Wang, Z., Chen, H., Li, Z., Lin, K., Jiang, N., & Xia, F. (2022). VRConvMF: Visual Recurrent Convolutional Matrix Factorization for Movie Recommendation. *IEEE Transactions on Emerging Topics in Computational Intelligence, 6*(3), 519–529.

Available online: https://doi.org/10.1109/TETCI.2021.3102619

VRConvMF: Visual Recurrent Convolutional Matrix Factorization for Movie Recommendation

Zhu Wang*, Honglong Chen*, Zhe Li*, Kai Lin*, Nan Jiang§, and Feng Xia¶
*College of Information and Control Engineering, China University of Petroleum, Qingdao, China
§College of Information Engineering, East China Jiaotong University, Nanchang, China
¶School of Engineering, IT and Physical Sciences, Federation University Australia, Ballarat, Australia

***Abstract*—Sparsity of user-to-item rating data becomes one of challenging issues in the recommender systems, which severely deteriorates the recommendation performance. Fortunately, context-aware recommender systems can alleviate the sparsity problem by making use of some auxiliary information, such as the information of both the users and items. In particular, the visual information of items, such as the movie poster, can be considered as the supplement for item description documents, which helps to obtain more item features. In this paper, we focus on movie recommender system and propose a probabilistic matrix factorization based recommendation scheme called *visual recurrent convolutional matrix factorization* (VRConvMF), which utilizes the textual and multi-level visual features extracted from the descriptive texts and posters respectively. We implement the proposed VRConvMF and conduct extensive experiments on three commonly used real world datasets to validate its effectiveness. The experimental results illustrate that the proposed VRConvMF outperforms the existing schemes.**

***Index Terms*—Recommender system, matrix factorization, visual feature, recurrent convolutional neural network.**

I. Introduction

The rapid development of the Internet of Things (IoTs) [1], [2] leads to the explosive growth of the number of items and users. Therefore, the sparseness of the user-to-item rating matrix in e-commerce becomes more and more serious, resulting in deteriorating the rating prediction precision of conventional collaborative filtering algorithms [3]. For instance, in the commonly used MovieLens-10m dataset, the average number of movies rated per user is about 142, which is much smaller than 10681, the total number of movies. The rating matrix in this situation is extremely sparse. Thus, the mere known ratings are far from enough in predicting the unobserved ratings.

To improve the rating prediction accuracy, the researchers considered both the explicit feedback (such as ratings) and implicit feedbacks (such as some observable behaviors of users, i.e., duration and repetition) [4] [5] in the recommender system [6]–[8]. Some auxiliary information such as users' reviews [9] and the category of items [10] have also been considered as implicit feedback to improve the recommendation performance. Moreover, Wang et al. [11] proposed to combine the probabilistic topic modeling and conventional collaborative filtering to provide an interpretable latent structure between the users and items. The deep learning algorithm has become another direction for the recommender system, since it can be used to obtain the deep representation of auxiliary information and improve the rating prediction accuracy [12] [13] [14]. Wang et al. [15] proposed a deep learning recommendation model, which adopted the stacked denoising auto-encoder (SDAE) based on probabilistic matrix factorization (PMF) [16] to obtain the accurate deep representations. Furthermore, in order to obtain comprehensive features, some research works have utilized the deep learning algorithm in the fields of Natural Language Processing (NLP) [17] [18] and Computer Vision (CV) [19] in the recommender systems. Especially, as for learning word embedding in the NLP field, Lai et al. [20] proposed the recurent convolutional neural network (RCNN), which can capture the entire context information.

Recently, Kim et al. [21] proposed a convolutional matrix factorization model (ConvMF), in which the convolutional neural network (CNN) and PMF are integrated to extract the contextual information features of documents and solve the problem that CNN cannot be directly applied to recommendation. Moreover, Chen et al. [22] proposed the deformable convolutional matrix factorization (DCNMF), which can capture more contextual information by adding offset layers in convolution layers to further improve the rating prediction accuracy. However, the aforementioned context-aware recommender systems can achieve limited performance improvement, since they just considered the textual information. Actually, the visual information is essential and primary in recognizing the world for the human beings. And the rich visual information of an item can well assist a user to determine whether he is interested in this item or not. For example, a vivid movie poster can help a user to judge whether he likes it or not [19], since it is difficult to obtain an intuitive and deep impression by just reading the description of the movie.

In this paper, we propose a PMF based movie recommendation scheme called *visual recurrent convolutional matrix factorization* (VRConvMF), which utilizes both the textual and multi-level visual features extracted from the descriptive texts and posters using RCNN and CNN respectively. Moreover, the confidence mechanism considering the user's preference is adopted to optimize the loss function and improve the rating prediction accuracy. To validate the effectiveness of our proposed model, we investigate the impacts of different single-level visual features on the recommendation performance. We

This work was supported by NSFC under Grant 61772551, Grant 62111530052 and the Major Scientific and Technological Projects of CNPC under Grant ZD2019-183-003.

Corresponding author: Honglong Chen. Email: chenhl@upc.edu.cn.

validate the performance of the proposed VRConvMF on three different real-world datasets. The experiment results illustrate that the proposed VRConvMF significantly outperforms the existing models.

The main contributions of this paper are as follows:

- We propose the visual recurrent convolutional matrix factorization based recommender scheme called VRConvMF, which can extract the textual features and multi-level visual features to alleviate the sparsity problem of user-to-item rating data in the movie recommender system.
- We adopt the confidence mechanism in the loss function to improve the rating prediction accuracy.
- We implement the proposed VRConvMF model and conduct extensive experiments on three real-world datasets to validate its effectiveness.

The rest of this paper is organized as follows. Section II discusses related works of recommendation. Section III introduces matrix factorization method and briefly describes word embedding representation, convolutional neural network for text and confidence mechanism. Section IV introduces the proposed movie recommender system VRConvMF in details. Section V conducts the performance evaluation. Section VI summarizes this paper and outlines the future work.

II. Related Work

In this section, we briefly discuss the collaborative filtering algorithm and the context-aware recommendations based on neural networks.

A. Collaborative Filtering

The collaborative filtering (CF) algorithm is to obtain the required recommendations by collaborative processing of the ratings given by a large number of users. Specifically, matrix factorization is one of the most commonly used methods in CF, which will be discussed in details in III-A. Since the observed ratings in user-to-item rating matrix are closely related to users' preference for items, the basic idea of CF is to estimate missing ratings from observed ratings. Furthermore, CF can be roughly divided into two types, one is the user-based CF (UserCF), and the other is the item-based CF (ItemCF). Specifically, UserCF makes use of user ratings, which are similar to target user i's preference, to predict user i's recommendations. The similarity function is used to find similar users by calculating each line of the rating matrix. While ItemCF is used to calculate the item set S, which is similar to item j. Then the ItemCF can predict whether user i appreciates item j or not based on user i's ratings to other items in S. It is easy to implement the traditional CF and the recommendations generated by CF are highly interpretable. However, CF is not suitable for the sparse rating matrix since it is difficult to find similar users or items. Therefore, alleviating the sparsity of the rating matrix is significant to improve the performance of collaborative filtering based recommendation.

B. Context-aware Recommendations

Recently, deep learning methods based on neural networks (such as recurrent neural networks (RNN), CNN and their derivative networks) have been widely applied to recommender systems [21] [22] [23]. Specifically, RNN are often used to process textual information, while CNN are more commonly used to process image information. Recommender systems can obtain the deep-level representation of various data types by using deep learning methods [24]. These deep-level representations are able to obtain the essential characteristics of items and users [25].

