

Research Article

DeepFusion: Fusing User-Generated Content and Item Raw Content towards Personalized Product Recommendation

Mingxin Gan  and Hang Zhang 

Department of Management Science and Engineering, Donlinks School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China

Correspondence should be addressed to Mingxin Gan; ganmx@ustb.edu.cn

Received 17 November 2019; Accepted 20 February 2020; Published 30 March 2020

Academic Editor: Chittaranjan Hens

Copyright © 2020 Mingxin Gan and Hang Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Personalized recommender systems, as effective approaches for alleviating information overload, have received substantial attention in the last decade. Learning effective latent factors plays the most important role in recommendation methods. Several recent works extracted latent factors from user-generated content such as ratings and reviews and suffered from the sparsity problem and the unbalanced distribution problem. To tackle these problems, we enrich the latent representations by incorporating user-generated content and item raw content. Deep neural networks have emerged as very appealing in learning effective representations in many applications. In this paper, we propose a novel deep neural architecture named DeepFusion to jointly learn user and item representations from numerical ratings, textual reviews, and item metadata. In this framework, we utilize multiple types of deep neural networks that are best suited for each type of heterogeneous inputs and introduce an extra layer to obtain the joint representations for users and items. Experiments conducted on the Amazon product data demonstrate that our approach outperforms multiple state-of-the-art baselines. We provide further insight into the design selections and hyper-parameters of our recommendation method. In addition, we further explore the relative importance of various item metadata information on improving the rating prediction performance towards personalized product recommendation, which is extremely valuable for feature extraction in practice.

1. Introduction

With the exploding growth of the network scale and the number of products, it is difficult for customers to deal with the large amount of available information. To alleviate information overload [1], personalized product recommendation has been utilized by e-commerce websites to present products that best meet customers' needs and expectations. The success of many e-commerce companies is due to their accurate and personalized product recommender systems, such as Amazon, eBay, Yelp, and Netflix [2].

Collaborative filtering (CF) [3–6] is one of the most successful recommendation algorithms in both industry and academic communities. The basic idea of the technique is that people who share similar ratings tend to have similar preferences. However, the CF-based method easily suffers

from the cold-start problem [7] when there are only a few ratings for items, which severely deteriorates the accuracy of recommendation.

Recently, researchers have found that additional data sources beyond ratings are extremely helpful in personalized recommendation. However, most existing recommender systems take into account only the user-generated content such as textual reviews in user/item profiling, while ignoring the raw content of items. First, the user-generated content is very sparse in many applications. For example, in the Netflix dataset [8], the number of movies that have been rated by users is only approximately 1 percent of the total number of movies. Second, the distribution of user-item interaction data is highly unbalanced. According to Anderson [9], due to the long-tail effect, only a few users interact with a large number of items, while most users rarely, or never, interact

with items. Besides, these additional data sources come in very different and heterogeneous forms, which make it difficult to fuse them in a unified way.

Fortunately, deep learning technique has shown powerful representation learning performance and outstanding scalability, which has shed light on this problem. Several deep learning-based recommender systems have been proposed. Yet, most of them [10–14] were usually restricted to limited data sources or learned the latent representations of users and items independently. As a result, these approaches cannot achieve fine-grained modeling of user preferences and item features. In addition, the metadata of product (item), i.e., price, brand, title, and description, on e-commerce websites plays an important role on user buying behaviors. However, to the best of our knowledge, no study has been conducted in which the relative importance of various item metadata information was considered.

In this paper, we propose a novel personalized recommendation method based on deep learning and multiview fusion, called DeepFusion, for rating prediction task. In this framework, each kind of data source is considered as a view and different views describe different aspects of user preferences and item features. Because of the nature of heterogeneous data sources, we utilize multiple types of deep neural networks, i.e., multilayer perceptrons (MLP) and convolutional neural networks (CNN), to make the best use of these inputs. Then, representations from each view are further mapped to a shared semantic space with a merged layer to obtain the integrated user/item representations. Finally, a multilayer perceptron layer is introduced to capture the complex relations between users and items. For model learning, the loss function is defined as the error between the predicted rating and the actual rating, and model parameters are adjusted via a backpropagation. Our proposed model is evaluated over the Amazon product data and compared with classic and state-of-the-art recommendation approaches. The experimental results demonstrate that DeepFusion significantly outperforms all baselines.

The main contributions of our work are summarized as follows:

- (i) We propose a novel personalized recommendation method based on deep learning and multiview fusion, called DeepFusion, for rating prediction task in product recommendation. The method is capable of incorporating user-generated content and item raw content including numerical ratings, textual reviews, and item metadata in a unified space.
- (ii) We utilize multiple types of deep neural networks that are best suited for each type of heterogeneous data sources to jointly learn user and item representations, which is beneficial to tackle the sparsity problem and the unbalanced distribution problem.
- (iii) We conduct a series of extensive experiments on a real-world dataset. The experimental results demonstrate that our approach outperforms all the baselines. And we further study the impact of the design selections and hyperparameters of our recommender system.
- (iv) To the best of our knowledge, we are the first to explore the relative importance of product (item) metadata, i.e., price, brand, title, and description, on user buying behaviors on e-commerce websites.

The remaining of the paper is organized as follows. In Section 2, we present the overview of related work. In Section 3, we describe our proposed model in depth. Then, we present the experimental results and analyses in Section 4. Finally, we conclude our work and introduce the future work in section 5.

2. Related Work

There have been extensive works on recommendation systems with a myriad of publication. In this section, we briefly review a representative set of approaches that are mostly related to our proposed approach.

2.1. Additional Data Sources for Recommender Systems.

In recent years, numerous works have been proposed to exploit additional data sources for personalized recommendation. A popular research line is the joint modeling of numerical ratings and textual reviews for recommendation. Textual reviews are able to express user opinions towards various item features. McAuley and Leskovec [15] proposed the hidden factors as topics (HFT) model. This model extracted latent topics from reviews via topic model latent dirichlet allocation [16] and associated topics with rating dimensions. Chambua et al. [17] introduced the linguistic similarity between review texts and incorporated it into the probabilistic matrix factorization (PMF) model [18]. Cheng et al. [19] emphasized the importance of considering users' varying attentions on different aspects and applied the aspect-aware topic model (ATM) on the review text to estimate the aspect attention weights of a user towards an item. Although the mixture of aspects discovered by topic-based methods may describe a corpus fairly well, aspects often consist of unrelated or loosely related concepts [20]. Therefore, failing to preserve the original order of words and ignoring their semantic meaning, the above methods cannot achieve the successful modeling of a given review. To tackle these limitations, researchers have paid extensive attentions to neural network methods. Zheng et al. [11] presented deep cooperative neural networks (DeepCoNN) for learning user behaviors and item properties by using two parallel CNNs. Chen et al. [12] aimed at exploiting the usefulness of reviews and developed a neural attention regression model for predicting ratings and selecting highly useful reviews simultaneously. Cheng et al. [21] developed a novel aspect-aware recommender model named A³NCF, which can capture a user's special attention on each aspect of the targeted item with an attention network.

