3). To validate the effectiveness of social relations, we compare CMF-SI with CMF-I and tune the hyperparameter $\lambda_S$ in JCMF to investigate the impact of the social relationships. The hyperparameter $\lambda_S$ controls how much JCMF will depend on user’s social influence. It plays a role to balance the information from user-item rating matrix and user’s social network.

Figure 6 and Table 4 present the experimental result of JCMF on NC, WI, and OH. From these results, we have the following observations: (1) CMF-SI achieves a better performance than CMF-I on both three datasets (as shown in Figure 6). This result demonstrates the importance of social relations and indicates that considering the user’s social influence is helpful for the CR task. (2) The hyperparameter $\lambda_S$ impacts the performance of JCMF significantly (as shown in Table 4). With the increase in $\lambda_S$, the recommendation performance of JCMF increases as first, but when $\lambda_S$ achieves a certain value (150 in NC and 70 in WI and OH), JCMF begins to decrease with further increasing the value of $\lambda_S$. From the above results, we can observe that social influence is important to the CR-based recommendations, jointly considering the social network and user-item ratings can significantly improve the recommendation performance.
5.3. Importance of User’s Review Documents (RQ4). To demonstrate the importance of user’s review documents and answer the RQ4, we further compare JCMF with CMF-SI and tune the hyperparameter $\lambda_U$ in JCMF on NC, WI, and OH, which is a parameter that balances between the importance of ratings and the review documents of users. From

![Comparison results on NC, WI, and OH.](image)

Figure 6: Comparison results on NC, WI, and OH. The significant improvements of JCMF over other baseline methods are given by the $t$-test with $p < 0.01$: (a) RMSE on NC; (b) MAE on NC; (c) RMSE on WI; (d) MAE on WI; (e) RMSE on OH; (f) MAE on OH.
whereas the performance of JCMF suddenly drops with reach the best performance on NC, WI, and OH respectively, in this section, we further 5.4. Convergence Analysis (RQ5). In this section, we further conduct experiments to compare the convergence performance of JCMF and ConvMF on NC, WI, and OH. In experiments, we set the max interaction as 200 and adopt early stopping when getting the expected prediction errors. The experimental results are shown in Figure 7, from which we can observe that the JCMF has similar convergence performance with ConvMF while achieving a better recommendation result. Incorporating the relations of users and items and the user’s review documents do not slow down the convergence of JCMF. In some cases, it can even converge faster than ConvMF (as shown in Figure 7(c)). This result directly demonstrates the efficiency of our JCMF method and answers the RQ5.

5.4. Convergence Analysis (RQ5). In this section, we further conduct experiments to compare the convergence performance of JCMF and ConvMF on NC, WI, and OH. In experiments, we set the max interaction as 200 and adopt early stopping when getting the expected prediction errors. The experimental results are shown in Figure 7, from which we can observe that the JCMF has similar convergence performance with ConvMF while achieving a better recommendation result. Incorporating the relations of users and items and the user’s review documents do not slow down the convergence of JCMF. In some cases, it can even converge faster than ConvMF (as shown in Figure 7(c)). This result directly demonstrates the efficiency of our JCMF method and answers the RQ5.

Table 3: Impact of parameter $\lambda_C$ in JCMF on NC, WI, and OH.

