IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 6, 2019, accepted March 11, 2019, date of publication March 18, 2019, date of current version April 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2905534

Deep Auto Encoder Model With Convolutional Text Networks for Video Recommendation

WENJIE YAN, DONG WANG, MENGJING CAO, AND JING LIU
School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China
Hebei Province Key Laboratory of Big Data Calculation, Tianjin 300401, China

Corresponding author: Wenjie Yan (wenjieyanhit@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702157, in part by the Natural Science Foundation of Tianjin City of China under Grant 16JCQNJC00400, and in part by the Science and Technology Program of Tianjin City of China under Grant 15ZXHLGX00130.

ABSTRACT Collaborative filtering (CF) approach has been successfully used in recommender system (RS). Sparsity and cold start are two common phenomena in the CF algorithms nearly for each data set. Hence, these drawbacks of the classical CF algorithms have limited the recommendation performance. Deep learning theory is a very useful tool to mine the latent features in many scientific areas, such as image processing, video processing, and signal processing. In this paper, a novel deep learning-based recommendation model is introduced to solve the sparsity and cold start recommendation problems by mining the auxiliary data of users' viewing behavior datasets (e.g., the user attribute features information and video item attribute features information) and to deeply mine the latent information and their correlations of the user features and item features. First of all, the user features and video item features are processed and deeply mined by the data preprocessing layer, embedding dense layer, convolution network layer, share layer, and the auto encoder layer of our proposed model. After that, the final predictive rating process is conducted in multi-layer perception by combining with the target rating vector data and the processed user and item feature data, which is deeply mined by the above-mentioned submodels of our proposed algorithm model. The extensive experiments have shown the benefits of the proposed algorithm in the measure of mean absolute error (MAE) and root mean square error (RMSE) compared with the state-of-the-art algorithms. Besides, the impact of choices of different components and parameters of our proposed algorithm model is also studied thoroughly.

INDEX TERMS Deep auto encoder, convolutional text networks, video recommendation.

I. INTRODUCTION

Recommender system(RS) has played an utmost role in internet era to solve the information overload problem. With the rapid development of internet technology and commercial business, the data size of the internet is becoming more and more huge. The massive data has lead to the severe information overload problem on the internet. Recommender systems have proven to be an effective way to address the internet information overload problem. The recommender system can effectively provide the users with valuable information on movie, music, shopping goods and news. Many big companies such as Amazon, Alibaba, Google and Netflix have successfully adopted recommender systems to analyze the potential preferences of the customers and recommend the relevant service, products and items to the target users. The recommendation algorithms are played an utmost role in the recommender system and directly determine the results and performance of the system. Existing classical algorithms for RS can be roughly categorized into two classes [1]: content-based methods [2] and collaborative filtering methods [3]. Content-based collaborative filtering algorithms make recommendations to the users by mining the descriptions of items and user preferences. These kinds of algorithms can still provide the accurate recommendations even if the rating matrix is highly sparse. However, These methods have some drawbacks. The content-based collaborative filtering algorithms cannot make the accurate recommendations if the content analyzed for the target item does not contain useful information for recommendation. The collaborative filtering algorithms have been widely used for recommendation system by recommending items to a given user through

The associate editor coordinating the review of this manuscript and approving it for publication was Yin Zhang.

considering other users' explicit ratings on their co-rated items. One big problem is that the insufficient set of user interactions(e.g., new user/item added, or set of co-rated items is small) in collaborative filtering algorithms can lead to the bad recommendation results even cannot be recommended, which are also known as the cold start and sparsity problems [4].

To tackle the existed problems, researchers have found that the additional information about the users or items, which is known as the side information, can be helpful to the recommendation models. The side information can be obtained from the user/item profiles. E.g., demographics of a user, genres and titles of a video, etc. The user demographics can be used to measure the similarity among the users. Similarly, the recognition of the similarity of item features could be considered as the important foundation to predict ratings to the new items. The use of side information to train the recommendation model has been proven to be an efficient method, however, only use the side information as the regularization part to train the recommendation model may not be very effective because of the sparse nature of the ratings and also the side information.