In addition to textual reviews, additional data sources such as tags, item descriptions, item images, and user social networks have been used as supplemental information for sparse ratings. Ma et al. [22] investigated the combination of tags and genre information for identifying user interests via

an augmenting matrix factorization approach. Kim et al. [23] utilized a CNN to capture contextual information of item description documents and integrated it into the PMF method with consideration of the Gaussian noise. Cheng et al. [24] applied a proposed multimodal aspect-aware topic model (MATM) on textual reviews and item images to model user preferences and item features from different aspects. Huang et al. [25] constructed a hybrid multigroup coclustering recommendation framework to cluster users and items into multiple categories simultaneously, which fully utilized various data sources including ratings and user social networks. Qian et al. [26] fused three types of representative heterogeneous information to comprehensively analyze user features, such as ratings, user social networks, and user review sentiments. Zheng et al. [27] considered the evolving nature of user preferences over time and developed a time-sensitive and tag-aware recommendation framework. Bougiatiotis and Giannakopoulos [28] presented a content-based movie recommender system that was based on textual information and audio and visual channels.

2.2. Deep Learning for Recommender Systems. Recently, deep learning has realized tremendous success in recommender systems. In this section, we review several representative-related approaches. There are extensive works that combine neural network structures with collaborative filtering. Li et al. [29] learned effective latent representations via a deep architecture for CF, which coupled PMF with marginalized denoising stacked autoencoders (mDAs). Wang and Wang [30] integrated the deep belief network and the probabilistic graphical model and used the hybrid model to learn features from audio content. Wu et al. [31] designed a collaborative denoising autoencoder (CDAE) for top-N recommendation by training on a corrupted version of the known user-item interactions.

Due to the strong performance of deep learning on feature representation and combination, many deep models have been developed for learning the latent representations of users and items. Wang et al. [32] proposed collaborative deep learning (CDL), which learned a deep representation from movie content by using the generalized stacked autoencoder (SAE) model. Yu et al. [33] proposed an interactive attention mechanism to learn the latent representations of users and items and provided interpretable item recommendation. Zhou et al. [34] developed the deep interest network (DIN) for adaptively learning the representation of user interests via a local activation unit.

Several recent studies focused on the interactions between features. He et al. [35] proposed a general neural network-based collaborating filtering (NCF) approach for modeling nonlinear interactions between users and items. After that they [36] developed the neural factorization machine (NFM), which seamlessly combined the linearity of the factorization machine (FM) [37] and the nonlinearity of the neural network to capture second-order and higher-order feature interactions between users and items. Cheng et al. [38] designed the wide&deep learning model for enhancing the memorization and generalization performances

of recommender systems. Guo et al. [39] presented DeepFM, which is an end-to-end learning model, for emphasizing both low- and high-order feature interactions. Chambua et al. [40] established a hybrid recurrent neural network-long short-term memory (RNN-LSTM) architecture for learning user preferences with item aspects. Zhou et al. [41] proposed a deep interest evolution network (DIEN) for click-through rate prediction, which can capture temporal interests via an interest extractor layer.

Recently, several deep learning recommendation methods with multiview fusion have been proposed. Chen et al. [42] argued that, in multimedia recommendation, there exists item- and component-level implicitness which blur the underlying user preferences and proposed an attention mechanism in CF. In their model, the CNN was used to extract image features and video features. Zhang et al. [43] used three heterogeneous data sources including ratings, reviews, and image information to jointly model the user and item representations based on deep representation learning architectures. Gan et al. [44] adopted a convolution neural network to extract the hidden feature from item description and then fused it with tag information. Elkahky et al. [45] proposed a content-based cross-domain recommender system, which learned rich user and item features according to user web browsing histories and search queries. Tal and Liu [46] presented a textual and contextual embedding-based neural recommender (TCENR) for point-of-interest (POI) recommendation. Multiple types of deep neural networks were utilized to analyze additional data sources including social networks, geospatial locations, and textual reviews. Guo et al. [47] presented a multimodal representation learning method to predict user preferences based on multimodal content, including visual features, text features, audio features, and user interactive history in short video understanding and recommendation. Chen et al. [48] proposed a novel neural architecture for fashion recommendation based on both image region-level features and user review information. Cui et al. [49] presented a visual and textural recurrent neural network (VT-RNN), which simultaneously learned the sequential latent vectors of user's interest and captured the content-based representations. Though better performance against modeling ratings is achieved, the above models learned the latent representations of users and items independently. Thus, it is difficult for the above models to effectively model user preferences and item features.

3. DeepFusion Approach

3.1. Overview of DeepFusion. The architecture of our proposed method DeepFusion is illustrated in Figure 1. The model consists of three components: Reviews Modeling, Ratings Modeling, and Item Metadata Modeling. The Reviews Modeling component, which is presented in the left part of the figure, utilizes two parallel convolutional neural networks to extract the rich relevant textual features of users and items by jointly considering their corresponding reviews. Shown in the middle of Figure 1, the Ratings Modeling component uses a neural architecture of matrix

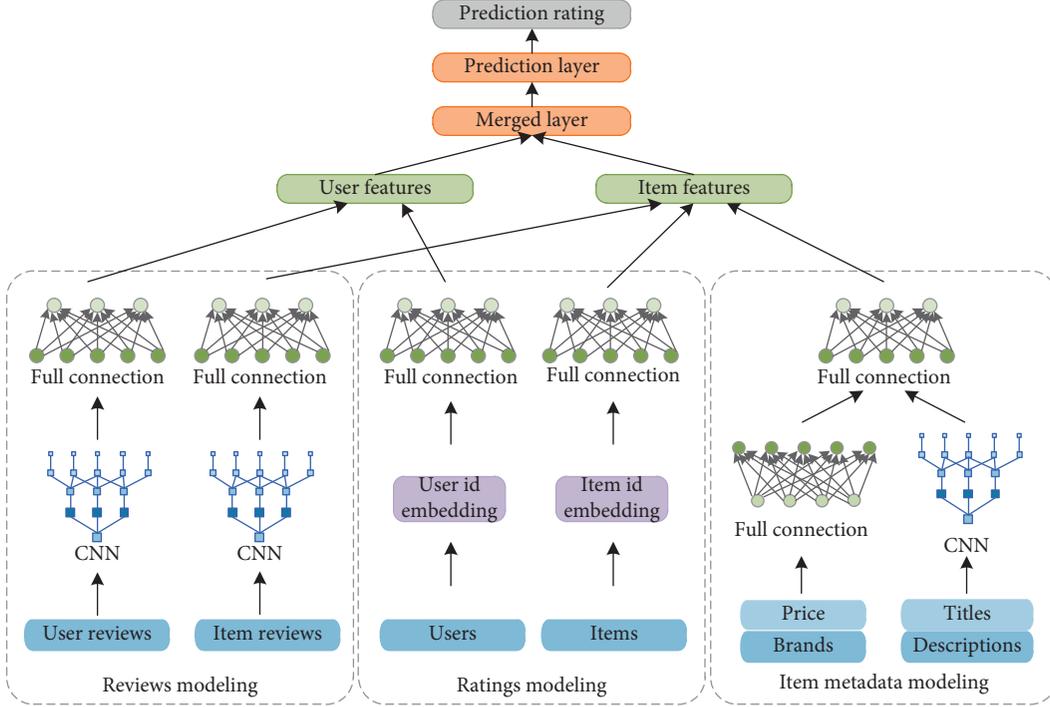


FIGURE 1: Architecture of DeepFusion.

factorization (MF) to acquire pair-dependent latent features on the basis of numerical ratings. Shown in the right side of Figure 1, the Item Metadata Modeling component introduces four deep neural networks to learn item complementary features based on item raw content, i.e., price, brand, title, and description. The outputs of three components are further mapped to a shared semantic space with a merged layer to obtain the integrated user preferences and item features. Then we develop an MLP architecture to capture the complex relations between users and items and predict ratings. Finally, the loss function is defined as the error between the predicted rating and the actual rating, and model parameters are adjusted via a backpropagation process. Key notations adopted in this paper are summarized in Table 1.

