CDMF: A Deep Learning Model based on Convolutional and Dense-layer Matrix Factorization for Context-Aware Recommendation

Mingxin Gan, Yingxue Ma and Kejun Xiao

Department of Management Science and Engineering
School of Economics and Management
University of Science and Technology Beijing, Beijing 100083, China
ganmx@ustb.edu.cn

Abstract

This paper proposes a novel deep neural network based recommendation model named Convolutional and Dense-layer Matrix Factorization (CDMF) for Context-Aware Recommendation, which is to combine multi-source information from item description and tag information. CDMF adopts a convolution neural network to extract hidden feature from item description as document and then fuses it with tag information via a full connection layer, thus generates a comprehensive feature vector. Based on the traditional matrix factorization method, CDMF makes rating prediction based on the fused multi-source information of both users and items. Experiments on a real dataset show that the proposed deep learning model obviously outperforms the state-of-art recommendation methods, especially exhibits high robustness in the comparison to a recently developed deep learning method.

1. Introduction

The last few years have witnessed an explosion of information caused by the exponential growth of the Internet and World Wide Web, which confronted us with information overload and brought about an era of big data, appealing for efficient personalized recommender systems to assist the screening of useful information from various sources. To further improve user experiences and reflect personalization of individual users, various recommender systems have also been proposed recently in an active way towards personalized nomination of resources and to assist individuals for efficiently identifying their potential interest, leading to a variety of successful stories in the online recommendations of books [1], movies [2], news [3], TV programs [4], microblogs [5]and many others [6, 7, 8].

Classic recommendation methods do have some shortcomings and deficiencies. The traditional collaborative filtering methods rely on user's rating matrix and it is inferior in recommendation accuracy. Secondly, such traditional methods as regression-based models, network based methods focus on similarity calculation, therefore lack associations extracting from multi-source information for both users and items. Thirdly, such matrix factorization methods currently developed resort to feature extraction, and lack the use of multi-source context information of items, as tags, description, and so on. Deep learning is a research focus in the big data era [9], which is a new and growing field in machine learning with powerful models are developed by utilizing a deep neural network. Existing studies try to use machine learning and deep learning models in recommendation [10, 11]. Such methods based on machine learning use Bayesian methods [12], clustering, ANNs, linear regression to minimize the error between predicted ratings and historical ratings.

Although deep learning models achieved outstanding performance in extracting hidden features, the problem of multi-source information fusion has not been effectively solved in their applications in recommendation. Therefore, to overcome these limitations, we propose a recommendation model based on a deep learning model convolutional neural network, a dense-layer and a matrix factorization. The remaining of the paper is structured as follows. In Section 2 we present an overview of the related work. In Section 3 we present details of the proposed model. We present specific details on experiments and result analysis in Section 4. Finally, we present the discussion and conclusions of the paper in Section 5.

2. Related work

2.1 Classical methods in Recommendation

Collaborative filtering is the most typically designed recommendation method. A user-based design uses historical data to calculate similarities between users and relies on such information to calculate discriminant scores for recommending candidate objects [3, 13]. An item-based design is formally equivalent to its user-based counterpart by exchanging the roles of users and objects [14]. To make use of information of objects, a content-based method further uses properties of objects to characterize their similarities [15]. To promote respective advantages of these two categories of approaches, hybrid approaches have also been proposed [16].

One of the most important aspects in a collaborative filtering method is the quantification of associations that measure relatedness of users or objects. Such a measure is typically derived from a transformation of matched relationships between characteristics of objects and preferences of users [17, 18]. For example, in widely used methods such as the cosine vector similarity, users are represented as vectors of objects according to historical data, and similarities are calculated as the cosine of the angle between the vectors.

Recent studies have also shown that recommendation approaches based on the simulation of a diffusion process exhibit higher performance over classical collaborative filtering methods [8, 19]. For instance, network based methods have been widely used in recent years, which has been demonstrated that the simulation of interactive between users and items effectively improve recommendation accuracy. Besides, the heat-spreading method enhances recommendation diversity [20]. Social network among users extends the user-item relationship network used in recommendation [21]. However, these methods are short in using multi-source information, such as item description and tag information.

2.2 Deep learning for Recommendation

Deep learning methods have been currently adopted in various areas in recommendation, including rating prediction [22], text recommendation [23], image recommendation [24], and location recommendation based on social networks [25, 26].

Deep neural network (DNN) creates a denser high-level semantic abstraction by combining low-level features to automatically discover distributed representations of data, which has been is mainly used in the recommendation system to learn the hidden features from users and items [27]. It usually reconstructs user or item-related information (including rating data, text and images, *etc.*) to obtain an implicit representation for a user or for an item, supporting further predictions on user preferences. DNNs fit complex nonlinear relationships by identifying and learning deep features, often used as tools to extract hidden-layer features in complex associations between users and items [28, 29]. Convolutional neural network (CNN) is a focus in current research for image understanding. It works well when processing image and dealing with feature learning via local perception, pooling dimension reduction.

The idea of local perception has been widely applied in the feature-extracted full-channel and local attention channels, used as feature extraction tools to obtain text features [30]. Multi-mode information also has been adopted in the recommendation of microblogging [31], using CNN and RNN to extract features from images and texts, and then combining them to make tag recommendations. In addition, CNN learns implicit representations of users and items from user reviews [32, 33]. Moreover, CNN has also been used for feature learning and extraction in image-related and music-related recommendations [34], generating better performance to traditional methods.