| $\lambda_C$ | NC RMSE | MAE | WI RMSE | MAE | OH RMSE | MAE |
|-------------|---------|-----|---------|-----|---------|-----|
| $1 \times 10^{-6}$ | 1.158 | 0.902 | $1 \times 10^{-6}$ | 1.050 | 0.870 | 1.216 | 0.956 |
| $1 \times 10^{-5}$ | 1.158 | 0.893 | $5 \times 10^{-5}$ | 1.103 | 0.866 | 1.212 | 0.951 |
| $1 \times 10^{-4}$ | 1.159 | 0.893 | $1 \times 10^{-4}$ | 1.101 | 0.862 | 1.211 | 0.949 |
| 0.001 | 1.158 | 0.892 | 0.005 | 1.104 | 0.865 | 1.210 | 0.947 |
| 0.05 | 1.157 | 0.892 | 0.05 | 1.105 | 0.868 | 0.001 | 1.211 | 0.947 |
| 0.1 | 1.155 | 0.889 | 0.1 | 1.106 | 0.869 | 0.1 | 1.209 | 0.945 |
| 0.5 | 1.163 | 0.903 | 0.5 | 1.120 | 0.880 | 0.5 | 1.212 | 0.946 |
| 0.8 | 1.171 | 0.913 | 0.8 | 1.128 | 0.891 | 0.8 | 1.214 | 0.953 |

Table 4: Impact of parameter $\lambda_S$ in JCMF on NC, WI, and OH.

| $\lambda_S$ | NC RMSE | MAE | WI RMSE | MAE | OH RMSE | MAE |
|-------------|---------|-----|---------|-----|---------|-----|
| 0.01 | 1.193 | 0.930 | 0.01 | 1.143 | 0.887 | 0.01 | 1.377 | 0.961 |
| 1 | 1.191 | 0.926 | 0.01 | 1.143 | 0.885 | 1 | 1.272 | 0.957 |
| 0.5 | 1.162 | 0.902 | 0.01 | 1.102 | 0.878 | 10 | 1.240 | 0.955 |
| 120 | 1.158 | 0.892 | 70 | 1.101 | 0.862 | 30 | 1.226 | 0.951 |
| 140 | 1.157 | 0.890 | 100 | 1.103 | 0.864 | 70 | 1.209 | 0.945 |
| 150 | 1.155 | 0.889 | 150 | 1.106 | 0.867 | 100 | 1.215 | 0.946 |
| 175 | 1.158 | 0.892 | 200 | 1.110 | 0.870 | 200 | 1.223 | 0.949 |
| 300 | 1.163 | 0.899 | 500 | 1.137 | 0.883 | 300 | 1.229 | 0.956 |

Table 5: Impact of parameter $\lambda_U$ in JCMF on NC, WI, and OH.

| $\lambda_U$ | NC RMSE | MAE | WI RMSE | MAE | OH RMSE | MAE |
|-------------|---------|-----|---------|-----|---------|-----|
| 0.1 | 1.296 | 1.000 | 0.1 | 1.202 | 0.940 | 0.01 | 1.490 | 1.147 |
| 1 | 1.227 | 0.951 | 1 | 1.147 | 0.908 | 0.1 | 1.336 | 1.029 |
| 5 | 1.169 | 0.900 | 5 | 1.111 | 0.873 | 0.5 | 1.228 | 1.102 |
| 10 | 1.159 | 0.890 | 8 | 1.101 | 0.862 | 1 | 1.209 | 0.945 |
| 15 | 1.155 | 0.889 | 10 | 1.103 | 0.864 | 5 | 1.225 | 0.956 |
| 20 | 1.160 | 0.896 | 15 | 1.109 | 0.868 | 10 | 1.234 | 0.962 |
| 50 | 1.173 | 0.925 | 20 | 1.111 | 0.868 | 20 | 1.236 | 0.965 |
| 100 | 1.180 | 0.939 | 50 | 1.124 | 0.875 | 50 | 1.331 | 0.978 |
| 150 | 1.208 | 0.994 | 100 | 1.154 | 0.925 | 100 | 1.339 | 0.981 |

6. Related Work

This section briefly reviews the related works from three aspects: social recommendation, context-aware recommendation, and CNN-based content modeling method.