3.2. Reviews Modeling. To capture the underlying meanings of textual reviews, we use the Reviews Modeling component to improve the model’s coverage. This process is conducted with two similar CNNs [11]: one network for users and one network for items. In the following, we describe the user network in detail.

The first layer is the word embedding layer, which receives user reviews as the input in their original order and outputs a c -dimensional distributed vector. In this paper, user u reviews refer to all the reviews that were written by user u . As a widely used word representation in information retrieval, the bag-of-words model is based on one-hot representation [50] and is usually used to transform a word into a feature vector. However, the one-hot representation of a word, ignoring semantic and grammatical interpretation, tends to suffer from the curse of

dimensionality [51]. To alleviate this problem, we resort to a widely used natural language processing method Word2Vec [52] to convert the dictionary of words into formal and uniform vectors. The output of the word embedding layer can be represented as

$$V_{1:T}^u = [f(d_1^u), f(d_2^u), \dots, f(d_T^u)], \quad (1)$$

where d_t^u denotes the t -th word of user u review, T represents the length of user u review, $f(d_t^u)$ is a lookup function that returns the corresponding word vectors, and $[f_1, f_2, \dots, f_T]$ denotes the concatenation of word vectors.

The next layer is the convolution layer, which consists of m neurons. Each neuron is associated with a convolution kernel $K \in \mathbb{R}^{w \times c}$, where w is the window size. The convolution kernel can perform multiple convolution operations on word vectors $V_{1:T}^u$ and add bias to obtain the feature map. The features of neuron K_n can be defined as

$$z_n = f(V_{1:T} * K_n + b_n), \quad (2)$$

where symbol $*$ denotes the convolution operation, b_n is the bias, and f is the nonlinear activation function. According to [33], we employ the rectified linear unit (ReLU), which has been widely used in neural networks, as an activation function:

$$\text{ReLU}(x) = \max\{0, x\}. \quad (3)$$

Then, we use the maxpooling operation to select the maximum value as the significant feature for extraction. The pooling layer not only reduces the dimension of data, but also retains the representative features. Suppose that $z_1, z_2, z_3, \dots, z_{(T-w+1)}$ are the features of neuron K_n on the

TABLE 1: A summary of key notations.

Notation	Definition
u	A user
i	A item
c	The dimension of word embedding
T	The length of user u review
d_T^u	User u 's review text consisting of T words
$V_{1:T}^u$	Word vectors of user u
m	The number of neurons in the convolution layer
K_n	The n -th kernel in the convolution layer
w	The window size of convolution kernel
z_n	The n -th feature map in the convolutional layer
b_n	The bias of n -th convolutional kernel
o_n	The output of neuron K_n in the convolutional layer
O	The output of the pooling layer
W	The weight matrix of the fully connected layer
b	The bias of the fully connected layer
P	The dense vector of the ID embeddings of user or item
V_i	The dense vector of price range of item i
E_i	The price feature of item i
F_i	The brand feature of item i
H_i	The title feature of item i
D_i	The description feature of item i
G_i	The concatenate of vector E_i, F_i, H_i , and D_i
U^{Reviews}	The latent features of users in the context of reviews
U^{Ratings}	The latent features of users in the context of ratings
I^{Reviews}	The latent features of items in the context of reviews
I^{Ratings}	The latent features of items in the context of ratings
I^{metadata}	The latent features of items in the context of item metadata
U, I	The latent features of users and items, respectively
Z	The concatenate of vector U and I
$\hat{R}_{u,i}$	The predicted value of user u on item i
$R_{u,i}$	The ground-truth value of user u on item i
ρ	The dropout ratio
k	The number of latent factors
λ_u, λ_v	Regularization parameters of user and item, respectively
l_r	The learning rate
a	The batch size
p	A percentage of the length of a review
r	A percentage of the number of vocabulary

sliding window w , the convolutional result is reduced to a fixed-size vector:

$$o_n = \max\{z_1, z_2, z_3, \dots, z_{(T-w+1)}\}. \quad (4)$$

After many applications of similar operations, the final output of the pooling layer is the concatenation of various features of m neurons, and the output is

$$O = [o_1, o_2, o_3, \dots, o_m]. \quad (5)$$

As expressed in equation (6), the output O is fed to the fully connected layer, which comprises weight matrix $W \in \mathbb{R}^{m \times k}$ and a bias term $b \in \mathbb{R}^k$ and uses ReLU as the activation function. The outputs of the fully connected layer are user preferences or item features based on textual reviews and are denoted as U^{Reviews} and I^{Reviews} , respectively:

$$U^{\text{Reviews}} = \text{ReLU}(WO + b). \quad (6)$$

3.3. Ratings Modeling. Matrix factorization is one of the most popular model-based CF methods and can learn linear interactions between users and items. Inspired by [35], we develop a full neural treatment of MF for deriving pair-dependent latent representations on the basis of user-item rating pairs, which is named as the Ratings Modeling component.

As is illustrated in Figure 1, the inputs of the neural network are the identities of users. The embedding layer projects the sparse representation to a dense vector via a lookup function \emptyset . Let ID^1, ID^2, \dots, ID^u be the ID embeddings of users. Then, the corresponding dense vector P is defined as

$$P = [\emptyset(ID^1), \emptyset(ID^2), \dots, \emptyset(ID^u)]. \quad (7)$$

The next layer is a full connection layer whose outputs can be regarded as the latent vector for user U^{Ratings} in the context of numerical ratings. Here, we still use ReLU as the activation function:

$$U^{\text{Ratings}} = \text{ReLU}(WP + b). \quad (8)$$

The same process is adopted by the item network with corresponding layers, and we can acquire the latent vector for item I^{Ratings} in the context of numerical ratings.

3.4. Item Metadata Modeling. The above two components are built based on user-generated content and easily suffer from the sparsity problem and the unbalanced distribution problem. Therefore, we develop the Item Metadata Modeling component to model item raw content. Different types of item metadata describe different aspects of item features. Considering feasibility and availability, we use four types of product (item) metadata including price, brand, title, and description as supplemental information for item features.