The application of neural networks in recommendation system is mainly in modeling the sequence impact from data, thus helping to obtain effective representation of users and items. It mainly relies on two aspects: one is the sequence pattern in users' behavior, such as *ratings*, *purchasing*, should be considered in recommendation, and the other is that the use of *words* in modeling text information of both user and item should be extracted with implicit representations [35].

3. Convolutional and dense-layers matrix factorization

3.1. Overview of CDMF

The overview of our proposed method CDMF is illustrated in Figure 1. We firstly initialize a latent vector from Gaussian distribution randomly for the representation of user information. Secondly, as for the item description, we resort to a CNN model in feature extraction and a vector that represents hidden features is generated Figure 1(1b). Thirdly, combined with the tag vectors as the input, we fuse multi-source information and generate feature vectors that represent items. Thirdly, the loss function is defined as the error between predicted rating and actual rating to adjust model parameters based on a back

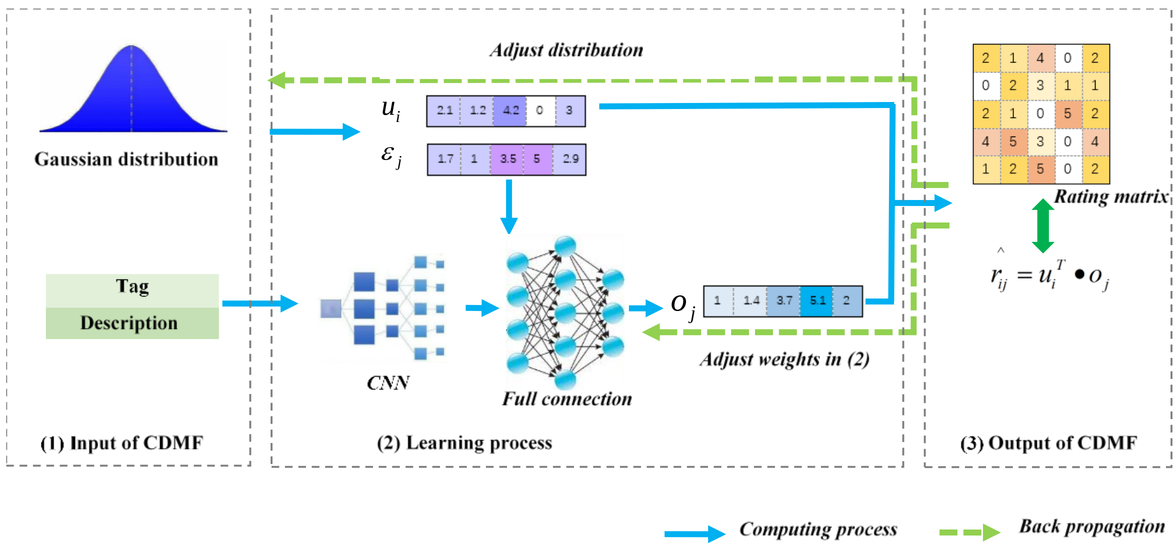

Figure 1: Convolutional and dense-layers matrix factorization (The black arrow indicate tensor forward flows and green is update parameters backpropagation)

propagation process, and then obtain the optimal parameters for recommendation. Finally, combined with the user vector, we obtain predicted ratings via matrix factorization.

3.2. Contextual information

When two items are similar in their descriptions, the combined consideration of tag information and item description can distinguish them obviously. For example, two movies with similar storyline may not be distinguished only by their descriptions, while tag information, such as different year and actors, can help. Therefore, we adopt two kinds of contextual information for items, as listed in Table 1.

Table 1. Tag information and item description.

Feature	Value	Examples
Tag	Year, genre, actor	1998, Adventure| Animation| Children|, Jones
Description	Summary, storyline	Four 12-year-old girls grow up together during an eventful small-town summer in 1970.

Tags Information. An item generally has multiple attributes with different values. For example, a certain attribute set Thus we denote an item matrix $\mathbf{A}_{(n,\ k)}$ as a binary matrix, where $A=\{T_{11},\ T_{12},\ \ldots T_{1n};\ T_{21},\ T_{22},\ \ldots T_{2m};\ \ldots T_{k1},\ T_{k2},\ \ldots T_{kt}\}$ and the element $\mathbf{A}_{i,j}=1$ presents that item i has attribute j.

Item Description. Contexts for items, *e.g.* the summary of a movie, reflect an overall profile or meaningful descriptions, which are comprehensive for human but need to be automatically extracted and converted into formal and uniform vectors for machine learning. To achieve this, we resort to a widely used natural language processing method Word2Vec, to get feature vectors for items and further calculate their similarities [36].

3.3. Architecture of CDMF

We illustrate the proposed method Convolutional and Dense-layers Matrix Factorization (CDMF), by firstly illuminating the feature extraction process from item description, secondly explaining the dimension reduction process via a full connection layer, and finally developing the information fusion process.

3.2.1 Improved convolutional neural network (CNN)

A general convolutional neural network includes an input layer, a convolution layer, a pooling layer, a full connection layer and an output layer. The convolutional layer performs convolution calculation on input data through convolution kernels, adds bias to obtain the feature map after feature extraction, and adds the weighted sum of multiple feature mappings as the input of pooling layer. After many repeats of similar operation, the features from bottom layer to the high layer are finally delivered to full connected layer to get significant feature extraction. The pooling layer is also called subsampling layer, which reduces the dimension of data, and retains representative features to reduce the amount of computation in the model.