Context-aware recommendations utilize the contextual information to improve the rating prediction accuracy of recommender systems. Generally speaking, contextual information is extracted from attributes of items and user comments. A lot of additional information (such as time stamps [26], locations [27] and social networks [28]) can be used as contextual information for recommender systems. These contextual information is interdependent or independent. Many works considered contextual information such as social networks [29], user comment texts [30] and description of items [15] to improve the recommendation performance so far. Accordingly, as for recommender systems with specific tasks, the contextual information is different. For instance, music recommendation utilized some specific information, such as acoustic features to improve the performance of recommendation with sequence of songs [31] [32]. Furthermore, contextual information such as singer information, date of publish and the style of the song are applied in music recommendation to solve the cold-start problem [33]. As for movie recommendation, contextual information contains a wealth of attributes such as actor, director, genre, poster, trailer and descriptive text. Zhao et al. [19] presented movie recommendation model using visual features extracted from picture data. Moreover, Fan et al. [34] extracted visual features from trailers [35] directly using Youtube-8M dataset [36]. Compared to all of the above forms of contexts, there are relatively few of works considering the fusion of multiple features (such as textual features and visual features) to recommender systems.

III. Preliminary

A. Matrix Factorization

Matrix factorization (MF) is one of the collaborative filtering algorithms in the application of recommendation [3] [37]. The purpose of MF is to predict missing ratings by using observed ratings. In the traditional MF, it is supposed that there are m users, n items and a user-to-item rating matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$. Each row represents a user and each column represents an item. Each element r_{ij} in $\mathbf{R}$ represents the rating of user i ($1 \leq i \leq m$) on movie j ($1 \leq j \leq n$). Typically, ratings are divided into several levels (i.e., from 1 to 5), where higher values mean stronger preference. If user i did not rate movie j, let $r_{ij} = 0$. The $k-dimensional$ latent models of users and latent models of items are represented as the user-factor matrix $\mathbf{U} \in \mathbb{R}^{k \times m}$ and the item-factor matrix $\mathbf{V} \in \mathbb{R}^{k \times n}$ respectively. Accordingly, the i^{th} user and j^{th} item are denoted as u_i and v_j respectively. Each rating r_{ij} of user i on item j can be approximated by inner product of corresponding latent models of user i and item j: $r_{ij} \approx \widehat{r_{ij}} = u_i^T v_j$. Generally, training latent models tends to minimize the sum of squared errors between the actual rating r_{ij} and the predictive rating $\widehat{r_{ij}}$, and L_2 regularization term is applied to avoid the over-fitting problem. The loss function $\mathscr{L}$ can be expressed as:

$$\mathscr{L} = \sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}\Big(r_{ij} - u_i^T v_j\Big)^2 + \lambda_u \sum_{i=1}^{m}\Big\|u_i\Big\|^2 + \lambda_v \sum_{j=1}^{n}\Big\|v_j\Big\|^2, \quad (1)$$

where I_{ij} is the index function which is shown as follows.

$$I_{ij} = \begin{cases} 1, & \text{if user } i \text{ rated item } j, \\ 0, & \text{elsewhere.} \end{cases}$$

B. Word Embedding Representation

Word embedding is the first step in deep learning methods for natural language processing. Global-Vector (GloVe) [18] is one of the models that can convert natural language into machine language. GloVe model can learn textual information according to the full text, and can obtain word embedding vectors, which are represented by dense numeric matrices. The obtained word embedding vectors are able to capture the semantic properties between different words (such as similarity and analogy). The semantic similarity between two words can be measured via calculation based on the word embedding vectors (such as Euclidean distance or cosine similarity). GloVe model can be approximately divided into three parts. The first part is to build a co-occurrence matrix **Co** according to the corpus, each element co_{ij} in **Co** represents the number of times that word i and its context word j coexist in a context window with a specific size. Instead of using natural numbers, the decreasing weighting function $decay = 1/d$ is applied to count the number of word co-occurrences in GloVe model. Specifically, d denotes the distance between the context windows of two words. Obviously, in the total count, the further the distance between two words is, the smaller the corresponding value of $decay$ will be. The second part is to build the approximate relationship of word embedding vectors and co-occurrence matrix as Eq. (2):

$$w_i^T \widetilde{w}_j + b_i + \widetilde{b}_j \approx log\Big(co_{ij}\Big), \quad (2)$$

where w_i^T and $\widetilde{w}_j$ represent the word embedding vectors, b_i and $\widetilde{b}_j$ denote the bias terms of the word embedding vectors w_i^T and $\widetilde{w}_j$ respectively. While the third part is to build the loss function as Eq. (3):

$$J = \sum_{i,j=1}^{V} f(co_{ij})\Big(w_i^T \widetilde{w}_j + b_i + \widetilde{b}_j - log(co_{ij})\Big)^2, \quad (3)$$

where,

$$f(x) = \begin{cases} \Big(x/x_{max}\Big)^{\beta}, & \text{if } x < x_{max}, \\ 1, & \text{elsewhere.} \end{cases}$$

In the GloVe model, the adagrad method is utilized to optimize the parameters in loss function J to obtain the word vector w that is closest to the non-zero elements in the co-occurrence matrix **Co**. The application of the co-occurrence matrix can easily obtain the full-text contextual information by changing different context window sizes.

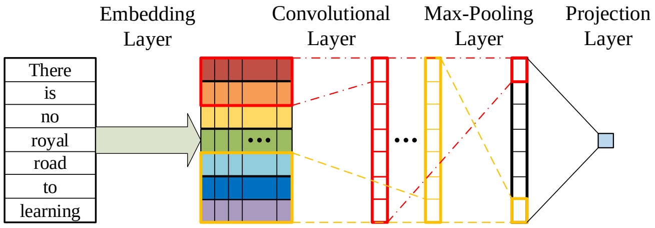


Fig. 1. Convolutional neural network for text.

C. Convolutional Neural Network for Text

1) Embedding Layer: The embedding layer is the first step to input natural language into mathematical models. We refer to each text as a sequence of words and transform all the sequences of words into a dense numeric matrix. In detail, all of the words are converted to vectors by pre-trained word embedding model (such as *Word2Vec* [38] and *GloVe*), then cascade all of the word vectors to represent textual documents mathematically. The descriptive texts or users' reviews of movie j are expressed as:

$$X_j = \Big[...w_{i-1} \oplus w_i \oplus w_{i+1}...\Big]^T_{p \times q}, \quad (4)$$

where p is the dimension of each word embedding vector, q is the length of descriptive texts of movies, and $\oplus$ represents cascade operation. Specifically, we use attributes of movies for MovieLens dataset and utilize users' reviews as descriptive texts for AIV dataset.

2) Convolutional Layer: A conventional convolution operation in sentence classification involves filters with the j^{th} shared weight $W_j = \mathbb{R}^{h \times p}$, where h is the convolutional window size. Specifically, a CNN with a small convolutional window size results in losing some far distance contextual information, while a large convolutional window size can capture more contextual information, however, this situation will lead to data sparsity. A textual feature $c_j = g(W_j \bullet w_{i:i+h-1} + b_j)$ is obtained by using convolution operation, where $\bullet$ represents convolution operator, i represents the i^{th} word embedding, b_j is the bias term for W_j, and $g(\cdot)$ is a non-linear activation function such as sigmoid, hyperbolic tangent and rectified linear unit ReLU. Then, all of the filters with different window sizes are applied to traverse each word vector of the entire text $\Big\{w_{1:h}, w_{2:h+1}, ..., w_{q-h+1:q}\Big\}$. A contextual feature vector of a textual document is given by:

$$\mathbf{C}_t = \Big[c_1, c_2, ..., c_{q-h+1}\Big], \quad (5)$$

where $\mathbf{C}_t$ refers to the textual feature of the t^{th} document extracted by convolution operator. Moreover, the multiple shared weights are used to capture the multiple textual features.