We regard ‘‘price’’ and ‘‘brand’’ as structured indicators, which are typically easy to understand. Thus, the network of price is analogous with that of the brand and they differ only in terms of their inputs. In the first layer of the price network, an embedding function $\varphi: M \rightarrow \mathbb{R}^k$ maps the price range into a k -dimensional corresponding vector V_i . Then, we use a full connection layer to learn the representation of price feature E_i :

$$E_i = \text{ReLU}(WV_i + b). \quad (9)$$

Similarly, we use the brand category as the input of the brand network and acquire the brand feature representation F_i .

In addition, title and description texts typically reflect an overall profile of item and are structured in natural language. Therefore, we use two parallel convolutional neural networks to learn the title and description feature representations H_i and D_i , respectively. The CNNs are similar to the user network that is described in Section 3.2.

After that, to map the above four aspect features into a unified feature space, we concatenate them directly:

$$G_i = [E_i, F_i, H_i, D_i]. \quad (10)$$

To prevent high dimensionality of feature vectors and emphasize the most important features of item, we feed G_i into a full connection layer to obtain the item features in the context of the item metadata:

$$I^{\text{metadata}} = WG_i + b. \quad (11)$$

3.5. Multiview Fusion and Prediction. To effectively use multiview information in recommender systems, we fuse the above three components by merging the outputs of their last layers. The final representations for users consist of two data sources: user-item numerical ratings and user-item textual reviews. Except for ratings and reviews, the final representations for items comprise the item metadata:

$$U = U^{\text{Reviews}} + U^{\text{Ratings}}, \quad (12)$$

$$I = I^{\text{Reviews}} + I^{\text{Ratings}} + I^{\text{metadata}}. \quad (13)$$

Then, to better capture the complex relations between users and items, we utilize an MLP architecture [34], which is a widely used technique with excellent scalability. We concatenate U and I into a single vector $Z = [U, I]$ and feed Z into new hidden layers to predict ratings.

$$\hat{R}_{u,i} = \text{ReLU}(WZ + b). \quad (14)$$

The number of hidden layers can be customized to better model the latent structure of user-item interactions. In our model, we use two hidden layers with 32 and 16 hidden units.

3.6. Learning. According to [12, 53], we adopt the squared loss as the objective function, which is commonly used in rating prediction problems. Suppose $\hat{R}_{u,i}$ and $R_{u,i}$ represent the predicted value and the ground-truth value of user u on item i , the objective function can be defined as

$$L_r = \sum_{u,i \in \Gamma} (\hat{R}_{u,i} - R_{u,i})^2, \quad (15)$$

where Γ denotes the set of instances for training.

Then, we optimize the model via adaptive moment estimation (Adam) [54] and adjust the model parameters via a backpropagation process. By automatically selecting the learning rate, the deep learning model quickly converges. In addition, to avoid overfitting, we adopt the dropout [55] strategy. After obtaining the merging vector Z , we randomly drop ρ percent of the neurons and their connections, where ρ is the dropout ratio.

4. Experiments and Evaluation

4.1. Dataset. To evaluate the performance of our model in terms of rating prediction, we conduct extensive experiments on three real-world datasets that were collected from Amazon product data (<http://jmcauley.ucsd.edu/data/amazon>): Musical_Instruments, Automotive, and Sports_and_Outdoors. These datasets consist of users' explicit ratings on items on a scale of 1 to 5 and textual

reviews for various products. In summary, each rating record is a four-tuple (userID, itemID, rating, and reviews). Each item contains itemID, price, brand, title, and description. Items that do not have metadata information such as the price, title, or description documents are removed from the dataset. The users and the items that have less than 5 ratings are also removed. The detailed statistics are presented in Table 2.

We preprocess textual documents including reviews, titles, and descriptions according to [12]. Since the length of the text and the size of vocabulary exhibit a long-tail effect [9], we only keep p percent of the length of reviews, where p is set to 0.85, and keep r percent the size of vocabulary, where r is set to 0.7. We project the prices into specified price ranges with intervals of 10 and classify items for which the price exceeds 260 into the same class. Items that lack the brand attribute are classified as "other." And, we argue that users who purchase those items do not pay much attention to the brand attribute.

4.2. Evaluation Criteria. We adopt the root mean square error (RMSE) as the evaluation criterion, which is a standard evaluation metric for rating prediction in recommender systems [12]. Given a predicted rating $\hat{R}_{u,i}$ and a ground-truth rating $R_{u,i}$ from user u to item i , the RMSE score is computed by

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{u,i} (\hat{R}_{u,i} - R_{u,i})^2}. \quad (16)$$

We also used the mean absolute error (MAE) to evaluate our model, which has been widely used in previous studies [18]. The MAE score is calculated as

$$\text{MAE} = \frac{1}{N} \sum_{u,i} |\hat{R}_{u,i} - R_{u,i}|, \quad (17)$$

where N denotes the number of ratings between users and items. A lower RMSE (equation (13)) and a lower MAE (equation (14)) correspond to a better recommendation performance.

4.3. Baseline Methods. To evaluate the performance of our proposed DeepFusion method, we select four comparison algorithms and describe them briefly as follows:

- (i) PMF [18]: probabilistic matrix factorization based on ratings is a standard rating prediction method that initializes the latent factors for users and items from a Gaussian distribution.
- (ii) NeuMF [35]: neural matrix factorization based on ratings is a state-of-the-art model that fuses generalized matrix factorization (GMF) and MLP to jointly model the linear and nonlinear interactions between user preferences and item features.
- (iii) ConvMF [56]: convolutional matrix factorization is based on ratings and textual description documents. ConvMF is a strong baseline that integrates CNN into PMF to improve the rating prediction accuracy.

TABLE 2: Statistics of the dataset.

Class	Musical_Instruments	Automotive	Sports_and_Outdoors
Users	6186	17470	59982
Items	9573	46554	46554
Reviews	37110	106165	342229

- (iv) DeepCoNN [11]: the deep cooperative neural network method based on ratings and textual reviews utilizes two parallel neural networks to jointly learn user preferences and item features and enables these two latent factors to interact with each other in a manner similar to factorization machine. Since DeepCoNN was evaluated against strong topic models, such as collaborative topic regression (CTR) [57], hidden factors as topics (HFT) [15], and collaborative deep learning (CDL) [32], and demonstrated superior performance, we do not repeat those same comparisons in this paper.

4.4. Experiment Details. Each dataset is split randomly into a training set (80%), a validation set (10%), and a test set (10%). The training set contains at least one rating on every user and item and is used to train our model. The validation set is used to tune hyperparameters and early stop the training phase. The test set is used to conduct the final performance comparison.

According to the results of parameter tuning, we set the number of latent factors to $k=50$ and regularization parameters to $\lambda_u = 0.1$ and $\lambda_v = 0.1$ for PMF. For ConvMF, we leverage reviews on the items as item description documents and set the latent dimension of U and V to 50 and the regularization parameters to $\lambda_u = 100$ and $\lambda_v = 10$. For deep learning-based methods, NeuMF, DeepCoNN, and DeepFusion, we set the learning rate to $l_r = 0.001$, the batch size to $a = 128$, the dropout ratio to $\rho = 0.5$, and the number of latent factors to $k = 16$. For the CNNs in DeepFusion, we reuse most of the hyperparameter settings that were presented by the authors of DeepCoNN, and we set the number of neurons to $m = 100$ and the window size to $w = 3$. We use Word2Vec as the pretrained word embedding model. We initialize word latent vectors via pretrained 300-dimensional word embeddings, which were trained on more than 100 billion words from Google News [41].