The improved convolutional neural network. Different from traditional methods of text similarity calculation, we take item description as the input to realize the automatically extraction process for hidden features that are first represented as vectors and then learned via different convolution kernels.

(1) Embedding layer. We first consider the item description as a series of words each of which corresponds to a vector. Thus each description can be represented by a matrix $\mathbf{W} \in R^{k \times l}$, where k denotes the number of words the description contains and l denotes the dimension of the word vector. Then we adopt Word2Vec in the embedding layer to convert item description into corresponding matrix **W** as the input of CNN to further learn hidden features of these descriptions (Figure 2), which has not been considered by existing studies using traditional text analysis.

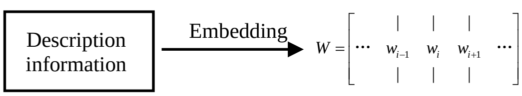

Figure 2: Embedding for description information

(2) Convolution layer. Based on local perception and parameter sharing. Local perception ideology originates from biological nervous system. The visual neuron only responds to local stimuli and obtains a global understanding from multiple local area. In CNN, the local feature extraction of image data is performed by setting a convolution kernel on the image.

Therefore, we use a convolution kernel $\mathbf{F}_k$ to represent the parameter set that is autonomously adjusted according the inverse propagation, whose size is usually $s \times s$, as $\mathbf{F}_k \in R^{s \times s}$. Then we obtain the feature map $\mathbf{M}_k$ via learning from the convolution kernel k calculated by the following formula:

$$M_k = f(W * F_k + b_k) \tag{1}$$

where * denotes convolution operation, b_k is the bias term and f is the nonlinear activation function (*e.g.* tanh).

(3) The pooling layer reduces the dimension of convolution results and reduces data volume in feature map while preserving important features, especially when multi-convolution kernels are used overfitting is controlled.

(4) The output layer. We use *max pooling* to extract features and get low-dimensional vectors for item descriptions, as below:

$$P_{\max} = max(M_1,\ M_2,\ \cdots\ M_{z-s+1}) \tag{2}$$

3.2.2 Dimension reduction via a full connection layer

To effectively use multi-source information of items for recommendation, we fuse information of item tags, text description and user-item ratings.

First, we merge two obtained vectors directly, one of which represents full information extracted form related tags for an item, and the other is the low-dimensional vector extracted by the improved convolutional neural network, explained in the above sub-section, as the item descriptions. Thus two vectors are merged into a o-dimension one representing full information for the item.

Then, to avoid the sparsity and high-dimension of the vectors and emphasize the most important features for the item, we use a full connection layer to make dimensional deduction for the merged vector and obtain an r-dimension vector.

3.4 Formulation of CDMF

Assuming there exist totally $|U|$ users and $|V|$ items, and the ratings matrix is 1, CDMF takes the initial item information as the input, as shown in Figure 1(1a).

(1) As for user i, ($i \in (1, |U|)$), CDMF initializes a latent vector from *Gaussian distribution* randomly, namely $u_i \in U, U \sim N\left(0, \sigma^2 \cdot \mathbf{I}_i\right)$, where $\mathbf{I}_{i,j}$ equals 1 representing the target user i rated the item j.

(2) For item j, the feature vector o_j consists of both the item description and tag information.

a. Hidden feature from item description. We obtain d_j from item description D_j via CNN, as

$$d_j = CNN\left(w_c, D_j\right) \tag{3}$$

where w_c represents the matrix containing the weights and bias of the constructed CNN.

b. Hidden feature from tag. We obtain t_j from tag information, as

$$t_j = \tilde{w}_t \cdot T_j, \tilde{w}_t = \left(w_t, b_t\right) \cdot w_e \tag{4}$$