3) Pooling Layer: The pooling layer is applied to further extract the textual features of the convolutional layer in Eq. (5).

Max pooling operation is applied over the feature map to reduce the dimension of the textual feature vector to a fixed length n_c as follows:

$$\mathbf{D} = \Big[max(C_1), max(C_2), ..., max(C_{n_c})\Big], \tag{6}$$

4) Projection Layer: $\mathbf{D}$ is projected in Eq. (6) on a $k-$ *dimensional* space of item latent models for VConvMF model. The final textual feature is obtained by using conventional nonlinear projection as follows:

$$\mathbf{\Phi} = tanh\Big\{W_{fc2}\Big[tanh(W_{fc1}\mathbf{D} + b_{fc1})\Big] + b_{fc2}\Big\}, \tag{7}$$

where $\mathbf{W}_{fc1} \in \mathbb{R}^{x \times n_c}$ and $\mathbf{W}_{fc2} \in \mathbb{R}^{k \times x}$ are the weights of fully connected layers, and x is the number of hidden layer nodes for fully connected layer. The framework of convolutional neural network for text is shown in Fig. 1.

D. Confidence Mechanism

In this section, the confidence levels of expressing preferences between different users' ratings are considered. Specifically, extreme ratings (such as 1 and 5) are more reliable than moderate ratings (such as 2, 3 and 4). Therefore, the confidence value c_{ij} of each rating is adopted to revise the loss function. The confidence mechanism is used to assign higher values for more effective ratings. The specific confidence factor expression is as follows:

$$c_{ij} = 1 + \alpha \cdot f\Big(r_{ij} - \frac{r_{max}}{2}\Big), \tag{8}$$

where α is a hyper-parameter to control the value of the confidence factor c_{ij}, $f(\cdot)$ is a distance function (such as the absolute function or square function), and r_{max} is the maximum of ratings. Obviously, $r_{max} = 5$.

IV. The Movie Recommendations

A. Textual Feature Extraction in Movies

The RCNN model, which is designed for text classification [20], is adopted for the textual feature extraction in the proposed VRConvMF. The extracted features are used as a part of the mean of gaussian distribution in the item latent models. In order to obtain more complete contextual information of the text, the recurrent structures are integrated into convolutional layers to further improve the quality of word representations. In this paper we propose the recurrent convolutional matrix factorization called *RConvMF*, which can take advantages of both the RCNN and PMF. In RCNN models, the word representations combine the word and its context together to understand the word more comprehensively. It is supposed that $ct_l(w_i)$ represents the left context of word w_i and $ct_r(w_i)$ represents the right context of word w_i. The left and right contexts of word w_i are defined as:

$$ct_l(w_i) = f\Big(\mathbf{W}^{(l)}ct_l\Big(w_{i-1}\Big) + \mathbf{W}^{(sl)}e\Big(w_{i-1}\Big)\Big), \tag{9}$$

$$ct_r(w_i) = f\Big(\mathbf{W}^{(r)}ct_r\Big(w_{i+1}\Big) + \mathbf{W}^{(sr)}e\Big(w_{i+1}\Big)\Big), \tag{10}$$

where $e(w_i)$ denotes the word embedding of word w_i, $\mathbf{W}^{(l)}$ and $\mathbf{W}^{(r)}$ denote the matrices, which combine all the left and the right context hidden layers respectively, $\mathbf{W}^{(sl)}$ and $\mathbf{W}^{(sr)}$ denote the matrices, which are used to combine the semantic of current word with the left and right contexts of next word respectively, $f(\cdot)$ is a nonlinear activation function, i.e., ReLU. Under this model, the word representation x_i of the word w_i is denoted as follows:

$$x_i = \Big[ct_l\Big(w_i\Big), e\Big(w_i\Big), ct_r\Big(w_i\Big)\Big]. \tag{11}$$

Obviously, $ct_l(w_i)$ and $ct_r(w_i)$ represent all the left and right contextual information of word w_i respectively, and different context window sizes are used to capture different contextual information in order to investigate the performance of RCNN model. For instance, the word representation of w_i is represented by $\Big[x(w_{i-2}); x(w_{i-1}); x(w_i); x(w_{i+1}); x(w_{i+2})\Big]$ when the context window size is 5. Furthermore, an activation function, i.e., *tanh*, is applied to transform x_i into $x_i^{(2)}$ as follows:

$$x_i^{(2)} = tanh\Big(\mathbf{W}^{(2)}x_i + b^{(2)}\Big). \tag{12}$$

Max-pooling layer is applied after all word representations are obtained, and max-pooling operation is able to capture the most significant factors of the whole text. The further textual features can be obtained as follows:

$$x^{(3)} = \max_{i\in\{1,2,\cdots,n\}} x_i^{(2)}. \tag{13}$$

Then, the conventional linear function is applied in neural networks to project the output from pooling layer on $k-$*dimensional* space of item latent models. The final textual feature of descriptive text of movie j is obtained as follows:

$$v_j = \mathbf{W}^{(4)}x^{(3)} + b^{(4)}. \tag{14}$$

For textual feature extraction, the RCNN architecture can be expressed as $\varphi_j = rcnn(\mathbf{W}, X_j)$, where $\mathbf{W}$ denotes all of the weights and biases, and φ_j denotes the textual latent vector of movie j.

B. VGG of Visual Feature Extraction

VGG is a kind of CNN with a deep structure designed by *Visual Geometry Group*. VGG can fully extract the features of each picture [39]. Generally speaking, high-level visual features focus more on semantic information while less on details. Low-level visual features contain more detailed information while they are related to background confusion and semantic ambiguity [40]. Specifically, the VGG19 model is utilized to extract visual features, which contains 19 weight layers, i.e., 16 convolutional layers and 3 fully connected layers. Moreover, the size of all of the convolution kernels is 3×3, which can get the pixel values of adjacent positions with the minimum size. In addition, all of the max-pooling operators are performed with 2×2 pixel windows. All of the hidden layers in VGG19 are equipped with ReLU, a non-linearity activation function. The specific architecture of VGG19 is shown in Fig. 2. We can extract visual features at different depth levels from several specific layers. Different single-level visual features are obtained from different depth pooling layers respectively. The feature cross technology is applied to cascading the multi-level information. Specifically,

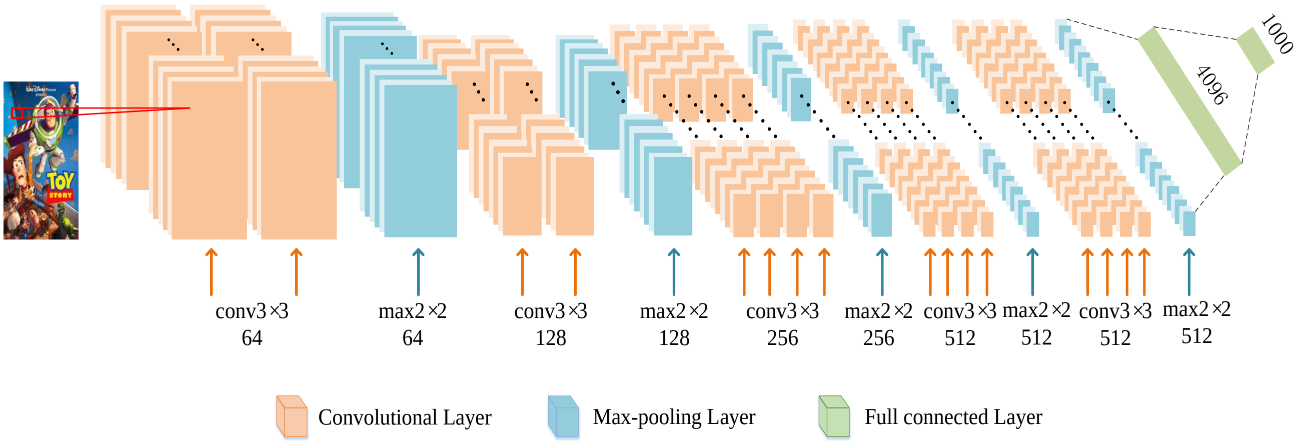


Fig. 2. The architecture of VGG19.

different depth visual features of each image are cascaded as the multi-level visual feature. For visual feature extraction, the VGG19 model is summarized as follows: $\phi_j = vgg(\mathbf{W}', Y_j)$, where Y_j denotes the image of item j (including posters and still frames), and ϕ_j denotes visual features of movie j.