One of the powerful approaches to mine the latent features of users and items that has appeared in the recent years is deep learning theory. The deep learning theory has attracted lots of attention according to its outstanding performance to mining the latent representations on various areas. Deep neural networks have been proven to achieve state of the art performance in computer vision [5], [6], natural language processing [7] and speech recognition [8]. The application of deep learning theory in recommendation systems is a new direction and attempt. The latent features are learned in a supervised or unsupervised learning manner in the deep learning models. Previous researchers have paid more attentions to directly make deep learning algorithms into the recommendation models like Restricted Botzmann Machines(RBM) [9], [10] or Multi-layer Perceptron(MLP) [11], [12] or Convolutional Neural Network(CNN). In this paper, we propose an efficient deep learning based video recommendation model to solve the sparsity and cold start problems in video recommendations. First, the side information of the users and items are input as the initial latent features. Second, the deep auto encoder and CNN model are constructed to train the proposed model by analyzing the side information of users and items as while as the explicit rating matrix. Third, the optimized prediction model is constructed after the training process is finished. Finally, the recommendation list could be created by running the optimized prediction model. In summary, the main contributions of our paper are demonstrated as follows.

First, for the sparsity problem of video rating data, this paper proposes an encoding method based on embedding word vector. Most papers currently use a one-hot encoding method to process data. However, due to the high dimensional and sparse nature of video rating data, the one-hot encoding method does not better represent the characteristics of its original data. The word embedding used in this paper makes the similarity between words and words easier to characterize, so its generalization ability is stronger than other methods.

Second, aiming at the problems of many parameters in traditional neural networks and the inherent shortcomings of single networks, this paper proposes to apply convolutional neural networks to the processing of the video related text data, thus achieving the purpose of reducing the computational complexity of model parameters by parameter sharing. On this basis, it effectively combines the unsupervised learning method of deep auto encoding, so that it can mine deeper into the hidden feature information of video users/items.

Third, in view of the shortcomings of cold start recommendation problem and insufficient use of additional data information of original video users/items, this paper proposes to use all attribute features of users and items as important feature input parameters of the model, so as to better assist video rating information for collaborative training of the predictive model of video ratings. In summary, the model can make full use of the useful information of video data to study the relationship between data and implicit feature information, thus effectively improving the accuracy of the model's ratings prediction. More importantly, the cold start recommendation problem is solved successfully.

The remain of this paper is organized as follows: In Section II, we introduce some related topics of our model. The detail description of our model is provided in Section III. Section IV contains the extensive experiments and analysis. We give a brief conclusion in Section V.

II. RELATED WORKS

Recent studies have demonstrated the effectiveness of applying neural network methods to recommendation systems, but most published papers combine a single neural network model with traditional recommendation algorithms or apply traditional algorithmic ideas to certain neural network for the recommendation sorting or rating prediction. Zhang *et al.* [13] proposed the Auto SVD ++ algorithm, which uses the video data features learned by shrinking auto encoder and the implicit feedback captured by SVD ++ to improve the recommendation accuracy of the algorithm model. Xue *et al.* [14] proposed a depth matrix decomposition model. The traditional matrix decomposition algorithm is used to decompose the user feature matrix and the item feature matrix, and then the multi-layer feed forward neural network is used to deeply mine the corresponding data features, and finally the inner product of the corresponding low-dimensional feature vector is the predicted rating of the algorithm model. The above two types of algorithms are classical algorithm models that combine traditional recommendation algorithms with neural network methods. The following are some of the topics closely related to our recommendation model: In SectionII-A, recommendation algorithms based on convolutional neural networks(CNN) are introduced. Recommendation algorithms based on deep neural networks(DNN) are proposed in Section II-B. Recommendation algorithms based on deep auto encoder(DAE) networks are demonstrated in Section II-C. We discuss the current research status about them in the below.

A. RECOMMENDATION ALGORITHMS BASED ON CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural network has become the focus of research in various popular fields due to its nonlinear mapping and high parallel processing capabilities. The advantage is that it is suitable for processing data with similar local attributes, so it has high research and application value in data feature extraction and prediction. The particularity of convolutional neural networks is manifested in two aspects: the neurons in the neural network are only partially connected, and some of the neurons share weights. Its weight sharing feature reduces the complexity of the network model and reduces the total number of weights in the neural network. This feature avoids the complex feature extraction and data reconstruction process in traditional neural networks when the network extracts multidimensional data. For example, Zheng *et al.* [15] use the advantages of convolutional neural networks to process text features of users and items to improve training speed, thereby improving the prediction accuracy of the algorithm. Wu *et al.* [16] used the content embedding method to obtain the corresponding text features, and then input the convolutional neural network model to obtain fixed-size feature data, and finally obtain the prediction result.