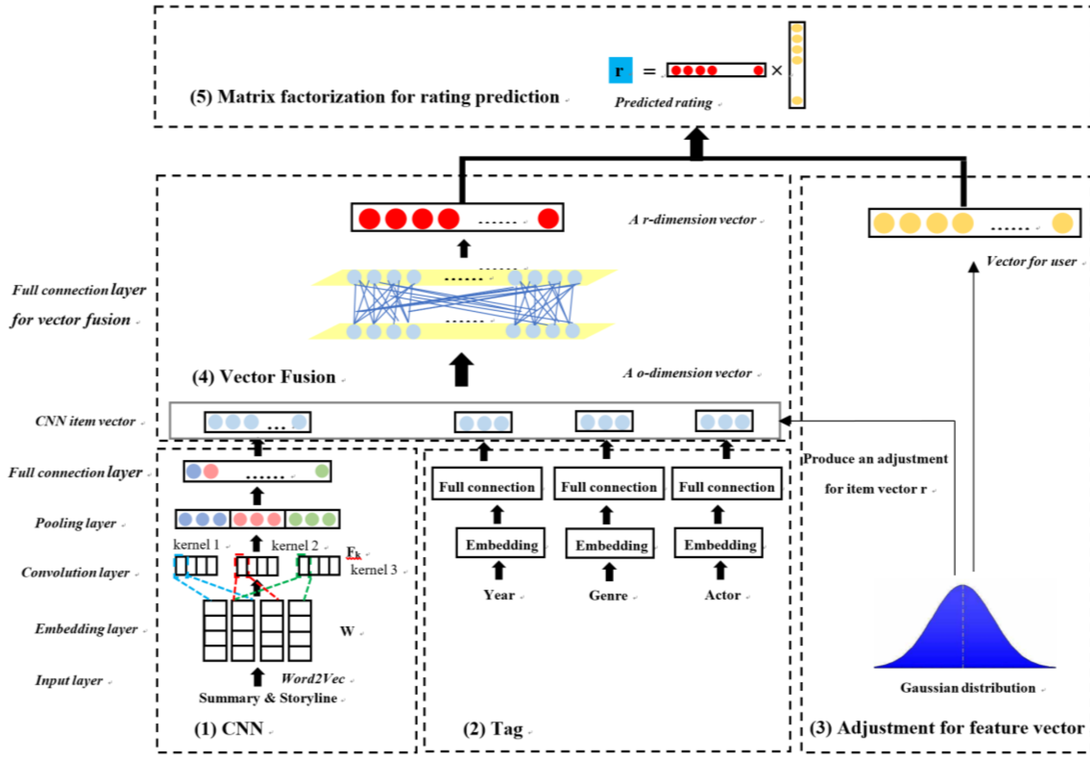

Figure 3: Convolutional and Dense-layers Matrix Factorization (CDMF)

where w_e is regard as the weights and bias in the embedding layer for item j, w_t is the weight and b_t is the bias when generating the tag vector from real tags.

c. To further optimize the model, we generate ε_j randomly for item j as an adjustment from *Gaussian distribution*.

$$\varepsilon_j \sim N\left(0, \sigma^2 \cdot I\right) \tag{5}$$

Then we fuse d_j and t_j to generate o_j, as

$$o_j = w_j^d \cdot d_j + w_j^t \cdot t_j + \varepsilon_j \tag{6}$$

where w_j^d and w_j^t are the weights in full connection layer, which control the dimension of the output vectors for the item description and tag information, respectively.

(3) For each user-item pair, we draw the rating r_{ij} as

$$r_{ij} \sim N\left(u_i^T \cdot o_j, \varepsilon_{ij} \cdot I_{ij}\right) \tag{7}$$

We adopt matrix factorization for rating prediction, which is based on the mathematical matrix eigenvalue factorization to decompose a rating score matrix $R_{(|U|, |V|)}$ into a user matrix $U_{(|U|, k)}$ and an item matrix $O_{(k, |V|)}$, and uses the product of $U_{(|U|, k)}$ and $O_{(k, |V|)}$ to represent $R_{(|U|, |V|)}$, where $|U|$ is the number of user and $|V|$ is the number of item. Thus, a high-dimensional sparse rating matrix is decomposed into two matrices with lower dimensions and less sparseness, so that the rating of each user on the item $\hat{r}_{ij} = u_i \times o_j^T$ can be obtained by the inner product of the two low-dimensional vectors. According to the descending order of the predicted rating, we obtain recommendation lists for all user [37].

3.5. The loss function

We use back propagation for parameters optimization of the loss function L which is defined as the error between the inner product of the hidden feature vectors of both the target user and his/her rated item and the history rating the user given to the item, as:

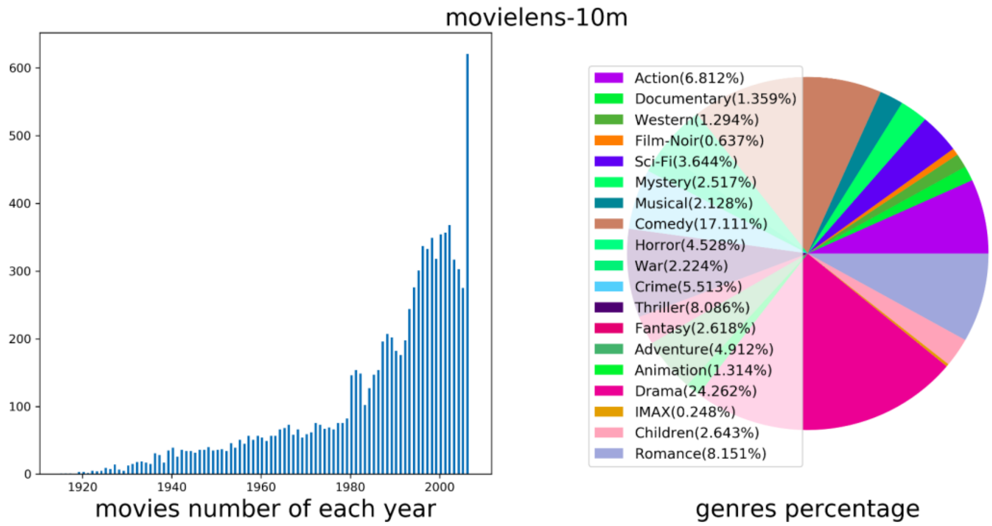

Figure 4: Distribution on year and genre.

$$
\begin{aligned}
L = & \sum_{i}^{|U|}\sum_{j}^{|V|}\left(r_{ij}-\hat{r}_{ij}\right)^2+\frac{\lambda_u}{2}\sum_{i}^{|U|}\left\|u_i\right\|^2 \\
& +\frac{\lambda_o}{2}\sum_{j}^{|V|}\left\|o_j-cnn\left(w_c,D_j\right)-\tilde{w}_t\cdot T_j\right\|^2 \\
& +\frac{\lambda_c}{2}\sum_{c}^{|w_c|}\left\|w_c\right\|^2+\frac{\lambda_t}{2}\left\|\tilde{w}_t\right\|^2
\end{aligned} \tag{8}
$$

where $\hat{r}_{ij}$ and r_{ij} is the predicted rating and the actual rating, respectively, λ_u, λ_o, λ_c and λ_t are regularizations for the user vector, the item vector, CNN kernels and weights of dense-layer for tag's features extraction.