C. Probabilistic Model of RConvMF

The probabilistic model of RConvMF can take fully advantage of both item description documents and ratings by bridging RCNN and PMF. The conditional distribution over observed ratings is as follows:

$$p\Big(\mathbf{R}|\mathbf{U},\mathbf{V},\sigma^2\Big)=\prod_{i=1}^{m}\prod_{j=1}^{n}\Big[N\Big(r_{ij}|u_i^T v_j,\sigma^2\Big)\Big]^{I_{ij}}, \qquad (15)$$

where $N(x|\mu,\sigma^2)$ is the probability density function of the gaussian distribution with mean μ and variance σ^2, and I_{ij} is the index function, which has been mentioned in Section III-A. Then, zero-mean spherical gaussian prior with variance σ_U^2 is placed on user feature vectors:

$$p\Big(\mathbf{U}|\sigma_{\mathbf{U}}^2\Big)=\prod_{i=1}^{m}N\Big(u_i|0,\sigma_{\mathbf{U}}^2 I\Big). \qquad (16)$$

Accordingly, in conventional PMF, item feature vectors with zero-mean spherical gaussian priors are represented by $p(\mathbf{V}|\sigma_{\mathbf{V}}^2) = \prod_{j=1}^{n} N(v_j|0,\sigma_{\mathbf{V}}^2 I)$. Moreover, in probabilistic model of RConvMF, the key idea of connecting the PMF and RCNN is to use a document latent vector obtained from the RCNN model as the mean of gaussian distribution, and gaussian noise of the item as the variance of gaussian distribution. Accordingly, v_j is changed into $v_j = rcnn(\mathbf{W}, X_j) + \varepsilon_j$, where $\varepsilon_j \sim N(0,\sigma_{\mathbf{V}}^2 I)$. The conditional distribution over item latent models is given by:

$$p\Big(\mathbf{V}|\mathbf{W},\mathbf{X},\sigma_{\mathbf{V}}^2\Big)=\prod_{j=1}^{n}N\Big(v_j|rcnn\Big(\mathbf{W},X_j\Big),\sigma_{\mathbf{V}}^2 I\Big), \qquad (17)$$

where $\mathbf{X}$ is the set of descriptive texts of items (such as description of movies and user reviews), and X_j represents the description of movie j. For the weight w in $\mathbf{W}$, which exists in RCNN, we place zero-mean spherical gaussian prior: $p(\mathbf{W}|\sigma_{\mathbf{W}}^2) = \prod_k N(w_k|0,\sigma_{\mathbf{W}}^2)$.

D. VRConvMF Model

In this Section, the proposed model will be further explained. In Sections IV-A and IV-B, the extractions of the textual features and visual features are introduced respectively. Then the above two features are merged by using the following method and integrate recurrent neural network and convolutional network as textual feature extraction. Then we separately cascade the textual features and the corresponding visual features of movies as the comprehensive features and project the comprehensive features through projection layer to specific dimensions.

Accordingly, as for conditional distribution over item feature vectors, we intensify them on the basis of traditional PMF by:

$$p\Big(\mathbf{V}|\mathbf{W}',\mathbf{X},\mathbf{Y},\sigma_V^2\Big)=\prod_{j=1}^{n}N\Big(v_j|\mu_j,\sigma_{\mathbf{V}}^2 I\Big), \qquad (18)$$

where $\mu_j = dense\Big(cascade(\varphi_j,\phi_j)\Big)$, $\mathbf{W}'$ represents the weights and biases of VGG, RCNN and dense layer, and $\mathbf{Y}$ is the set of images of movies. Then, the item latent factor v_j can be expressed as follows:

$$v_j = dense\Big(cascade\Big(\varphi_j,\phi_j\Big)\Big)+\varepsilon_j, \qquad (19)$$

where $\varepsilon_j \sim N(0,\sigma_{\mathbf{V}}^2 I)$, and the *cascade* refers to concatenating the textual features obtained from the RCNN model and the visual features obtained from the VGG network. Then, the features from two different vector spaces can be fused as the comprehensive features. Finally, the obtained v_j is adopted in PMF. The framework of the proposed VRConvMF is shown in Fig. 3. We adopt the Maximum A Posteriori (MAP) estimation to optimize the variables, weights and biases as follows:

$$\max_{\mathbf{U},\mathbf{V},\mathbf{W}'} p\Big(\mathbf{U},\mathbf{V},\mathbf{W}'\Big|\mathbf{R},\mathbf{X},\mathbf{Y},\sigma^2,\sigma^2_{\mathbf{U}},\sigma^2_{\mathbf{V}},\sigma^2_{\mathbf{W}'}\Big)$$
$$= \max_{\mathbf{U},\mathbf{V},\mathbf{W}'} \Big[p\Big(\mathbf{R}\Big|\mathbf{U},\mathbf{V},\sigma^2\Big)p\Big(\mathbf{U}\Big|\sigma^2_{\mathbf{U}}\Big)p\Big(\mathbf{V}\Big|\mathbf{W}',\mathbf{X},\mathbf{Y},\sigma^2_{\mathbf{V}}\Big)p\Big(\mathbf{W}'\Big|\sigma^2_{\mathbf{W}'}\Big)\Big]. \tag{20}$$

The confidence mechanism is adopted in loss function, and the negative logarithm of the posterior distribution over the user features and movie features in Eq. (20) is given by:

$$\mathcal{L}(\mathbf{U},\mathbf{V},\mathbf{W}') = \sum_{i}^{m}\sum_{j}^{n}\frac{c_{ij}}{2}\Big[I_{ij}\Big(r_{ij}-u_i^T v_j\Big)^2 + \lambda_{\mathbf{U}}\sum_{i=1}^{m}\Big\|u_i\Big\|_2 + \lambda_{\mathbf{V}}\sum_{j=1}^{n}\Big\|v_j-\mu_j\Big\|_2 + \lambda_{\mathbf{W}'}\sum_{k}^{w'_k}\Big\|w'_k\Big\|_2\Big], \tag{21}$$

where $\lambda_{\mathbf{U}} = \sigma^2/\sigma^2_{\mathbf{U}}$, $\lambda_{\mathbf{V}} = \sigma^2/\sigma^2_{\mathbf{V}}$, $\lambda_{\mathbf{W}'} = \sigma^2/\sigma^2_{\mathbf{W}'}$.