B. RECOMMENDATION ALGORITHMS BASED ON DEEP NEURAL NETWORKS

The essence of deep neural network is to mine the characteristics of input data through multiple layers of hidden layers, thus effectively improving the accuracy of algorithm prediction results. The method extracts the implicit features of the corresponding data on the basis of the traditional recommendation algorithm, or directly inputs the original data into the deep neural network model after data processing, so that the implicit feature information can be effectively improved after extracting the implicit feature information. Hence, the prediction accuracy of the algorithm can be improved. Zhang *et al.* [17] proposed a method of deep mining and integrating user and item features using DNN method to improve the accuracy of algorithm prediction rating. He and Chua [18] proposed that after the pooling operation of the original data features, all the data features are combined into one vector representation, and then the high-order data features are learned through the multi-layer fully connected neural network. The implicit interaction relationship can effectively improve the accuracy of prediction of the algorithm model. In addition, Guo *et al.* [19] proposed the fusion of factorization-machine and feed-forward neural network. The feed-forward neural network learns the high-order interaction features when dealing with the corresponding attribute features, so that the high-dimensional sparse data features can effectively improve the accuracy of feature representation after reducing the data dimension. In the end, the prediction accuracy of the model has been effectively improved.

C. RECOMMENDATION ALGORITHMS BASED ON DEEP AUTO ENCODER

Auto encoder (AE) is an important structure in the deep learning model. Its ability to learn hidden features has been recognized by many researchers. The model that first applied the automatic encoder to the recommendation algorithm is the auto encoder-based collaborative filter (ACF) proposed by Yuan *et al.* [20] in 2014. The ACF algorithm divides the user's rating value for the item into five vectors. However, the ACF model has the following two shortcomings: First, ACF can only solve the integer scoring prediction problem. Second, subdividing the user's rating value into five vectors increases the sparsity of the scoring matrix, which reduces the accuracy of the scoring prediction of the algorithm. In addition, the most typical auto encoder model is represented by AutoRec [21]. The AutoRec model respectively uses the row vector and column vector in the user rating matrix as the user vector and the item vector, and uses it as the input data for training the automatic encoder model. The core purpose of the algorithm is to reconstruct the original input data. Although the AutoRec model can solve the problem of non-integer scoring value prediction, it does not add noise to the input, which makes the algorithm less robust and the algorithm is prone to over-fitting. The above models belong to the scoring prediction model, and CDAE [22] is used to make the ranking prediction model. The input to the model is the user's implicit feedback data for the item. Specifically, each node of the model input portion corresponds to an item, and can also be regarded as a user's preference for the item's interest. The user's preference for an item is represented by a 0-1 value. Finally, the items corresponding to the predicted values of the output layer nodes in the model are sequentially recommended to the user. The disadvantage of the above two methods is that there is a cold start problem. A CFN [23] model combining content information and a scoring matrix then appeared. The recommendation accuracy of the algorithm is improved compared to the previous method. The disadvantage is that the content information is relatively simple and the data is very sparse.

III. ALGORITHM MODEL AND ANALYSIS

In this section, the detail information about the hybrid collaborative filtering recommendation model for fusion auto encoder and convolutional neural networks will be demonstrated clearly. The architecture of our proposed algorithm is shown as Fig.1. First, the user features and video item features are processed by the data processing layer of our model. Second, the original high-dimensional sparse feature data is processed by the embedding layer and the implicit relationship between the video user and the attribute data of the video item is mined. Third, the convolutional layer operations in this model are used to process text data for video