We use back propagation process for parameters update of the loss function L, which iteratively optimizes all the latent variables one by one, while fixing the remaining variables. For example, we take u_i (o_j) as single variable while temporarily assuming $\tilde{w}_t$ and o_j (u_i) to be constant. Then, the optimal solution of u_i (o_j) are analytically computed with a closed form as simply differentiating the optimization function L. The principles of updating for all vectors and weights are as follows.

$$
\begin{aligned}
& u_i \leftarrow \left(V\cdot I_i\cdot V^T+\lambda_u\cdot I_k\right)^{-1}\cdot V\cdot R_i \\
& o_j \leftarrow \left(U\cdot I_j\cdot U^T+\lambda_o\cdot I_k\right)^{-1}\cdot\left(U\cdot R_j+\lambda_o\cdot CNN\left(w_c,D_j\right)+\lambda_o\cdot\tilde{w}_t\cdot T_j\right) \\
& \tilde{w}_t \leftarrow \left(\lambda_t\cdot I_k\right)^{-1}\cdot\left(V\cdot R_i+\lambda_o\cdot CNN\left(w_c,D_j\right)+\lambda_o\cdot T_j\right)
\end{aligned} \tag{9}
$$

When optimizing convolution kernel parameters, we regard the above regularizations and other variables as constant to obtain the following principle for kernel optimization.

$$
\begin{aligned}
F\left(w_c\right) = & \frac{\lambda_o}{2}\sum_{j}^{|V|}\left\|o_j-CNN\left(w_c,D_j\right)-constant_1\right\|^2 \\
& +\frac{\lambda_c}{2}\sum_{c}^{|w_c|}\left\|w_c\right\|^2+constant_2
\end{aligned} \tag{10}
$$

Consequently, we obtain the parametric model with the optimalized loss function and thus predicted ratings.

4. Experiment and result

4.1 Dataset and evaluation criteria

We use two datasets obtained from MovieLens-1m and MovieLens-10m (https://grouplens.org/). Both of the datasets consist of users' explicit ratings on items on range of 1 to 5 and tags information of items (movies), such as year, genres, actors (Table 1). About description documents such as summaries and storylines of movie and clerks table of item, we obtained them from movies information pages on IMDB provided by MovieLens researchers (https://www.imdb.com/). For optimizing results, we drop items that are lack of above information and rated less than twice. Thus the dataset contains 9,390,713 ratings by 69,878 users on 9,492 movies, the mean length of description of each movie is 72.5 and the overall density is 1.42%.

We preprocessed item description according to [37, 38], by firstly sentences to set maximum length of raw documents to 195 (the maximum length of all cleaned sentences in the dataset), then removing stop words, stemming word by Porter stemmer and thirdly selecting most frequency occurring 8000 words to build vocabulary, and finally setting all out-of-vocabulary words to special label. To fuse tags information of movies into our model, we gathered three important tags for each movie, they are year, genres and actors. We only select actors who occurs twice or more in the dataset and top 10 actors for each movie. Figure 4 shows the distribution of the tag information.

To evaluate recommendation performance of each model on the movielens dataset, we randomly divided the data set into a training set (80%), a validation set (10%) and a test set (10%). We use RMSE as the evaluation criteria, and the difference between the predicted and actual ratings represents the performance of the model. We use results in validation set to tune our model and finally report the results evaluating in test set.

$$RMSE = \sqrt{\frac{\sum_{i}^{|U|}\sum_{j}^{|V|}\left(r_{ij} - \hat{r}_{ij}\right)^2}{n\_ratings}} \tag{11}$$

4.2. Experiment and Parameter setting

The experiment is performed on 3 NVidia Geforce GTX 1080 Ti GPU. We adopt Tensorflow 1.6 to conduct our proposed model CDMF based on the convolutional neural network. For training CDMF, we used mini-batch based RMSprop and the batch size is 256. We initialize different matrices for different tags and utilize optimization method mentioned above to learning context-aware representation from item description, historical ratings as well as tag information with their vectorization. The dimension of each tag vector w_t is set to 50 and thus the matrix of tag information is with the size of the number of distinct tags × 50.

After the mapping process for tag information via the embedding layer, as Figure 3 (2) illustrated, the dimension of the dense-layer is set to 100 for learning high-level features. When training the CDMF model for item descriptions as documents, we set the word embedding size to 128 and the word window size to 3, 4, and 5, respectively, in the parallel processing. In each of the three convolution kernels, the number of the output channels is 100. To prevent the overfitting in convolutional neural network, we use L_2 regularization for the convolution kernel, and set dropout keep probability of outputs of max pooling process to 0.2.