For the optimization process, the coordinate descent method is adopted to iteratively optimize the latent variable while fixing remaining variables. Specifically, coordinate descent method performs a one-dimensional search along the coordinate direction at the current point to find the minimum of loss function. In addition, the optimal solution of $\mathbf{U}$ and $\mathbf{V}$ can be calculated by the loss function $\mathcal{L}$ with respect to u_i and v_j respectively as follows:

$$u_i \leftarrow \Big(\mathbf{V}I_i\mathbf{V}^T + \lambda_{\mathbf{U}}I_K\Big)^{-1}\mathbf{V}R_i, \tag{22}$$

$$v_j \leftarrow \Big(\mathbf{U}I_j\mathbf{U}^T + \lambda_{\mathbf{V}}I_K\Big)^{-1}\Big(\mathbf{U}R_j + \lambda_V\mu_j\Big), \tag{23}$$

where I_i is a diagonal matrix with I_{ij}, $j = 1, ...n$ as its diagonal elements. And R_i is a vector for user i with $(r_{ij})_{j=1}^n$. For movie j, I_j and R_j are similarly defined as I_i and R_i respectively. The back-propagation algorithm is applied to optimize $\mathbf{W}'$. In the whole optimization process, the unobserved rating of user i on movie j can be predicted as: $\widehat{r_{ij}} \approx E\Big[r_{ij}\Big|u_i^T v_j, \sigma^2\Big] = u_i^T v_j = u_i^T\Big(\mu_j + \varepsilon_j\Big)$.

V. Experiments

A. Experimental Settings

*1) **Datasets**:* We validate the proposed VRConvMF on three different real-world datasets including MovieLens-1m, MovieLens-10m and Amazon Instant Video. In the following parts of this paper, the three datasets are abbreviated as ML-1m, ML-10m and AIV respectively.

- **ML-1m**: It is a movie rating dataset which includes the user ratings (each of which is in terms of a value from 1 to 5), metadata of movies (Movie ID, title and genres) and user attribute information (User ID, gender, age and occupations), etc. Specifically, it contains 6040 users, 3883 movies and 1000209 ratings. It is called 1m because the dataset contains 1M rating data.
- **ML-10m**: Similar to ML-1m, it is also a dataset which is provided by MovieLens website. The dataset contains 71567 users, 10681 movies and 10000054 ratings specifically. And the movies can be classified into 19 categories according to our statistics.
- **AIV**: The Amazon rating dataset contains the user ratings and the metadata of movies from Amazon website. It contains 29757 users, 15149 items and 135188 ratings. According to statistics, there are 21 categories for all the movies.

TABLE I: Data statistics on three real-world datasets.

Datasets	Users	Movies	Ratings	Sparsity
ML-1m	6040	3883	1000209	95.73%
ML-10m	71567	10681	10000054	98.69%
AIV	29757	15149	135188	99.97%

Specifically, ML-1m and ML-10m were obtained from *MovieLens*[*], and the AIV was obtained from *Amazon website of video*[†]. Due to the lack of corresponding auxiliary information, specifically, we utilize web crawler technique to crawl movie description documents for MovieLens datasets, users' reviews for AIV dataset, and movie posters or still frames on *IMDb*[‡], *Netflix*[§] and *Douban*[¶] websites for all of the datasets. Before feeding the proposed VRConvMF, auxiliary information is preprocessed for all of the datasets as follows: 1) for the movies without posters, choose the still frames which are extracted from trailers; 2) remove the movies without any text or image information. The sparsity of the three datasets is shown in TABLE I, which illustrates that the sparsity problem of each dataset is extremely serious.

For the textual feature extraction, similar to [17], we preprocess descriptive texts of movies for all of the datasets as follows: 1) set maximum length of raw text to 400; 2) remove stop words; 3) select top 6000 distinct words as a vocabulary and remove the other words from raw documents. For the visual feature extraction: 1) resize all of the RGB images with the size of $(182 \times 268 \times 3)$; 2) utilize $min-max$ *scaling normalization* to get feature maps; 3) perform *Histogram Equalization* to enhance the quality of posters since a few of movies were released in the early time, the posters of which are not sharp enough.

*2) **Model Settings**:* The dimension of user latent vectors and item latent vectors ($\mathbf{U}$ and $\mathbf{V}$) are both set to 50 (as reported in [15]) and we initialize $\mathbf{U}$ and $\mathbf{V}$ randomly in the range of (0,1). We implement the proposed VConvMF and VRConvMF by using Python and Keras library with Theano backend. The hardware is *Nvidia*1080*Ti GPU* on Linux Ubuntu platform. The dropout method is used to avoid over-fitting and the dropout rate is set to 0.3 during the training of the RCNN and CNN models.

[*] http://files.grouplens.org/datasets/movielens/

[†] http://jmcauley.ucsd.edu/data/amazon/

[‡] http://www.imdb.com/

[§] http://www.netflix.com/

[¶] https://movie.douban.com/

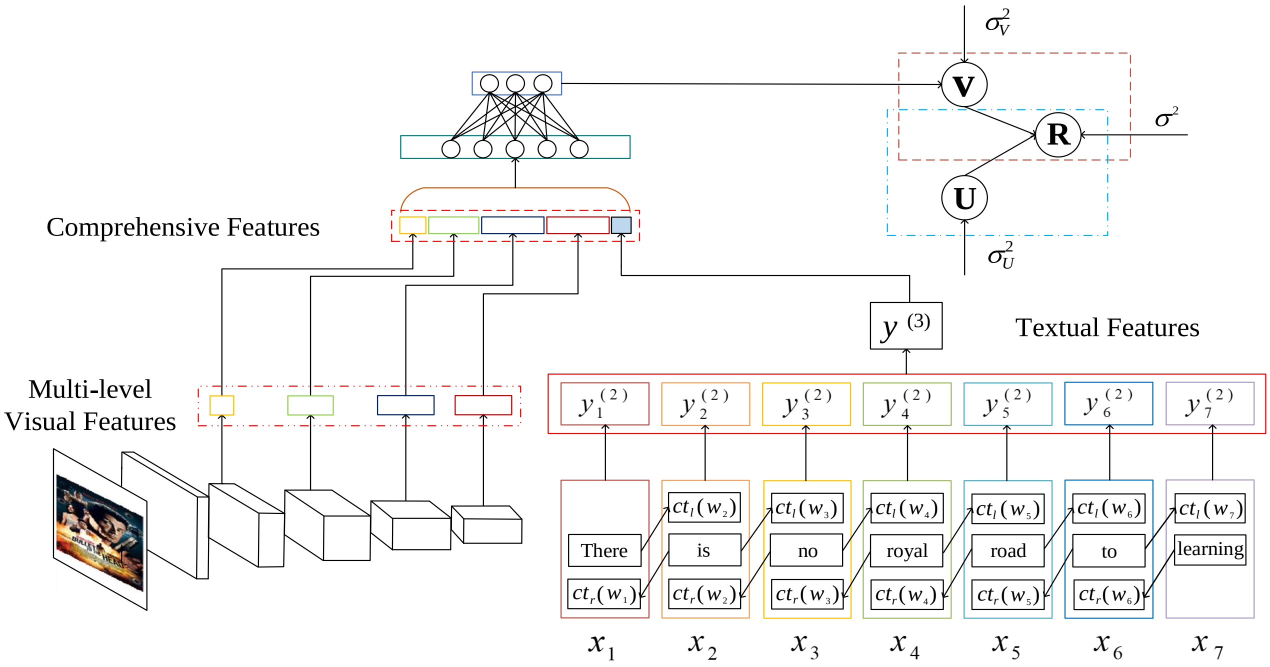


Fig. 3. The architecture of VRConvMF model.