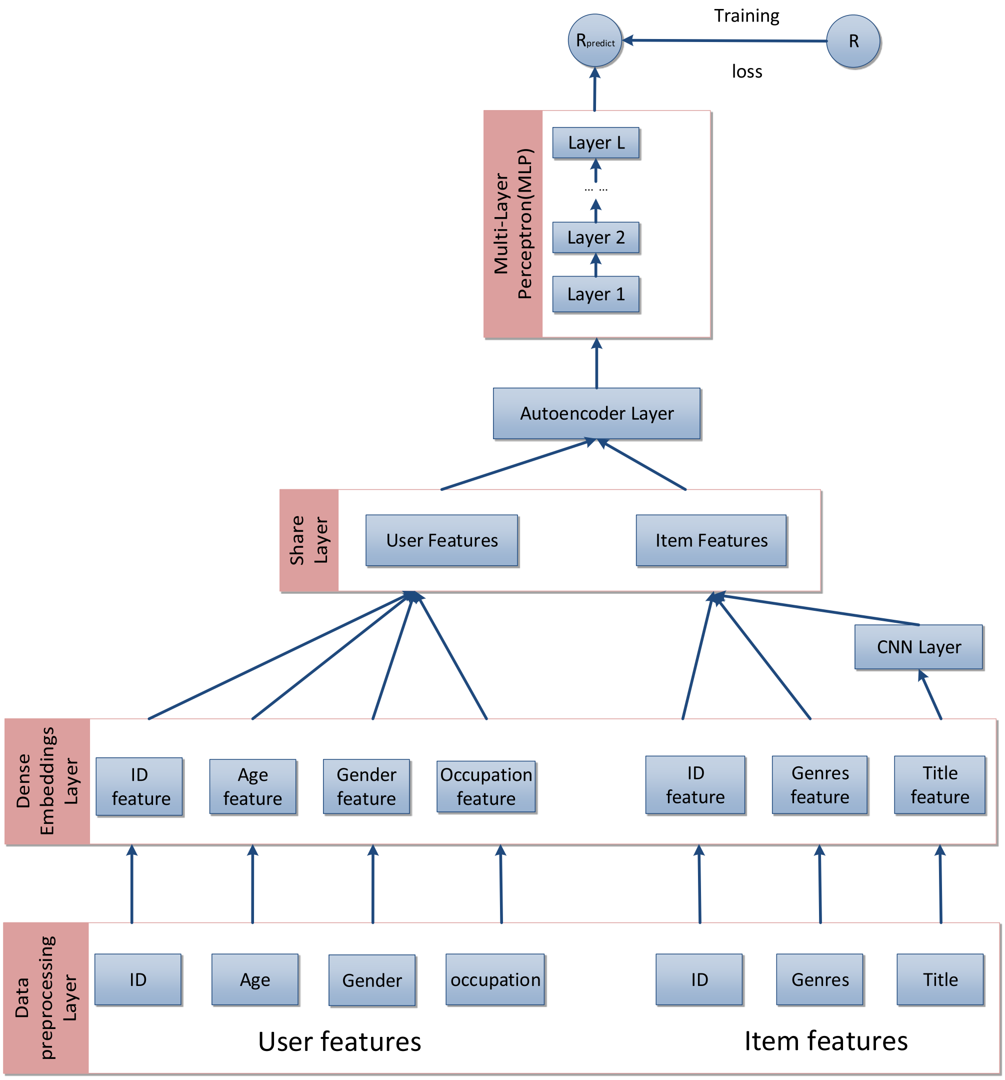


FIGURE 1. The architecture of the proposed deep learning based recommendation model.

name features. Forth, the characteristics of the video user and the features of the video item are combined into a unified feature matrix in the share layer. Fifth, the auto encoder layer can deeply mine the implicit attribute features of the user/item and the correlations between the two. Finally, the original high-dimensional video user/item attribute feature matrix is mapped into a vector of the same dimension as the target rating vector in multi-layer perceptron layer, and then the final predictive rating process is conducted by combining with the target rating vector data. The following are the submodels involved in the proposed algorithm model.

A. DATA PREPROCESSING

The MovieLens dataset has been used to implement the proposed algorithm model. The algorithm model proposed in this paper combines the additional information of movie users/items with the user's rating matrix, and can cooperate with the training rating prediction model to achieve the purpose of improving the accuracy of rating prediction. Therefore, the pre-processing of additional information for users/items is very important.

The attribute characteristics of the movie user mainly include the ID feature of the user, the age feature of the user, the gender feature of the user, and the occupational characteristics of the user. The ID data of the movie user is relatively simple, and is represented by a variable e_1. We replace the age of the movie user with 7 integer numbers [0-6]. The more mathematical representation of this method is demonstrated as Equation (1).

$$e_2 = \sigma_2(X_1, \cdots, X_n) \tag{1}$$

Here e_2 represents the processed age feature data, and $\sigma_2(.)$ represents the mapping function. $X_1 \sim X_n$ represent the original data representation of the age characteristics.

The gender characteristics of the movie user are also relatively simple. Here, the gender feature of the user is represented by an integer value of 0 and 1. The more mathematical representation of this method is in Equation (2).

$$e_3 = \sigma_3(F, M) \tag{2}$$

e_3 represents gender feature data, $\sigma_3(.)$ represents the mapping function, and F, M represents female and male users, respectively.

Since the user's professional feature data is text data, it is also needed to convert into an integer data representation. The more mathematical representation of this method is in Equation (3).

$$e_6 = \sigma_6(S_1, \cdots, S_n) \tag{3}$$

e_6 represents the processed user's occupational feature data representation, and $\sigma_6(.)$ represents a mapping function, which maps 21 professional feature data of the movie user to 21 numerical data representations, which is 0 to 20.