4.3 Methods for Comparison

To evaluate our proposed method, we select four comparison methods and illustrate them in brief as below.

NMF: Non-negative Matrix Factorization is a state-of-the-art recommendation model [39].

PMF: Probabilitisitic Matrix Factorization, which is a standard rating prediction model that only uses ratings [40].

CTR: Collaborative topic regression (CTR) is a recently proposed tightly coupled method, which was proposed for recommendation on implicit feedback data. CTR is a probabilistic graphical model that seamlessly integrates a topic model, Latent Dirichlet Allocation (LDA), and a model-based CF method, probabilistic matrix factorization (PMF) [41]. In our experiment, CTR is for processing text topics in item description but without using tag information. Because we use explicit feedback datasets, we set the rating to 1 if u_i rate on o_j and 0 otherwise.

ConvMF: Convolutional Matrix Factorization is a deep learning method for recommendation, it use CNN to extract abstract description hidden features. It has been demonstrated that ConvMF significantly outperform the state-of-the-art recommendation models even if the rating data is extremely sparse [42].

The best performance for each model is at the parameter values shown in Table 2.

Table 2. Parameters for each model.

Model	λ_u	λ_o
NMF	0.2	0.2
PMF	0.002	0.02
CTR	0.02	0.2
ConvMF	0.02	0.2
CDMF	0.02	0.02

As for CTR, such parameters as *number of topics*, α and β were set according to [41]. As for ConvMF, other parameters are set according to [42], except that the maximum length of each sentence has been set to 195 mentioned above.

4.4. Experiment result

4.4.1 Result analysis

Experiment results on MovieLens-10m dataset are shown in Table 3 and 4. In general, our proposed model CDMF obviously outperforms other models. For instance, the improvements of CDMF compared to NMF and PMF are around 11.00%, and 4.26%, respectively.

Table 3. RMSE of different model

Model	RMSE	Improvement
NMF	0.885	11.00%
PMF	0.822	4.26%
CTR	0.819	3.93%
ConvMF	0.799	1.56%
CDMF	0.787	-

Table 4. RMSE under different quantity of noises

Noise	0	5%	10%	15%	20%
ConvMF	0.799	0.801	0.804	0.805	0.807
CDMF	0.787	0.788	0.790	0.791	0.793
Improvement	1.56%	1.62%	1.69%	1.70%	1.79%

NMF is very similar to PMF, but the latent factors of users and items are positive. So NMF only reflects users' preferences, and cannot reflect users' negative opinions on items. Consequently, the performance of NMF is worst. Although PMF is good at avoiding overfitting, the ability of learning latent factors is a little bit poor, and history ratings are adopted only in PMF, so less information is utilized than CTR, ConvMF and CDMF.

The comparison between ConvMF and CTR indicates that the representation of sentences learnt by topic model is worse than convolutional neural network. The limitation of LDA topics mixture of CTR is the lack of consideration of the relative positions among words, perhaps caused by its underlying unigram text pattern [41]. Hence, the convolutional neural network ConvMF adopted which resorts to three different window-size kernels outperforms CTR on semantic extraction.

ConvMF use convolutional neural network to extract features from item descriptions, this is proved that is an effectivity method [42]. However, the phenomenon that item descriptions may contain many noises from text, *e.g.* html labels, wrong characters, has not been considered by ConvMF, consequently, may influence the performance of the recommendation.

Therefore, we additionally consider the noises from item descriptions and implemented an extra experiment, adding noises to item descriptions. We choose ConvMF, the most effective one in comparisons, to carry out this experiment. Insert randomly different proportion of disturb words to the text according to the length of origin sentences (For example, there is a sentence consisting of 70 words, we insert randomly 14 words into the sentence if the noise rate is 20%).

The results of the experiment are shown in Table 3. We see that the RMSE of CDMF is always lower than ConvMF. With the increase of the ratio of inserted noise, the improvement bright by CDMF is become larger, which shows that CDMF effectively learn difference from context as tags, on the based of which, to decrease errors caused by noises from other information source, *e.g.* ratings, item descriptions. Therefore, CDMF has higher robustness than ConvMF.

The results of both experiments demonstrate that CDMF integrating tags and documents simultaneously outperforms ConvMF. It is proved that the fused comprehensive feature vector effectively reflect external characteristics of items, *e.g.* tags, and help to generate item latent vector.

It indicates that using tag information and textual modeling together addresses the problems lacking of clearly source of text. In a word, CDMF inherits the idea of classical recommendation method on using rating matrix, and at the same time, the integration of the comprehensive item information has obtained a more accurate prediction result, and the RMSE was decreased over 1.56% by the best competitor ConvMF.

4.4.2 Parameter analysis

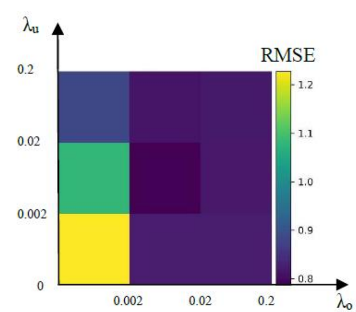

Figure 5. Relation of regularizations with RMSE

Two important regularizations in CDMF are tested. Regularization plays the role of variable punishment, the larger the regularization, the greater the punishment, and the weaker the learning of the variable. The change of RMSE with two parameters is shown in Figure 5. It can be seen that the lowest RMSE is produced by λ_u*=0.02* and λ_o*=0.02*.