The specific settings and implementations of models are as follows:

- **VConvMF**: 1) for the pre-trained word embedding model, we use *GloVe*, and initialize word vectors randomly with the dimension size of 200, which will be trained through the optimization process; 2) various convolutional window sizes (3, 4, 5) with 100 feature maps are used to obtain different contextual information; 3) mini-batch size is set to 128.
- **VRConvMF**: 1) the pre-trained *GloVe* model is also used as our word embedding model to speed up the training process, 2) the size of word embedding and context window, i.e., $|e|$ and $|c|$, and β are set to 200, 50 and 0.75 respectively, 3) the hidden layer size h is set to 100.

As for visual feature extraction, firstly, imagenet dataset is applied to pre-training VGG19 network. Then, we utilize the weights of the pre-trained VGG19 network as initial weights so that the network can be converged in a short time and improve the performance of visual feature extraction. Similar to [39], batch size, momentum and dropout rate are set to 256, 0.9 and 0.5 respectively. Different from traditional VGG19 network, none of fully connected layers is used when extracting visual features. VGG19 network contains 5 max-pooling layers from shallow to deep. The last 4 max-pooling layer feature maps are taken after pooling operation. Global average pooling method is applied to reduce the multi-dimensional vectors to one-dimensional vectors. The specific dimensions are (1×1×128), (1×1×256), (1×1×512), (1×1×512) for the second, third, forth, and fifth max-pooling layers respectively. We cascade each part of these four layers with the obtained textual feature vector. Obviously, the comprehensive feature for each movie j consists of textual feature and visual feature. Finally, the comprehensive feature vectors are put into the projection layer, fix their dimension to 50, and choose the user latent vectors with the same dimension. For the confidence factor, α is set to 0.3 in Eq. (8). The *Grid Search* method is utilized to find the best performing values of hyper-parameter (λ_U, λ_V) of each model and the results are summarized in TABLE II. Specifically, the detailed parameter optimization experiment processes of the proposed VRConvMF model are shown in Fig. 4. Since the UserCF is the basic collaborative filtering method without PMF, the parameters are not listed. Besides, the parameters of ConvMF+ are the same with that of ConvMF. Since λ_U and λ_V are completely independent, the corresponding optimal solutions can be found respectively.

TABLE II: Parameter setting of λ_U and λ_V.

Model	ML-1m		ML-10m		AIV	
	λ_U	λ_V	λ_U	λ_V	λ_U	λ_V
PMF	0.01	10000	10	100	0.1	0.1
ConvMF	100	10	10	100	1	100
RConvMF	100	10	10	100	1	100
VConvMF	100	10	10	100	10	100
VRConvMF	100	10	10	100	10	100

*3) **Evaluation Metric**:* Before training, we randomly split each dataset into a training set (80%), a test set (10%) and a validation set (10%). Then, in order to validate the performance of the proposed VConvMF model, we select the root mean squared error (RMSE) as the evaluation metric. RMSE can be directly related to an objective function of

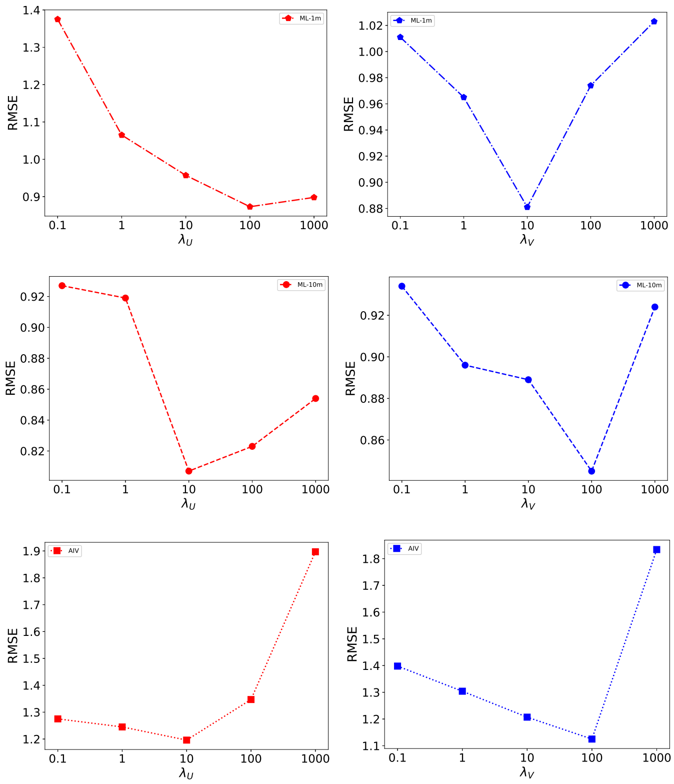


Fig. 4. Parameter analysis of λ_U and λ_V on three datasets of the proposed VRConvMF model.

conventional rating prediction model as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(r_{ij} - \widehat{r_{ij}})^2}{|N|}}, \tag{24}$$

where $|N|$ is the number of test ratings. We set the number of iterations to 30. An average value of 5 repeated experiments are performed as the final result to reduce the random error.

B. Compared Schemes

- **UserCF** [41]: UserCF is a classical recommendation algorithm in collaborative filtering, and predicts missing ratings of target users in terms of ratings of similar users.
- **PMF** [16]: Probabilistic matrix factorization is a basic rating prediction that only uses ratings for collaborative filtering.
- **ConvMF** [21]: Convolutional matrix factorization is another context-aware recommendation model, which combines CNN and PMF to enhance the accuracy of rating prediction.
- **ConvMF+**: The ConvMF+ method considers the confidence mechanism on the basis of ConvMF.
- **RConvMF**: Recurrent convolutional matrix factorization is a context-aware recommendation model, which combines RCNN and PMF to improve the accuracy of rating prediction.
- **VConvMF**: Visual convolutional matrix factorization is the preliminary model, which we have proposed above. VConvMF combines both textual features and multi-level visual features by convolutional neural networks.

*1) **Overall Performance:*** TABLEIII shows the overall rating prediction error of the proposed VRConvMF and other six schemes on three real-world datasets. It shows that compared with the traditional schemes (UserCF and PMF), the context-aware recommendation (ConvMF, RConvMF and VConvMF) can achieve a large improvement and the proposed VRConvMF outperforms other schemes. The comparison of the PMF and ConvMF schemes intuitively illustrates the improvement of the accuracy of the recommender systems by deep context understanding. The comparison of the ConvMF and ConvMF+ methods in TABLE III shows that the confidence mechanism can slightly improve the accuracy of rating prediction. In addition, by comparing the RConvMF model with the ConvMF model, the RCNN model is more effective than the CNN model in textual feature extraction. Finally, as for the proposed VRConvMF model, the visual features are considered to further improve the performance of the RConvMF model. Moreover, from the comparison of the results of the RConvMF model and the VConvMF model, the effect of textual feature is lower than that of visual feature in improving the context-aware recommendation performance. Specifically, compared with the ConvMF, the improvements of VRConvMF are 4.194%, 5.285% and 8.414% for ML-1m, ML-10m and AIV datasets respectively. Moreover, the improvement for the AIV is much more than that for the ML-1m and ML-10m datasets. The results illustrate that the proposed VRConvMF achieves a greater improvement when the dataset has higher sparsity. TABLE III shows that VRConvMF model has smaller RMSE than VConvMF. Therefore, RCNN is more competent in textual feature extraction than CNN in context-aware recommendations.