6.1. Social Recommendation. Due to the recent development of social networks, the methods that leverage the social network information to improve traditional recommender systems have been widely studied [20, 28–31, 34–38]. For example, to consider the social network information for making recommendations, Ma et al. [20] proposed a fused matrix factorization framework to model the user’s social influence and rating matrix simultaneously. Zhao et al. [36] incorporated social connections into the local low-rank framework and solved the problem of constructing the submatrices by social regularization terms. Based on the intuition that users always prefer the items recommended by the friends they trust, Ma et al. [28] fused the users and their trusted friends’ tastes together by an ensemble parameter. Guo et al. [31] extended the state-of-the-art recommendation algorithm SVD++ by further incorporating both the explicit and implicit influence of trusted users on the prediction of items for an active user.

Although the above methods have achieved great success in modeling user’s social influence, they only focus on the social relationships, and other kinds of context information (such as description content, labels, and tags) are not investigated. To further improve the performance of social recommendation, recent works start to find ways to merge these information [6, 21, 22, 39]. For example, Jiang et al.
Figure 7: Convergence analysis of JCMF and ConvMF on NC, WI, and OH: (a) RMSE on NC; (b) MAE on NC; (c) RMSE on WI; (d) MAE on WI; (e) RMSE on OH; (f) MAE on OH.
[39] explored the social recommendation on the basis of psychology and sociology studies and incorporated two social context factors (i.e., individual preference and interpersonal influence) into the probabilistic matrix factorization framework. Hu et al. [40] investigated the effectiveness of fusing social relations and review texts to rating prediction by the alignment between the latent factors found by social matrix factorization (SocialMF) and topics found by topic MF. However, the above methods only understand the content information in a shallow way, and the performance of modeling the document content by a CNN network in social networks is still unexplored.

6.2. Context-Aware Recommendation. To alleviate the inherent data sparsity issue of recommender systems, context-aware recommendation (CR) methods have been widely studied in recent years, which aim to leverage the context information (such as temporal information, location, and content) to improve the performance of traditional recommender systems. For example, Zhang and Chow [18] proposed a probabilistic framework called TICRec by taking into account time context information for recommendation. In this work, TICRec utilized temporal influence correlations of both weekdays and weekends for time-aware location recommendation and estimated the probability density over the continuous time of a user visiting a location to avoid the loss of time information. To incorporate multitypes of context for recommendation, Karatzoglou et al. [41] extended the concept of matrix factorization to that of tensor factorization and proposed the high-order singular value decomposition (HOSVD) method to construct a tensor decomposition model of "user, commodity, and context" to personalize recommendations. Baltrunas et al. [42] extended the classical MF approach by taking into account contextual information in the rating prediction step of a recommender system. Yao et al. [43] employed tensor factorization to model the multidimensional contextual information, where they exploited a high-order tensor to fuse heterogeneous contextual information about check-ins, rather than the traditional two-dimensional user-location matrix.

Besides the temporal information, locations, and profiles, the content contexts are also been considered recently. For example, Gao et al. [44] investigated various types of content information on LBSNs in terms of sentiment indications, user interests, and POI properties and then modeled the three types of information under a unified POI recommendation framework. Wang et al. [45] proposed a preference enrichment approach via leveraging auxiliary data of online reviewers’ aspect-level opinions so as to predict the buyers’ missing preferences. To consider the opinion spam in product reviews, Raghavan et al. [46] proposed a collaborative filtering method using quality scores consisting of two stages; that is, they first estimated the quality scores for individual ratings in the dataset. Then, they utilized the quality scores estimated from the previous stage as weights for the ratings in the probabilistic matrix factorization framework. However, the above content-based methods cannot model the contextual information of the content, leading to a shallow understanding of the content semantics. Furthermore, none of the above methods model the relationships and content context of both users and items simultaneously, which motivates us to develop a joint context-aware recommendation method. Recently, Wang et al. [25] explored document review and social relations to improve traditional recommendation, but they assume items are i.i.d, and did not investigate the influence of item relationships in the CR task.