The RMSE on the right side of the whole graph is lower than that on the left, especially for the same user regularization λ_u and different item regularization λ_o, which indicates that the learning results of the model are not influenced by user vectors. Besides, the higher punishment for item latent factors lead to a better result.

So the improvement of RMSE achieved by CDMF is caused by the latent factors of items, which fully proves that the feature vectors extracted by CDMF can represent items more effectively and promote recommendation accuracy.

4.4.3. Complexity analysis

The time complexity of CDMF is related to N_R, the number of ratings, $|U|$ and $|V|$ the number of users and items. Stipulate $O(N_R)$ is the time complexity when lookup user/item vectors. Defined k as the dimension of user and item vectors, matrix multiple time complexity is $O(k^3 \cdot |U| + k^3 \cdot |V|)$, updating user/item vectors complexity is $O(k2 \cdot N_R)$. In addition, the time complexity of updating CNN kernel is $O(N_C \cdot h \cdot w \cdot |V|)$, where h and w are the height and width of kernels, the time complexity for updating w_t is $O(l^3 \cdot |V| + l^2 \cdot N_R)$. So total time complexity is $O(N_R + k^3 \cdot |U| + k^3 \cdot |V| + l^3 \cdot |V| + k^2 \cdot N_R + l^2 \cdot N_R + N_C \cdot h \cdot w \cdot |V|)$, and this scales linearly with the size of given data, such as $|U|$,$|V|$ and N_R.

5. Discussion and conclusions

In this paper, we combine item description and tag information for personalized recommendation. Firstly, the item description is embed to a matrix via the embedding method word2vec, then hidden features are extracted by a convolution neural network, and the tag information of items is integrated. After a full connection layer for fusing different features, a comprehensive feature vector for items is generated. We set deviation adjustment for vectors of both users and items, and finally get the predicted user ratings by the inner product of the item and user vector according to the idea of matrix factorization.

Experiments on a real-world dataset Movielens shows that the proposed model CDMF outperforms other methods in comparison, indicating that, considering item information does have a positive effect on the improvement of recommendation accuracy to recommendation methods which merely rely on the ratings.

As for the future work, the method of matrix factorization is a basic one, so we can consider other prediction methods, such as PMF, SVD and deep neural network models. Secondly, we choose a simple structure when we use CNN to extract hidden features from item description. Therefore, the innovation of CNN structure are supposed to be further considered, especially, the idea of three-channel for the fusion of multi-source information in the structure of CNN, like image recognition.

6. References

[1] B. Jeong, J. Lee, and H. Cho, "Improving memory-based collaborative filtering via similarity updating and prediction modulation", Information Sciences, 2010, pp. 602-612.

[2] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering", IEEE Internet Computing, 2003, pp. 76-80.

[3] G. Nie, H. Xia, and X. Li, "An Ontology-Based Approach on Intelligent Recommendation in Movie Field", Proceedings of the 6th International Conference on Innovation and Management, 2009, pp. 1489–1494.

[4] T. Bogers, and A. V. D. Bosch, "Fusing Recommendations for Social Bookmarking Web Sites", International Journal of Electronic Commerce, 2011, pp. 31-72.

[5] S. Prawesh, and B. Padmanabhan, "Probabilistic news recommender systems with feedback", 2012, pp. 257-260.

[6] A. B. Barragáns-Martínez, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition", Information Sciences, 2010, pp. 4290-4311.

[7] J.S. Cheng, A. Sun, D. N. Hu, and D. Zheng, "An Information Diffusion Based Recommendation Framework for Micro-Blogging", Social Science Electronic Publishing, 2011, pp. 463-486.

[8] B. Sarwar, G. Karypis, G.Badrul, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms", International Conference on World Wide Web ACM, New York, 2001, pp. 285-295.

[9] A. M. Elkahky, Y. Song, and X. He, "A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems", International Conference on World Wide Web, Florence, Italy, 2015, pp. 278-288.

[10] X. X. Wang, and Y. Wang, "Improving Content-based and Hybrid Music Recommendation using Deep Learning." ACM International Conference on Multimedia ACM, Orlando Florida, USA, 2014, pp. 627-636.

[11] A. Singhal, P. Sinha, and R. Pant, "Use of Deep Learning in Modern Recommendation System: A Summary of Recent Works", International Journal of Computer Applications, 2017, pp. 17-22.

[12] C. Ono, Y. Motomura, and H. Asoh, "Recommendation Algorithm based on Bayesian Network", Ieice Technical Report Neurocomputing, 2004, pp. 55-60.

[13] O. Georgiou, and N. Tsapatsoulis, "The Importance of Similarity Metrics for Representative Users Identification in Recommender Systems", Artificial Intelligence Applications and Innovations Proceedings DBLP, Ayia Napa, Cyprus, 2010, pp. 12-21.

[14] S.Y. Wei, N. Ye, S. Zhang, X. Huang, and J. Zhu, "Item-Based Collaborative Filtering Recommendation Algorithm Combining Item Category with Interestingness Measure", International Conference on Computer Science & Service System IEEE, 2012, pp. 2038-2041.

[15] I. J. Pérez, F. J. Cabrerizo, and E. Herrera-Viedma, "Group decision making problems in a linguistic and dynamic context", Expert Systems with Applications, 2011, pp. 1675-1688.