*2) **Impact of Contextual Information:*** We further conduct experiments to analyze the performance of RCNN and CNN with different context window sizes among two of three datasets (ML-10m and AIV) since ML-1m is a subset of ML-10m. Specifically, odd context window sizes are chosen from 1 to 19 to capture the different contextual information. Small context window sizes may lose a lot of textual semantic information at a long distance, while large context window sizes will lead to data sparsity. Fig. 5 and Fig. 6 show the results of the VRConvMF model when considering contextual information, which is captured by RCNN and CNN with different context window sizes respectively. Obviously, RCNN models outperform CNN models under all different context window sizes. Specifically, the CNN models can achieve the best performance in capturing the contextual information when the window size is set to 5. However, the effect of the context window size on the RCNN models is extremely limited, since the recurrent structure in RCNN can already retain sufficient contextual information.

*3) **Impact of Different Visual Features:*** We conduct extensive experiments to analyze the impact of single-level visual features at different convolutional layers on the performance of VRConvMF. Fig. 7 shows the improvement of

TABLE III: RMSE of overall test sets.

Datasets \ Methods	UserCF	PMF	ConvMF	ConvMF+	RConvMF	VConvMF	VRConvMF	Improved
ML-1m	1.5763	1.0129	0.8560	0.8494	0.8453	0.8361	**0.8201**	4.194%
ML-10m	1.6266	0.9572	0.7966	0.7896	0.7840	0.7756	**0.7545**	5.285%
AIV	1.3780	1.4322	1.1326	1.1297	1.1125	1.0892	**1.0373**	8.414%

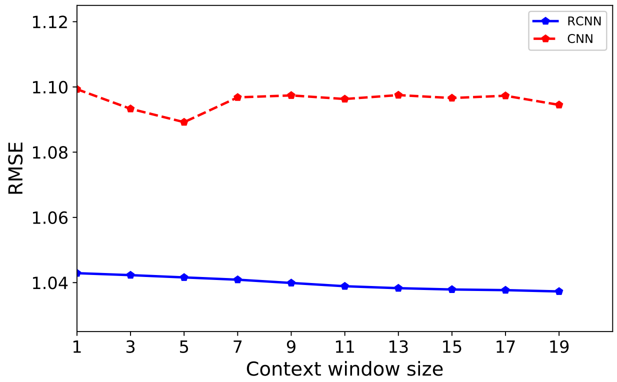


Fig. 5. Comparison of different context window sizes of recurrent structure (AIV).

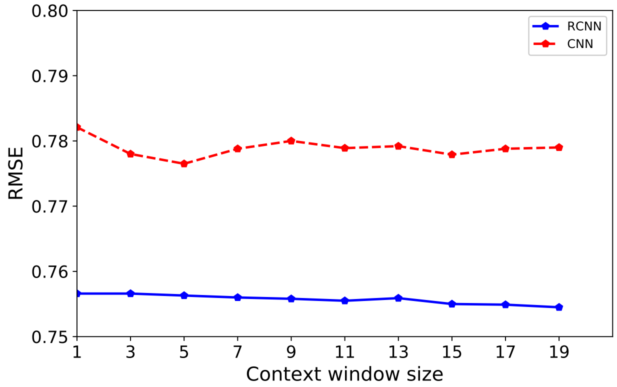


Fig. 6. Comparison of different context window sizes of recurrent structure (ML-10m).

the recommender system when considering the visual features from different levels. Pooling 1 to 4 in Fig. 7 represent the visual features of 4 pooling layers from shallow to deep in the VGG19 network respectively. These single-layers of visual information are combined with RConvMF respectively. It shows that the single-level visual features can reduce the RMSE of the rating prediction to a certain extent. The visual features which are extracted from the mid-level such as pooling 2 tend to outperform those extracted from other pooling layers. Fig. 7 shows that compared with other schemes, the proposed VRConvMF model achieves the best performance as it adds multi-layer visual features extracted from VGG19 on the basis of RCNN as textual feature extraction. Furthermore, the experimental results show that the multi-level visual features can greatly improve the context-aware recommendation performance.

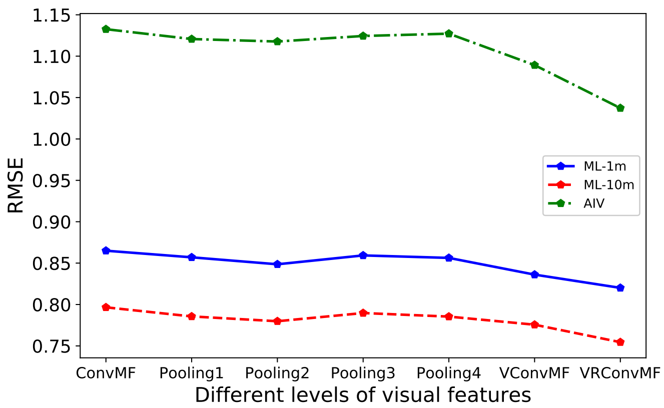


Fig. 7. Comparison of the effects of visual features extracted from different convolutional layers on three real-world datasets.

VI. Conclusions and Future Work

In this paper, we propose a probabilistic matrix factorization based recommendation scheme called visual recurrent convolutional matrix factorization (VRConvMF). The proposed VRConvMF scheme utilizes the textual and multi-level visual features extracted from the descriptive texts and posters respectively to alleviate the sparsity problem. We implement the proposed VRConvMF scheme and conduct extensive experiments on three commonly used real world datasets. The experimental results illustrate that the proposed VRConvMF scheme outperforms the existing schemes.

In the future work, we will crawl corresponding trailers information of movies directly and transfer them into textual information. Since for visual features, more useful information will appear in the trails. Besides, we also intend to consider the user attributes (such as gender, age and occupations) to improve the rating prediction accuracy.

References

[1] X. Ai, H. Chen, K. Lin, Z. Wang, and J. Yu, "Nowhere to Hide: Efficiently Identifying Probabilistic Cloning Attacks in Large-Scale RFID Systems," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 714–727, 2021.

[2] K. Lin, H. Chen, N. Yan, Z. Li, J. Li, and N. Jiang, "Fast and Reliable Missing Tag Detection for Multiple-Group RFID System," *IEEE Transactions on Industrial Informatics, DOI: 10.1109/TII.2021.305895*, 2021.

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

[4] D. W. Oard and J. Kim, "Implicit feedback for recommender systems," in *Proc. of AAAI*, 1998, pp. 81–83.

[5] X. Kong, F. Xia, J. Wang, A. Rahim, and S. K.Das, "Time-location-relationship combined service recommendation based on taxi trajectory data," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1202–1212, 2017.

[6] Y. Xiao, Q. Pei, L. Yao, S. Yu, and X. Wang, "An enhanced probabilistic fairness-aware group recommendation by incorporating social activeness," *Journal of Network and Computer Applications*, vol. 156, pp. 1–17, 2020.

[7] Z. Li, H. Chen, K. Lin, V. Shakhov, L. Shi, and J. Yu, "From edge data to recommendation: A double attention-based deformable convolutional network," *Peer-to-Peer Networking and Applications, DOI: 10.1007/s12083-020-01037-7*, 2021.

[8] H. Chen, S. Wang, N. Jiang, Z. Li, N. Yan, and L. Shi, "Trust-aware Generative Adversarial Network with Recurrent Neural Network for Recommender," *International Journal of Intelligent Systems*, vol. 36, pp. 778–795, 2021.

[9] G. Ling, M. R. Lyu, and I. King, "Ratings meet review, a combined approach to recommend," in *Proc. of ACM RecSys*. Foster City, Silicon Valley, CA, USA: NIPS, October 2014, pp. 105–112.

[10] Y. Moshfeghi, B. Piwowarski, and J. M.Jose, "Handling data sparsity in collaborative filtering using emotion and semantic based features," in *Proc. of ACM SIGIR*, New York, NY, USA, 2011, pp. 625–634.

[11] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. of ACM SIGKDD*, 2011, pp. 448–456.