6.3. CNN-Based Content Modeling Method. CNN as one of the most powerful feature representation techniques has achieved great success in many research areas, such as computer vision and natural language processing. Recently, researchers found that CNN can also be utilized to model the document content information. For example, as CNN can effectively capture the local features of documents through modeling components such as local receptive fields and shared weighted, Kim et al. [11] utilized CNN to capture the contextual information of documents and proposed a convolutional matrix factorization (ConvMF) method to enhance the rating prediction accuracy. To cope with the ambiguity and variability of linguistic expression, He et al. [47] modeled each sentence using a CNN that extracts features at multiple levels of granularity and uses multiple types of pooling, and proposed a model for comparing sentences that use a multiplicity of perspectives. Ruder et al. [48] used a CNN for both aspect extraction and aspect-based sentiment analysis and proved its effectiveness in sentence modeling tasks. Zheng et al. [33] utilized two parallel CNNs to jointly model the reviews text from both users and items and proposed a deep collaborative neural network (DeepCoNN). In their method, a shared is introduced on the top to couple these two networks together. However, DeepCoNN does not consider the relationships of users and items and can only achieve the supoptimal result. The above methods demonstrate the ability of CNN in extracting document features and provide us an effective way to mine the semantic clues from review documents.

7. Conclusions

In this work, we investigated the CR problem and proposed a joint convolutional matrix factorization (JCMF) to jointly consider the review text and connections of both users and items in a unified recommendation framework. More specifically, to consider the impact of item’s correlations, we assume items that are bought/clicked by the same user within a short time window having a strong correlation with each other. Then, we incorporated the predefined correlation network into ConvMF by a shared item latent feature space. To consider the impact of user’s social influence, we further integrate the social relationship of users by bridging the user latent space between user-item matrix and social network. To consider the impact of user’s review document, we exploit another CNN module to extract user’s document contextual information. The experimental results on the real-world
dataset demonstrate the superiority of our proposed method compared with other baseline methods.

Data Availability

The data supporting this study are from previously reported studies and datasets, which have been cited. The processed data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61602282) and China Postdoctoral Science Foundation (No. 2016M602181).

References

[1] S. Li, J. Kawale, and Y. Fu, “Deep collaborative filtering via marginalized denoising auto-encoder,” in Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM ’15), pp. 811–820, ACM, Melbourne, Australia, October 2015.

[2] Y. Bao, H. Fang, and J. Zhang, “Topicmf: simultaneously exploiting ratings and reviews for recommendation,” in Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Quebec, Canada, July 2014.

[3] H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative deep learning for recommender systems,” in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1235–1244, ACM, Sydney, Australia, August 2015.

[4] D. Kim, C. Park, J. Oh, and H. Yu, “Deep hybrid recommender systems via exploiting document context and statistics of items,” Information Sciences, vol. 417, pp. 72–87, 2017.

[5] Y. Tan, M. Zhang, Y. Liu, and S. Ma, “Rating-boosted latent topics: understanding users and items with ratings and reviews,” in Proceedings of the International Joint Conferences on Artificial Intelligence, vol. 16, pp. 2640–2646, New York, NY, USA, July 2016.

[6] C. Chen, X. Zheng, Y. Wang, F. Hong, and Z. Lin, “Context-aware collaborative topic regression with social matrix factorization for recommender systems,” in Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Quebec, Canada, July 2014.

[7] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, and X. Luo, “A context-aware user-item representation learning for item recommendation,” ACM Transactions on Information Systems (TOIS), vol. 37, no. 2, pp. 1–29, 2019.

[8] X. Zhou, D. Qin, L. Chen, and Y. Zhang, “Real-time context-aware social media recommendation,” The VLDB Journal, vol. 28, no. 2, pp. 197–219, 2019.

[9] C. Wang and D. M. Blei, “Collaborative topic modeling for recommending scientific articles,” in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’11), pp. 448–456, ACM, San Diego, CA, USA, August 2011.

[10] J. McAuley and J. Leskovec, “Hidden factors and hidden topics: understanding rating dimensions with review text,” in Proceedings of the 7th ACM Conference on Recommender Systems, pp. 165–172, ACM, Hong Kong, China, October 2013.