[16] R. Burke, "Hybrid Recommender Systems: Survey and experiments", Interaction, 2002, pp. 331-370.

[17] F. Francois, A. Pirotte, J.M. Renders, and M. Saerens, "Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation", IEEE Transactions on Knowledge & Data Engineering, 2007, pp. 355-369.

[18] Q. M. Zhang, A. Zeng, and M. S. Shang, "Extracting the information backbone in online system", Plos One, 2013, pp. e62624.

[19] F. François Fouss, K. Francoisse, L. Yen, A. Pirotte, and M. Saerens, "An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification", Neural Networks, 2012, pp. 53-72.

[20] T. Zhou, Z. Kuscsik, J. G. Liu, M. Medo, J. R. Wakeling, and Y. C. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems", Proceedings of the National Academy of Sciences of the United States of America, 2008, pp. 4511-4515.

[21] J. C. Ying, W. N. Kuo, V. S. Tseng, and H. C. Lu, "Mining user check-in behavior with a random walk for urban point-of-interest recommendations", Acm Transactions on Intelligent Systems & Technology, 2014, pp. 1-26.

[22] S. Seo, J. Huang, H. Yang, and Y. Liu, "Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction", Eleventh ACM Conference on Recommender Systems ACM, Como, Italy, 2017, pp. 297-305.

[23] Y. Li, J. Nie, Y. Zhang, B. Wang, B. Yan, and F. Weng, "Contextual recommendation based on text mining", International Conference on Computational Linguistics: Posters Association for Computational Linguistics, 2010, pp. 692-700.

[24] C. Lei, D. Liu, W. Li, Z. J.Zha, and H. Li, "Comparative Deep Learning of Hybrid Representations for Image Recommendations", Computer Vision and Pattern Recognition IEEE, Las Vegas, Nevada, 2016, pp. 2545-2553.

[25] R. F. Feitosa, S. Labidi, and A. L. S. D. Santos, "Social Recommendation in Location-Based Social Network Using Text Mining", 2013, pp. 67-72.

[26] Zeng, Xuelin, and W. U. Bin, "Parallelized recommendation algorithm in location-based social network", Journal of Computer Applications, 2016.

[27] S. David, H. Aja, J. M. Chris, G. L. S. Arthur, and D. George, “Mastering the game of go with deep neural networks and tree search”, Nature, 2016, pp. 484-489.

[28] J. Peng, and D. D. Zeng, "Exploring Information Hidden in Tags: A Subject-Based Item Recommendation Approach", Social Science Electronic Publishing, 2010.

[29] Q. T. Truong, and H. W. Lauw, "Visual Sentiment Analysis for Review Images with Item-Oriented and User-Oriented CNN", ACM on Multimedia Conference ACM, California, USA, 2017, pp. 1274-1282.

[30] S. Novoselov, O. Kudashev, V. Schemelinin, I. Kremnev, and G. Lavrentyeva, "Deep CNN based feature extractor for text-prompted speaker recognition", 2018.

[31] Y. Y. Gong, and Q. Zhang, "Hashtag recommendation using attention-based convolutional neural network", International Joint Conference on Artificial Intelligence AAAI Press, New York, 2016, pp. 2782-2788.

[32] S. Seo, J. Huang, and H. Yang, "Representation Learning of Users and Items for Review Rating Prediction Using Attention-based Convolutional Neural Network", International Workshop on Machine Learning Methods for Recommender Systems, 2017.

[33] X. Wang, L. Yu, and K. Ren K, "Dynamic Attention Deep Model for Article Recommendation by Learning Human Editors' Demonstration", Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2017, pp. 2051-2059.

[34] S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation", International Conference on Neural Information Processing Systems Curran Associates Inc, 2013, pp. 2643-2651.

[35] L. H. Li, W. Chu, Langford, and John, "A contextual-bandit approach to personalized news article recommendation", 2010, pp. 661-670.

[36] Y. A. Chung, C. C. Wu, C. H. Shen, H. Y. Lee, and L. S. Lee, "Audio Word2Vec: Unsupervised Learning of Audio Segment Representations Using Sequence-to-Sequence Autoencoder", INTERSPEECH, 2016, pp. 765-769.

[37] Y. Koren, R. Bell, and C. Volinsky. "Matrix Factorization Techniques for Recommender Systems", Computer, 2009, pp. 30-37.

[38] C. Wang, and D. M. Blei, "Collaborative topic modeling for recommending scientific articles", ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM, San Diego, U.S.A, 2011, pp. 448-456.

[39] H. J. Duan, "Matrix Factorization Techniques with Temporal Dynamics for Recommender Systems," Microcomputer Applications, 2013.

[40] Z. J. Zhang, and H. Liu, "Application and Research of Improved Probability Matrix Factorization Techniques in Collaborative Filtering. Int J Control Autom", International Journal of Control & Automation, 2014, pp. 79-92.

[41] S. Purushotham, Y. Liu, and C. C. J. Kuo, "Context-aware collaborative topic regression with social matrix factorization for recommender systems", Palo Alto California AAAI Press, 2014, pp. 239-242.

[42] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional Matrix Factorization for Document Context-Aware Recommendation", ACM Conference on Recommender Systems ACM, Boston, MA, USA, 2016, pp. 233-240.