4.5. Performance Evaluation. The performances of our proposed algorithm and all baselines on the three datasets are reported in Tables 3 and 4. The experiments are repeated 3 times, and the averages are reported with the best results indicated in bold.

Between the methods that consider ratings, NeuMF outperforms PMF on both evaluation criteria. The main limitation of PMF is that it learns the latent factors via a global optimization strategy and predicts an unknown rating by the dot product of the targeted user and item latent factors. As a result, the performance could be severely compromised locally for individual users or items. In contrast, NeuMF combines the linearity of MF and the

nonlinearity of deep neural networks to learn user and item latent features. The hybrid architecture accurately models the interactions between them and thus offers better performance.

In addition, the experimental results demonstrate that all review-based rating prediction methods (ConvMF, DeepCoNN, and DeepFusion) outperform PMF. This is because a rating only reflects the overall satisfaction or judgment of a user towards an item. Relying solely on ratings makes it hard for PMF to explicitly and accurately model user and item features. Textual reviews indicate user opinion and emotion towards items' various features. Therefore, these methods can provide such fine-grained analysis. Moreover, DeepCoNN and DeepFusion outperform ConvMF. ConvMF only utilizes a CNN to acquire item features from reviews, while user preferences are just learned from numerical ratings. As a result, ConvMF suffers from poor rating prediction.

Overall, our method DeepFusion outperforms all the baseline methods. The reasons are as follows. First, our approach jointly learns user preferences and item features from multiview data: numerical ratings, textual reviews, and item metadata, which support our hypothesis that incorporating user-generated content and item raw content will improve recommendation performance. Second, our model utilizes multiple types of deep neural networks that are best suited for each type of heterogeneous data sources, which is beneficial to tackle the sparsity problem and the unbalanced distribution problem. Third, our method adopts a multilayer perceptron architecture to learn the complex relationships between users and items and thus obtains a better rating prediction.

4.6. Model Analysis. In this section, we discuss the effects of hyperparameters and several design selections on our model's performance.

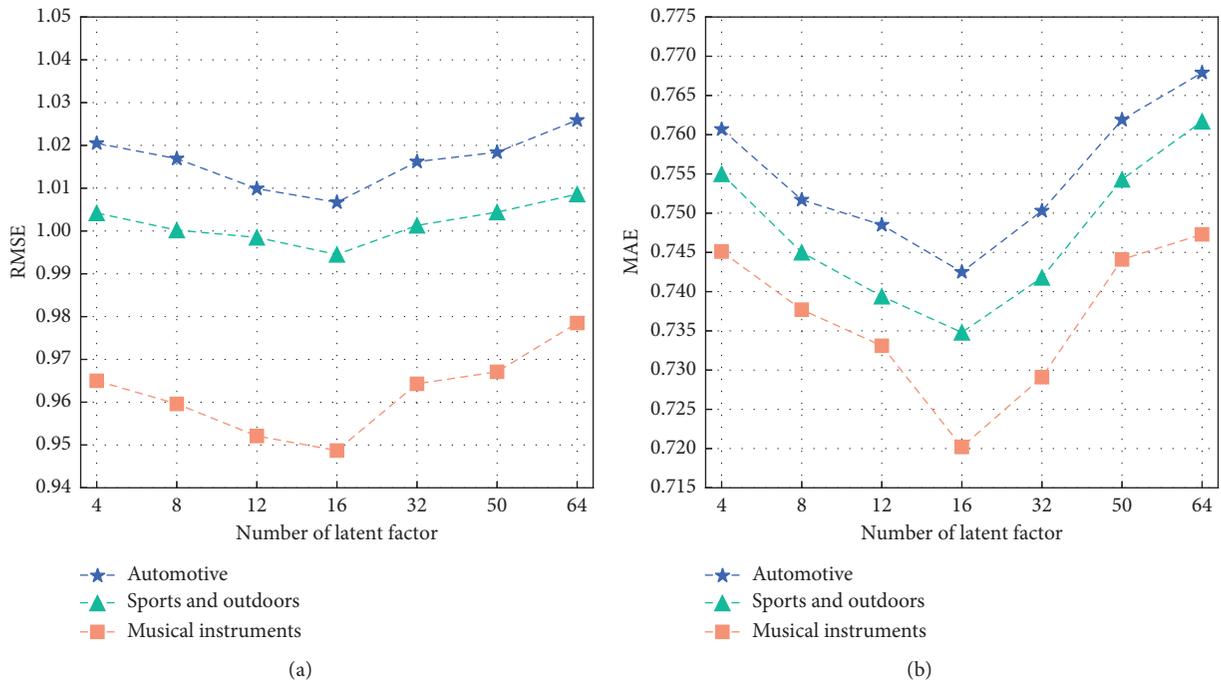
4.6.1. Impact of k . Our method encodes feature representations of users and items into two k -dimensional latent vectors, where k denotes the number of latent factors. To empirically study the effect of k , we compare the performances of DeepFusion on the validation set among the values of k in [4, 8, 12, 16, 32, 50, and 64]. Figure 2 plots the RMSE and MAE changes in our model as functions of k on three datasets. The increase of k from 4 to 16 promotes the performance, while the increase of k from 16 to 64 does not boost the performance, but rather causes degradation. Hence, a relatively low or high value of k may cause underfitting or overfitting. Therefore, a proper value of k can enhance the performance of the recommender systems, and $k = 16$ is the optimal setting for our experiment.

TABLE 3: RMSE comparison of five models on three datasets.

Model	Musical_Instruments	Automotive	Sports_and_Outdoors
PMF	1.0820	1.1362	1.1026
NeuMF	0.9952	1.0347	1.0049
ConvMF	1.0415	1.0785	1.0677
DeepCoNN	0.9998	1.0457	1.0105
DeepFusion	0.9757	1.0292	1.0037

TABLE 4: MAE comparison of five models on three datasets.

Model	Musical_Instruments	Automotive	Sports_and_Outdoors
PMF	0.8275	0.8704	0.8362
NeuMF	0.7563	0.7631	0.7594
ConvMF	0.7674	0.7861	0.7739
DeepCoNN	0.7367	0.7554	0.7570
DeepFusion	0.7275	0.7464	0.7363

FIGURE 2: Recommendation performance under various values of k . (a) RMSE. (b) MAE.

4.6.2. *Impact of the Merged Layer.* To further investigate the performance of the merged layer in capturing intricate relations between the users and items, we compare the DeepFusion with variants that are based on three types of latent feature interactions: DeepFusion-dp, DeepFusion-lmf, and DeepFusion-fm. These three variants are summarized as follows:

- (i) DeepFusion-dp: we use a simple dot product of the latent features of users and items as the rating predictor.
- (ii) DeepFusion-lmf: the merged layer enables the latent features of users and items to interact with each other, similarly to latent matrix factorization (LMF) [6].