The attribute characteristics of the movie item mainly include the ID feature of the movie item, the name feature of the movie item, and the type characteristics of the item. The ID feature data of the movie item is relatively simple, and is represented by a variable e_7. For the name feature of the movie items, the year after the name is first filtered out and then stored as a dictionary. Finally, as with the movie type feature, each movie name is represented by a list of fixed length of 15. The more mathematical representation of this method is in Equation (4).

$$e_5 = g(\sigma_5(f(Title_1, \cdots, Title_n))) \tag{4}$$

e_5 represents the data representation of the processed movie name, $f(.)$ represents the function for filtering the movie name, and $\sigma_5(.)$ represents the conversion function for converting the text data to the corresponding index value. $g(.)$ indicates that the previously processed feature data is represented by a vector having a fixed length of 15.

Further, the remaining user ID feature data is represented by e_1. For the type feature of a movie, since it is text data, it is converted into numerical data. There are a total of 19 movie type features here, so we use a mapping function to map each movie type feature to a list of length 19. The more mathematical representation of this method is in Equation (5).

$$e_4 = s(\sigma_4(Style_1, \cdots, Style_n)) \tag{5}$$

e_4 represents the feature data representation of the movie type after mapping processing, σ_4 representing the mapping function, converting the text data into numerical data, and $s(.)$ representing the processing function, which processes the mapped data into a fixed-length vector representation. In summary of the above description, all attribute characteristics of the movie user/item in the model can be described in detail in the following Fig.2.

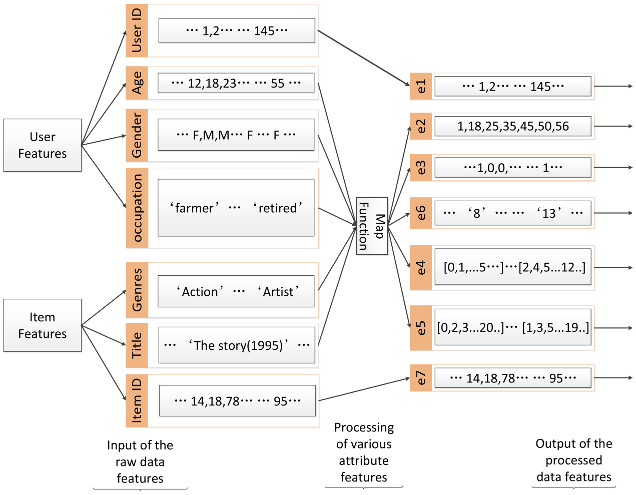


FIGURE 2. The whole attributes features of the movie users and items.

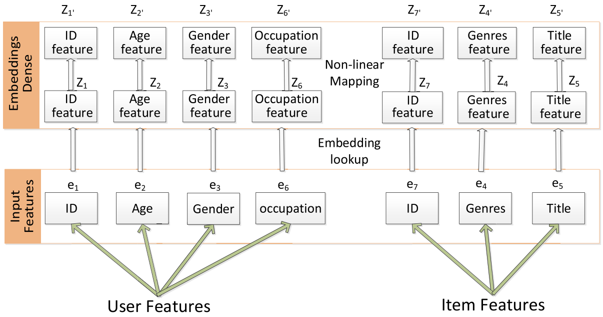


FIGURE 3. Data of user and item attributes processed through embedding and dense operation.

B. EMBEDDING DENSE LAYER

After the above data preprocessing, all attribute feature data of the movie user/item becomes a representation of the shaped data. The model of the embedding dense layer for the input form of all attribute feature data is shown as Fig.3.

First, after the original high-dimensional sparse feature data is processed by the embedding layer, its dimensions are reduced, and it is processed into a predetermined fixed-size vector form. Its mathematical representation is in Equation (6).

$$Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7 = f(e_1, e_2, e_3, e_4, e_5, e_6, e_7) \tag{6}$$

Here, $f(e_1, e_2, e_3, e_4, e_5, e_6, e_7) = \phi(e_1) \otimes \phi(e_2) \otimes \phi(e_3) \otimes \phi(e_4) \otimes \phi(e_5) \otimes \phi(e_6) \otimes \phi(e_7)$, and $\phi(.)$ indicates that the attribute feature vectors $e1$ to $e7$ are processed into functions using the dimensional representation of the C-dimension, and the meanings of $\phi(e_4)$ and $\phi(e_7)$ represent the mean values of their matrix data, and their dimensions are not changed. $\otimes$ indicates its corresponding conversion method.