[12] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, "A novel deep learning-based collaborative filtering model for recommendation system," *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1084–1096, March 2019.

[13] F. Xia, J. Liu, H. Nie, Y. Fu, L. Wan, and X. Kong, "Random walks: a review of algorithms and applications," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 2, pp. 95–107, 2020.

[14] F. Xia, H. Liu, I. Lee, and L. Cao, "Scientific article recommendation: exploiting common author relations and historical preferences," *IEEE Transactions on Big Data*, vol. 2, no. 2, pp. 101–112, 2016.

[15] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. of ACM SIGKDD*, Sydney, NSW, Australia, August 2015, pp. 1235–1244.

[16] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. of NIPS*, 2008, pp. 1257–1264.

[17] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. of EMNLP*, Doha, Qatar, October 2014, pp. 1746–1751.

[18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vector for word representation," in *Proc. of EMNLP*, 2014, pp. 1532–1543.

[19] L. Zhao, Z. Lu, S. J. Pan, and Q. Yang, "Matrix factorization+ for movie recommendation," in *Proc. of IJCAI*, 2016, pp. 3945–3951.

[20] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. of AAAI*, Sydney, NSW, Australia, August 2015, pp. 2267–2273.

[21] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. of ACM RecSys*, Boston, MA, USA, September 2016, pp. 233–240.

[22] H. Chen, J. Fu, L. Zhang, S. Wang, K. Lin, and L. Shi, "Deformable convolutional matrix factorization for document context-aware recommendation in social networks," *IEEE Access*, vol. 7, pp. 66 347–66 357, May 2019.

[23] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proc. of ACM RecSys*, Boston, MA, USA, 2016, pp. 191–198.

[24] S. Duan, D. Zhang, Y. Wang, L. Li, and Y. Zhang, "Jointrec: A deep-learning-based joint cloud video recommendation framework for mobile iot," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1655–1666, March 2020.

[25] Z. Huang, J. Tang, G. Shan, J. Ni, Y. Chen, and C. Wang, "An efficient passenger-hunting recommendation framework with multitask deep learning," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7713–7731, October 2019.

[26] Y. Koren, "Collabortive filtering with temporal dynamics," *Communications of the ACM*, vol. 53, no. 4, pp. 89–97, 2010.

[27] H. Stormer, "Improving e-commerce recommender systems by the identification of seasonal products," in *Proc. of AAAI*, 2007, pp. 92–99.

[28] A. Q. Macedo, L. B. Marinho, and R. L. T. Santos, "Context-aware event recommendation in event-based social networks," in *Proc. of ACM Recsys*, Vienna, Austria, September 2015, pp. 123–130.

[29] S. Purushotham, Y. Liu, and C.-C. J. Kuo, "Collaborative topic regression with social matrix factorization for recommendation systems," in *Proc. of ICML*, Edinburgh, Scotland, UK, 2012, pp. 759–766.

[30] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proc. of ACM RecSys*, New York, NY, USA, 2013, pp. 165–172.

[31] N. Hariri, B. Mobasher, and R. Burke, "Context-aware music recommendation based on latent topic sequential patterns," in *Proc. of ACM RecSys*, Dublin, Ireland, September 2012, pp. 131–138.

[32] R. Cheng and B. Tang, "A music recommendation system based on acoustic features and user personalities," in *Proc. of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2016, pp. 203–213.

[33] S. Oramas, O. Nieto, M. Sordo, and X. Serra, "A deep multimodal approach for cold-start music recommendation," in *Proc. of DLRS*, Como, Italy, June 2017, pp. 32–37.

[34] Y. Fan, Y. Wang, H. Yi, and B. Liu, "Movie recommendation based on visual features of trailers," in *Proc. of International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, Torino, Italy, July 2017, pp. 242–252.

[35] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: a neural image caption generator," in *Proc. of IEEE CVPR*, 2015, pp. 3156–3164.

[36] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, September 2016.

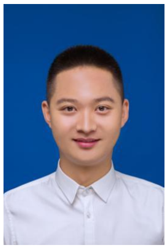
[37] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 42–49, August 2009.

[38] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. of IJCAI*, 2013.

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Science*, pp. 770–778, 2014.

[40] W. Yu, K. Yang, H. Yao, X. Sun, and P. Xu, "Exploiting the complementary strengths of multi-layer cnn features for image retrieval," *Neurocomputing*, vol. 237, pp. 235–241, July 2017.

[41] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. of ACM SIGIR*, Berkley, CA, USA, October 1999, pp. 230–237.



Zhu Wang received the B.E. degree in Electrical Engineering and Automation from University of Jinan, China, in 2017. He is currently a graduate student in the College of Control Science and Engineering, China University of Petroleum, China. His research interest is in the field of recommender system.

Honglong Chen received the M.E. degree in Control Science and Engineering from Zhejiang University, China, in 2008, and the Ph.D degree in computer science from The Hong Kong Polytechnic University, Hong Kong, in 2012. He was a Postdoctoral Researcher in the School of CIDSE at Arizona State University from 2015 to 2016. He is currently an Associate Professor and Ph.D supervisor with the College of Control Science and Engineering, China University of Petroleum, China. He served as a Guest Editor of IEEE Transactions on Industrial Informatics. His current research interests are in the areas of Internet of Things and recommender systems. He has published more than 70 research papers in prestigious journals and conferences including IEEE TIFS, IEEE TMC, IEEE IoT-J, IEEE TII, IEEE TWC, IEEE INFOCOM, IEEE ICPP, IEEE ICDCS, etc. He is a senior member of IEEE and CCF (China Computer Federation).

Zhe Li received the B.E. degree in Automation from Shandong University of Science and Technology, China, in 2018. She is currently a graduate student in the College of Control Science and Engineering, China University of Petroleum, China. Her research interest is in the field of recommender system.

Kai Lin received the B.E. degree in Automation from Tianjin Polytechnic University, China, in 2015 and the M.E. degree in Control Engineering from China University of Petroleum, China, in 2019. He is currently pursuing his Ph.D degree in control science and engineering in the College of Control Science and Engineering, China University of Petroleum, China. His current research interests include RFID system and Internet of things.

Nan Jiang received the Ph.D degree from the Nanjing University of Aeronautics and Astronautics in 2008. From 2013 to 2014, he was a research scholar in the Complex Networks and Security Research Lab at Virginia Tech, Blacksburg, VA, USA. He is currently a Professor with Department of Internet of Things and the director of the Intelligent Sensor Networks Lab at East China Jiaotong University. His current research interests lie in the Internet of Things, Cyber Physical Systems, and Social Networks. He has published more than 60 papers in international journals and conferences and served as program chairs and committee for many international conferences such as CSS 2019, IEEE EUC 2017, ISICA 2015 etc. He is a member of IEEE, a member of ACM, and a senior member of the China Computer Federation (CCF).

Feng Xia received the BSc and PhD degrees from Zhejiang University, Hangzhou, China. He was a Full Professor and Associate Dean (Research) in School of Software, Dalian University of Technology, China. He is currently an Associate Professor and Discipline Leader in School of Engineering, IT and Physical Sciences, Federation University Australia. Dr. Xia has published 2 books and over 300 scientific papers in international journals and conferences (such as IEEE TAI, TKDE, TBD, TCSS, TNSE, TETCI, TC, TMC, TPDS, TETC, THMS, TVT, TITS, TASE, ACM TKDD, TIST, TWEB, TOMM, WWW, AAAI, SIGIR, CIKM, JCDL, EMNLP, and INFOCOM). His research interests include data science, artificial intelligence, and social computing. He is a Senior Member of IEEE and ACM.