- (iii) DeepFusion-fm: we utilize the factorization machine [37] in place of the original neural prediction layer.

As shown in Figure 3, DeepFusion-fm outperforms DeepFusion-dp and DeepFusion-lmf on three datasets. This is because the factorization machine models not only the first-order interactions, but also the second-order interactions between the representations of users and items. In addition, the factorization machine automatically selects helpful features and thus offers better performance. More importantly, our original method DeepFusion significantly outperforms all variant methods. The results demonstrate that the multilayer perceptron architecture effectively learns the hidden intricate relationships between users and items

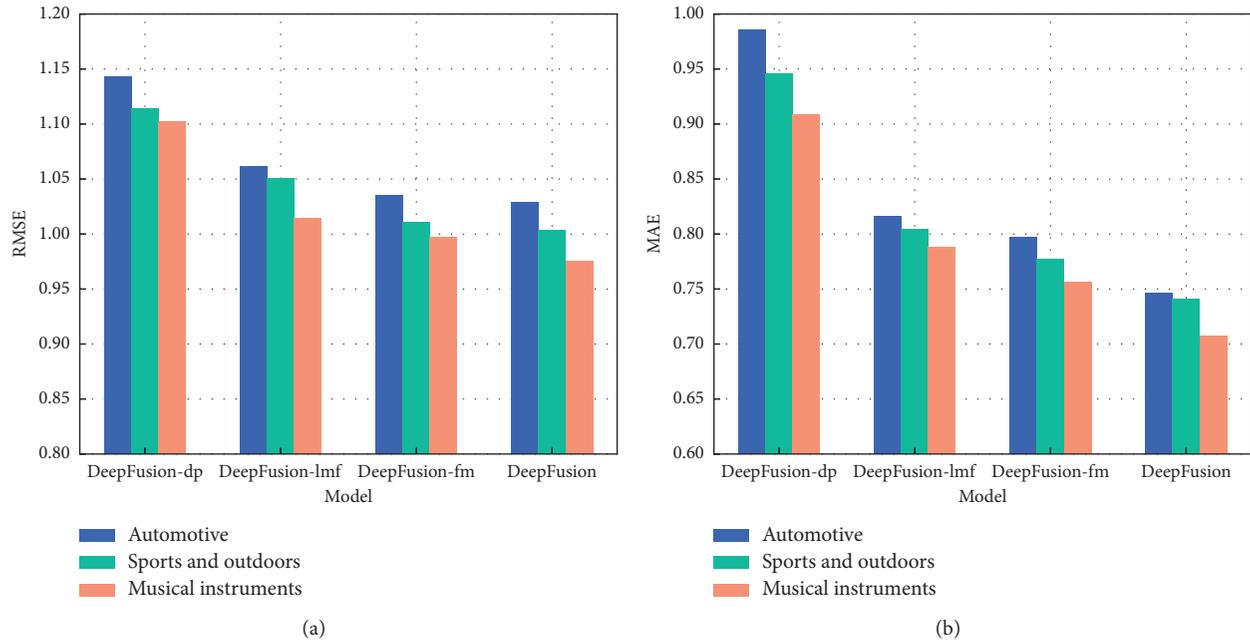


FIGURE 3: Performance comparison of merged layers. (a) RMSE. (b) MAE.

and models the nonlinear interactions between them. Therefore, the multilayer perceptron architecture was integrated into our final model.

4.6.3. Impact of Multiview Fusion. Does the multiview fusion boost the recommendation performance? To answer this question, we investigate the impacts of the three independent learning components: Reviews Modeling, Ratings Modeling, and Item Metadata Modeling on the performance of our proposed method. We compare DeepFusion with its three variants: DeepFusion-Reviews, DeepFusion-Ratings, and DeepFusion-Metadata. These three variants are summarized as follows:

- (i) DeepFusion-Reviews: the Reviews Modeling component is omitted, namely, this variant only consists of Ratings Modeling and Item Metadata Modeling.
- (ii) DeepFusion-Ratings: similar to DeepFusion-Reviews, the Rating Modeling component is excluded.
- (iii) DeepFusion-Metadata: the Item Metadata Modeling component is excluded.

Figure 4 plots the prediction accuracy of the three variants and DeepFusion in terms of RMSE values. First, DeepFusion outperforms DeepFusion-Reviews on the three datasets. This is because the semantic and syntactic information of textual reviews compensates for a shortage of ratings. Moreover, this result demonstrates that textual reviews cover rich information, which is beneficial to reveal user preferences and item features.

Second, DeepFusion-Ratings performs relatively weaker than DeepFusion. Hence, the application of the interaction-based learning process on the basis of user-item pairs is conducive to rating prediction. The pair-dependent latent

representations complement the independent reviews and item metadata learning approaches. Meanwhile, MF easily captures linear interactions between users and items and is thus effective in improving the prediction performance.

Finally, DeepFusion-Metadata performs the worst by far on all three datasets. The results show that the item metadata information effectively reflects the comprehensive characteristics of items. Since user-item interaction data are extremely sparse in e-commercial recommender systems, approaches only using user-generated content easily suffer from poor performance. By incorporating item metadata, richer features can be acquired for items that had few ratings and reviews, which is conducive to tackling the sparsity problem and the unbalanced distribution problem. Since the three components complement with each other, our proposed model DeepFusion can reap the benefits of additional data sources and realize superior rating prediction accuracy.

Furthermore, to further explore the relative importance of various product (item) metadata in promoting the rating prediction performance, we conduct a set of experiments using DeepFusion-Price, DeepFusion-Brand, DeepFusion-Title, and DeepFusion-Des. These variants are summarized as follows:

- (i) DeepFusion-Price: the price attribute is neglected.
- (ii) DeepFusion-Brand: the brand attribute is neglected.
- (iii) DeepFusion-Title: the title texts are neglected.
- (iv) DeepFusion-Des: the description texts are neglected.

Figure 5 illustrates the reductions in RMSE that resulted from DeepFusion compared to DeepFusion-Price, DeepFusion-Brand, DeepFusion-Title, and DeepFusion-Des on the three datasets. In terms of RMSE, DeepFusion-Brand

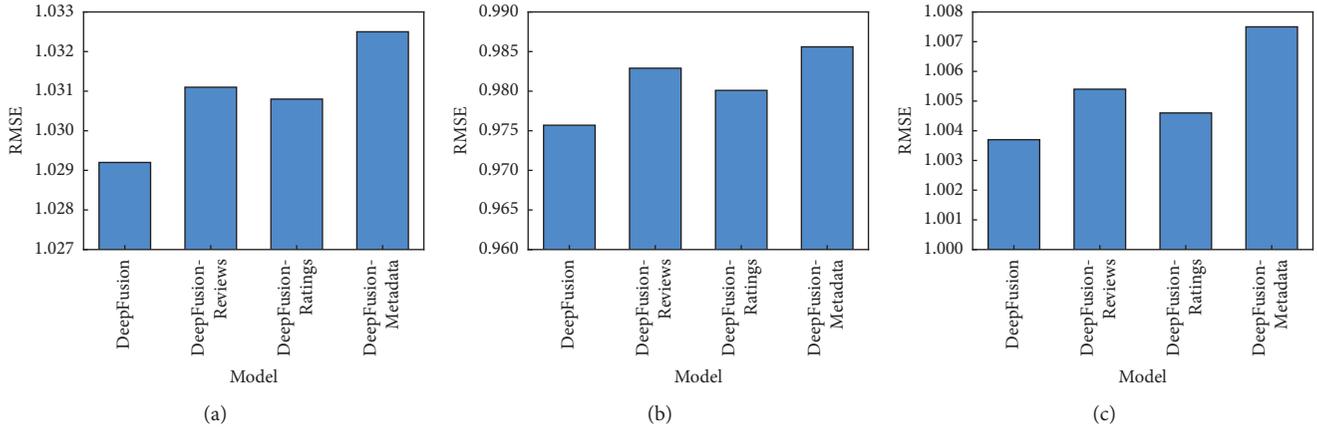


FIGURE 4: Performance comparison of the three components. (a) Automotive. (b) Musical_Instruments. (c) Sports_and_Outdoors.