[11] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, “Convolutional matrix factorization for document context-aware recommendation,” in Proceedings of the 10th ACM Conference on Recommender Systems, pp. 233–240, ACM, Boston, MA, USA, September 2016.

[12] Y. Koren, “Factorization meets the neighborhood: a multi-faceted collaborative filtering model,” in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434, ACM, Las Vegas, NV, USA, August 2008.

[13] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: item-to-item collaborative filtering,” IEEE Internet Computing, vol. 7, no. 1, pp. 76–80, 2003.

[14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in Proceedings of the 10th International Conference on World Wide Web, pp. 285–295, ACM, Hong Kong, China, May 2001.

[15] X. Ning and G. Karypis, “Slim: sparse linear methods for top-n recommender systems,” in Proceedings of the 2011 IEEE 11th International Conference on Data Mining, pp. 497–506, IEEE, Vancouver, Canada, December 2011.

[16] S. Kabbur, X. Ning, and G. Karypis, “Fism: factored item similarity models for top-n recommender systems,” in Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 659–667, ACM, Chicago, IL, USA, August 2013.

[17] J.-D. Zhang, C.-Y. Chow, and Y. Li, “Lore: exploiting sequential influence for location recommendations,” in Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 103–112, ACM, Dallas, TX, USA, November 2014.

[18] J.-D. Zhang and C.-Y. Chow, “Trecf: a probabilistic framework to utilize temporal influence correlations for time-aware location recommendations,” IEEE Transactions on Services Computing, vol. 9, no. 4, pp. 633–646, 2015.

[19] M. McPherson, L. Smith-Lovin, and J. M. Cook, “Birds of a feather: homophily in social networks,” Annual Review of Sociology, vol. 27, no. 1, pp. 415–444, 2001.

[20] H. Ma, H. Yang, M. R. Lyu, and I. King, “Sorec: social recommendation using probabilistic matrix factorization,” in Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 931–940, ACM, Napa Valley, CA, USA, October 2008.

[21] J. Tang, X. Hu, H. Gao, and H. Liu, “Exploiting local and global social context for recommendation,” in Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 2013.

[22] X. Yang, H. Steck, and Y. Liu, “Circle-based recommendation in online social networks,” in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1267–1275, ACM, Beijing, China, August 2012.

[23] X. Shen, B. Yi, Z. Zhang, J. Shu, and H. Liu, “Automatic recommendation technology for learning resources with convolutional neural network,” in Proceedings of the 2016 International Symposium on Educational Technology (ISET), pp. 30–34, IEEE, Los Alamitos, CA, USA, July 2016.

[24] S. Zhang, L. Yao, and X. Xu, “Autosvd++: an efficient hybrid collaborative filtering model via contractive auto-encoders,”
in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 957–960, ACM, Tokyo, Japan, August 2017.

[25] X. Wang, X. Yang, L. Guo, Y. Han, F. Liu, and B. Gao, “Exploiting social review-enhanced convolutional matrix factorization for social recommendation,” *IEEE Access*, vol. 7, pp. 82826–82837, 2019.

[26] S. M. Täheri, H. Mahyar, M. Firouzi, E. K. Ghalebi, R. Grosu, and A. Movaghar, “Extracting implicit social relation for social recommendation techniques in user rating prediction,” in *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 1343–1351, International World Wide Web Conferences Steering Committee, Perth, Australia, April 2017.

[27] A. Mnih and R. R. Salakhutdinov, “Probabilistic matrix factorization,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1257–1264, Vancouver, Canada, December 2008.

[28] H. Ma, I. King, and M. R. Lyu, “Learning to recommend with social trust ensemble,” in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 203–210, ACM, Boston, MA, USA, July 2009.

[29] M. Jamali and M. Ester, “A matrix factorization technique with trust propagation for recommendation in social networks,” in *Proceedings of the Fourth ACM conference on Recommender Systems*, pp. 135–142, ACM, Barcelona, Spain, September 2010.

[30] B. Yang, Y. Lei, J. Liu, and W. Li, “Social collaborative filtering by trust,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.

[31] G. Guo, J. Zhang, and N. Yorke-Smith, “Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings,” *AAAI*, vol. 15, pp. 123–125, 2015.

[32] J. Yu, M. Gao, W. Rong, Y. Song, and Q. Xiong, “A social recommender based on factorization and distance metric learning,” *IEEE Access*, vol. 5, pp. 21557–21566, 2017.

[33] L. Zheng, V. Noroozi, and P. S. Yu, “Joint deep modeling of users and items using reviews for recommendation,” in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 425–434, ACM, Cambridge, UK, February 2017.

[34] Z. Sun, L. Han, W. Huang et al., “Recommender systems based on social networks,” *Journal of Systems and Software*, vol. 99, pp. 109–119, 2015.

[35] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, “Recommender systems with social regularization,” in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pp. 287–296, ACM, Hong Kong, China, February 2011.

[36] H. Zhao, Q. Yao, J. T. Kwok, and D. L. Lee, “Collaborative filtering with social local models,” in *Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM)*, pp. 645–654, New Orleans, LA, USA, November 2017.

[37] C.-H. Lai, S.-J. Lee, and H.-L. Huang, “A social recommendation method based on the integration of social relationship and product popularity,” *International Journal of Human-Computer Studies*, vol. 121, pp. 42–57, 2019.

[38] P. Liu, L. Zhang, and J. A. Gulla, “Real-time social recommendation based on graph embedding and temporal context,” *International Journal of Human-Computer Studies*, vol. 121, pp. 58–72, 2019.

[39] M. Jiang, P. Cui, F. Wang, W. Zhu, and S. Yang, “Scalable recommendation with social contextual information,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 11, pp. 2789–2802, 2014.

[40] G.-N. Hu, X.-Y. Dai, Y. Song, S.-J. Huang, and J.-J. Chen, “A synthetic approach for recommendation: combining ratings, social relations, and reviews,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, July 2015.

[41] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, “Multivariate recommendation: n-dimensional tensor factorization for context-aware collaborative filtering,” in *Proceedings of the Fourth ACM Conference on Recommender Systems*, pp. 79–86, ACM, Barcelona, Spain, September 2010.

[42] L. Baltrunas, B. Ludwig, and F. Ricci, “Matrix factorization techniques for context aware recommendation,” in *Proceedings of the Fifth ACM Conference on Recommender Systems*, pp. 301–304, ACM, Chicago, IL, USA, October 2011.

[43] L. Yao, Q. Z. Sheng, X. Wang, W. E. Zhang, and Y. Qin, “Collaborative location recommendation by integrating multi-dimensional contextual information,” *ACM Transactions on Internet Technology (TOIT)*, vol. 18, no. 3, p. 32, 2018.

[44] H. Gao, J. Tang, X. Hu, and H. Liu, “Content-aware point of interest recommendation on location-based social networks,” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, TX, USA, January 2015.

[45] F. Wang, W. Pan, and L. Chen, “Recommendation for new users with partial preferences by integrating product reviews with static specifications,” *User Modeling, Adaptation, and Personalization*, Springer, in *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*, pp. 281–288, Springer, Rome, Italy, June 2013.

[46] S. Raghavan, S. Gunasekar, and J. Ghosh, “Review quality aware collaborative filtering,” in *Proceedings of the Sixth ACM conference on Recommender Systems*, pp. 123–130, ACM, Dublin, Ireland, September 2012.

[47] H. He, K. Gimpel, and J. Lin, “Multi-perspective sentence similarity modeling with convolutional neural networks,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1576–1586, Lisbon, Portugal, September 2015.

[48] S. Ruder, P. Ghaffari, and J. G. Breslin, “Insight-1 at semeval-2016 task 5: deep learning for multilingual aspect-based sentiment analysis,” 2016, https://arxiv.org/abs/1609.02748.