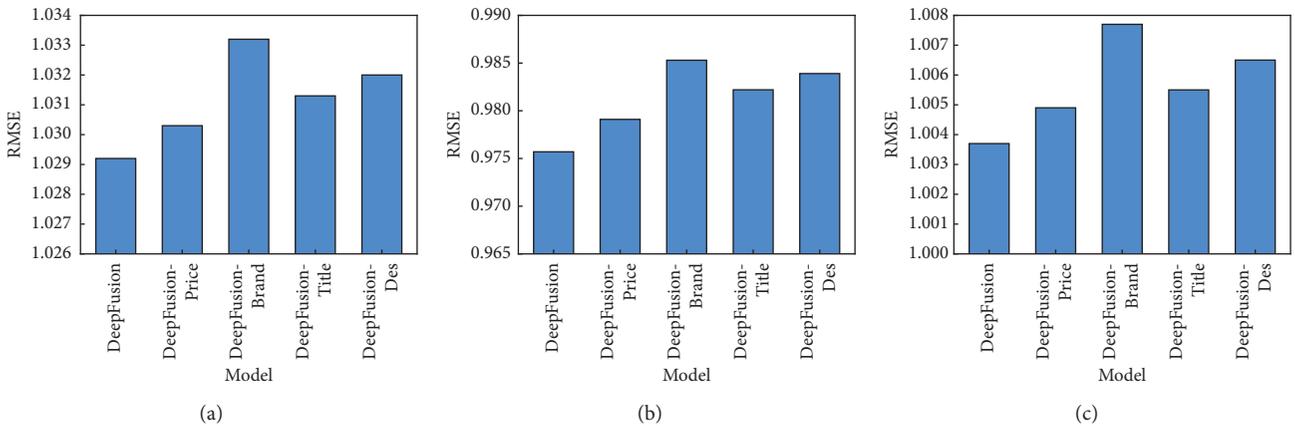


FIGURE 5: Performance comparison of four item metadata. (a) Automotive. (b) Musical_Instruments. (c) Sports_and_Outdoors.

performs the worst among these variants. Hence, the item brand attribute plays the most important role in product recommendation. Furthermore, this phenomenon demonstrates that customers pay more attention to the brand attribute of products; therefore, businesses should make efforts to enhance brand images. In addition, although DeepFusion-Des and DeepFusion-Title slightly outperform DeepFusion-Brand, they are both outperformed by DeepFusion. The results demonstrate that incorporating textual titles and textual descriptions into our model promotes the rating prediction accuracy. Textual titles and textual descriptions typically reflect an overall profile of an item and facilitate to acquire the item features. Finally, DeepFusion-Price outperforms other variants but is outperformed by DeepFusion; hence, the price attribute slightly contributes to boosting the recommendation accuracy. In summary, the relative importance of product (item) metadata is different on user buying behaviors in e-commerce. The brand attribute plays the most important role in product recommender systems, followed by textual descriptions and textual titles, and the price attribute has a relatively slight effect on the rating prediction performance.

5. Conclusions

In this paper, we proposed a novel personalized recommendation method based on deep neural networks and multiview fusion, called DeepFusion, for rating prediction task in product recommendation. The model is capable of incorporating user-generated content and item raw content including numerical ratings, textual reviews, and item metadata in a unified space. Meanwhile, we designed three neural network components to jointly learn user and item representations. Finally, a multilayer perceptron layer was employed to capture the complex relations between users and items. Evaluated over the Amazon product dataset, our proposed DeepFusion model achieved better performance than all the baselines. In addition, we further explored the relative importance of product (item) metadata on user buying behaviors on e-commerce websites.

In the future, we intend to evaluate our proposed method over additional datasets. In addition, we will consider incorporating more heterogeneous data sources such as item images and user social networks towards a unified, multiview informed practical recommender system.

Data Availability

The Amazon product data supporting the findings of 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 partly supported by the National Science Foundation of China (nos. 71871019 and 71471016) and by the Fundamental Research Funds for the Central Universities under Grant no. FRF-TP-18-013B1.

References

- [1] A. Edmunds and A. Morris, "The problem of information overload in business organisations: a review of the literature," *International Journal of Information Management*, vol. 20, no. 1, pp. 17–28, 2000.
- [2] Y. Yu, L. Zhang, C. Wang, R. Gao, W. Zhao, and J. Jiang, "Neural personalized ranking via poisson factor model for item recommendation," *Complexity*, vol. 2019, Article ID 3563674, 16 pages, 2019.
- [3] O. Georgiou and N. Tsapatsoulis, "The importance of similarity metrics for representative users identification in recommender systems," in *Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations*, Springer, Larnaca, Cyprus, pp. 12–21, October 2010.
- [4] S. Wei, N. Ye, S. Zhang, X. Huang, and J. Zhu, "Item-based collaborative filtering recommendation algorithm combining item category with interestingness measure," in *Proceedings of the 2012 International Conference on Computer Science and Service System*, IEEE, Nanjing, China, pp. 2038–2041, August 2012.
- [5] D. Agarwal and B. C. Chen, "Regression-based latent factor models," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 19–28, Paris, France, June 2009.
- [6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
- [7] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, pp. 1–35, Springer, Boston, MA, USA, 2011.
- [8] C. A. Gomez-Urbe and N. Hunt, "The netflix recommender system: algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, p. 13, 2016.
- [9] C. Anderson, *The Long Tail: Why the Future of Business Is Selling Less of More*, Hachette Books, New York, NY, USA, 2006.
- [10] 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*, vol. 37, no. 2, pp. 1–29, 2019.
- [11] 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*, ACM, Cambridge, UK, pp. 425–434, February 2017.
- [12] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proceedings of the 2018 World Wide Web Conference International World Wide Web Conferences Steering Committee*, pp. 1583–1592, Lyon, France, April 2018.
- [13] X. Ma, X. Lei, G. Zhao, and X. Qian, "Rating prediction by exploring user's preference and sentiment," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 6425–6444, 2018.
- [14] M. Alfarhood and J. Cheng, "DeepHCF: a deep learning based hybrid collaborative filtering approach for recommendation systems," in *Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, Orlando, FL, USA, pp. 89–96, December 2018.
- [15] 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*, ACM, Hong Kong, China, pp. 165–172, October 2013.
- [16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
- [17] J. Chambua, Z. Niu, A. Yousif, and J. Mbelwa, "Tensor factorization method based on review text semantic similarity for rating prediction," *Expert Systems with Applications*, vol. 114, pp. 629–638, 2018.
- [18] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in Neural Information Processing Systems*, pp. 1257–1264, MIT Press, Cambridge, MA, USA, 2008.
- [19] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, "Aspect-aware latent factor model: rating prediction with ratings and reviews," in *Proceedings of the 2018 World Wide Web Conference*, pp. 639–648, Lyon France, April 2018.
- [20] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An unsupervised neural attention model for aspect extraction," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 388–397, Vancouver, Canada, July 2017.
- [21] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. Kankanhalli, "A³NCF: an adaptive aspect attention model for rating prediction," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3748–3754, Stockholm, Sweden, July 2018.
- [22] T. Ma, X. Suo, J. Zhou et al., "Augmenting matrix factorization technique with the combination of tags and genres," *Physica A: Statistical Mechanics and Its Applications*, vol. 461, pp. 101–116, 2016.
- [23] 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.
- [24] Z. Cheng, X. Chang, L. Zhu, R. C. Kanjirathinkal, and M. Kankanhalli, "MMALFM," *ACM Transactions on Information Systems*, vol. 37, no. 2, pp. 1–28, 2019.
- [25] S. Huang, J. Ma, P. Cheng, and S. Wang, "A hybrid multi-group coclustering recommendation framework based on information fusion," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 2, pp. 1–22, 2015.
- [26] Y. Qian, Y. Zhang, X. Ma, H. Yu, and L. Peng, "EARS: emotion-aware recommender system based on hybrid information fusion," *Information Fusion*, vol. 46, pp. 141–146, 2019.
- [27] E. Zheng, G. Y. Kondo, S. Zilora, and Q. Yu, "Tag-aware dynamic music recommendation," *Expert Systems with Applications*, vol. 106, pp. 244–251, 2018.
- [28] K. Bougiatiotis and T. Giannakopoulos, "Enhanced movie content similarity based on textual, auditory and visual

- information,” *Expert Systems with Applications*, vol. 96, pp. 86–102, 2018.
- [29] S. Li, J. Kawale, and Y. Fu, “Deep collaborative filtering via marginalized denoising auto-encoder,” in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, ACM, Melbourne, Australia, pp. 811–820, October 2015.
- [30] X. Wang and Y. Wang, “Improving content-based and hybrid music recommendation using deep learning,” in *Proceedings of the 22nd ACM International Conference on Multimedia*, ACM, Orlando, Florida, USA, pp. 627–636, November 2014.
- [31] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, “Collaborative denoising auto-encoders for top-n recommender systems,” in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, ACM, San Francisco, CA, USA, pp. 153–162, February 2016.
- [32] 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, Sydney, Australia, August 2015.
- [33] S. Yu, M. Yang, Q. Qu, and Y. Shen, “Contextual-boosted deep neural collaborative filtering model for interpretable recommendation,” *Expert Systems with Applications*, vol. 136, pp. 365–375, 2019.
- [34] G. Zhou, X. Zhu, C. Song et al., “Deep interest network for click-through rate prediction,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, London, UK, pp. 1059–1068, July 2018.
- [35] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, “Neural collaborative filtering,” in *Proceedings of the 26th International Conference on World Wide Web*, International World Wide Web Conferences Steering Committee, Perth, Australia, pp. 173–182, April 2017.
- [36] X. He and T. S. Chua, “Neural factorization machines for sparse predictive analytics,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Tokyo, Japan, pp. 355–364, August 2017.
- [37] S. Rendle, “Factorization machines with libfm,” *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 3, p. 57, 2012.
- [38] H. T. Cheng, L. Koc, J. Harmsen et al., “Wide & deep learning for recommender systems,” in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, ACM, Boston, MA, USA, pp. 7–10, September 2016.
- [39] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, “DeepFM: a factorization-machine based neural network for CTR prediction,” 2017, <https://arxiv.org/abs/1703.04247>.
- [40] J. Chambua, Z. Niu, and Y. Zhu, “User preferences prediction approach based on embedded deep summaries,” *Expert Systems with Applications*, vol. 132, pp. 87–98, 2019.
- [41] G. Zhou, N. Mou, Y. Fan et al., “Deep interest evolution network for click-through rate prediction,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5941–5948, Honolulu, HI, USA, July 2019.
- [42] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T. S. Chua, “Attentive collaborative filtering: multimedia recommendation with item-and component-level attention,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–344, Tokyo, Japan, August 2017.
- [43] Y. Zhang, Q. Ai, X. Chen, and W. B. Croft, “Joint representation learning for top-n recommendation with heterogeneous information sources,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1449–1458, Singapore, November 2017.
- [44] M. Gan, Y. Ma, and K. Xiao, “CDMF: a deep learning model based on convolutional and dense-layer matrix factorization for context-aware recommendation,” in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, Maui, HI, USA, January 2019.
- [45] A. M. Elkahky, Y. Song, and X. He, “A multi-view deep learning approach for cross domain user modeling in recommendation systems,” in *Proceedings of the 24th International Conference on World Wide Web*, International World Wide Web Conferences Steering Committee, Florence, Italy, pp. 278–288, May 2015.
- [46] O. Tal and Y. Liu, “A joint deep recommendation framework for location-based social networks,” *Complexity*, vol. 2019, Article ID 2926749, 11 pages, 2019.
- [47] D. Guo, J. Hong, B. Luo, Q. Yan, and Z. Niu, “Multi-modal representation learning for short video understanding and recommendation,” in *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, Shanghai, China, pp. 687–690, July 2019.
- [48] X. Chen, H. Chen, H. Xu et al., “Personalized fashion recommendation with visual explanations based on multimodal attention network: towards visually explainable recommendation,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 765–774, Paris, France, July 2019.
- [49] Q. Cui, S. Wu, Q. Liu, and L. Wang, “A visual and textual recurrent neural network for sequential prediction,” 2016, <https://arxiv.org/abs/1703.04247>.
- [50] Z. Liu, M. Sun, Y. Lin, and R. Xie, “Knowledge representation learning: a review,” *Journal of Computer Research and Development*, vol. 53, no. 2, pp. 247–261, 2016, in Chinese.
- [51] J. Turian, L. Ratinov, and Y. Bengio, “Word representations: a simple and general method for semi-supervised learning,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Hohhot, China, pp. 384–394, July 2010.
- [52] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems*, pp. 3111–3119, MIT Press, Cambridge, MA, USA, 2013.
- [53] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T. S. Chua, “Attentional factorization machines: learning the weight of feature interactions via attention networks,” 2017, <https://arxiv.org/abs/1708.04617>.
- [54] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” 2014, <https://arxiv.org/abs/1412.6980>.
- [55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [56] 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*, ACM, Boston, MA, USA, pp. 233–240, September 2016.
- [57] 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*, ACM, San Diego CA, USA, pp. 448–456, August 2011.