Second, after a layer of nonlinear function dense operation(full connection layer), the dimensions of the output feature matrix with inconsistent original dimensions are unified into the dimensional representations suitable for various attribute features, so as to facilitate the subsequent prediction and rating module. Its mathematical representation is in

Equation (7).

$$\begin{aligned} Z_1' &= \psi_1(w_1 Z_1 + b_1) \\ Z_2' &= \psi_2(w_2 Z_2 + b_2) \\ &\cdots\cdots\cdots\cdots \\ Z_L' &= \psi_L(w_L Z_L + b_L) \end{aligned} \tag{7}$$

Here $\psi_L(.)$ is a nonlinear processing function, and the subscript L belongs to [1 ……N]. Similarly, w_L represents the weighting feature of the corresponding function, and Z_L represents the embedded representation matrix of various features after processing. b_L indicates its corresponding deviation.

This section mainly deals with the various attributes of the movie item, and mines the implicit relationship between the item user and the attribute data of the movie item. In general, the reason we use the embedding method instead of the one-hot method is as follows. On the one hand, the vector dimension encoded by the One-hot method is very high and the data is sparse. Suppose we have encountered a dictionary of 2000 words in Natural Language Processing(NLP). When coded using the One-hot method, each word is represented by a vector containing 2,000 integers, where the 1999 number is zero. The word embedding dimension in the embedding method can be pre-set without using high-dimensional sparse data representation like one-hot encoding. On the other hand, in the process of training the neural network using the embedding method, each embedded vector is updated. The embedding method can dig deeper into the similarity between words and words in multidimensional space. Not only that, any content that is converted to a vector by embedding the embedding layer can do so. For example, the attribute characteristics of movie users/items in this model.

C. CNN LAYER

The convolutional layer operations in this model are used to process text data for movie name features. The movie name data obtained through the previous data preprocessing stage is a fixed list of length 15. The convolutional layer in this model contains both convolution and pooling operations. Here we use a window of size 2, 3, 4, 5 to perform convolutional operations on movie name features. First, a new feature matrix for the movie item name attribute can be generated by the convolution operation. Its mathematical representation is in Equation (8).

$$C_j = F(DV_j + b_j) \tag{8}$$

Here, D represents the name feature e_5 of the movie item after the above Embedding process. V_j denotes its corresponding convolution kernel. b_j represents the deviation of its corresponding convolution kernel. $F(.)$ is a nonlinear function, such as *Sigmoid* or *ReLu* activation functions. Second, the following data can be obtained by performing a pooling operation on the generated C_j feature matrix. The mathematical representation is in Equation (9).

$$B_j = g(C_1, \cdots, C_j) \tag{9}$$

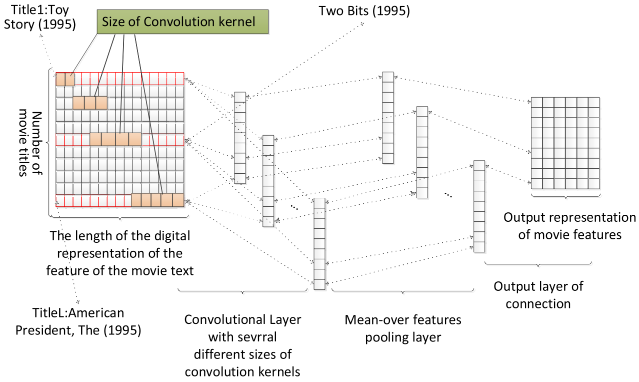

Title1:Toy Story (1995)
Size of Convolution kernel
Two Bits (1995)
Number of movie titles
The length of the digital representation of the feature of the movie text
TitleL:American President, The (1995)
Convolutional Layer with sevrral different sizes of convolution kernels
Mean-over features pooling layer
Output layer of connection
Output representation of movie features

FIGURE 4. Title attributes of the movie processed by convolutional text networks.

where $g(.)$ represents the pooling function, generally has the $maxpool(.)$ function and the $avgpool(.)$ function. Here we use the $avgpool(.)$ average pooling operation, which is used to more deeply explore the implicit attributes of the movie name feature. Specifically, the convolutional layer model performs a convolution operation by using a window with a loop operation size of 2, 3, 4, and 5, and finally connects the obtained four results using the list. The method is demonstrated in Equation (10).

$$B = [B_1, B_2, \cdots, B_j] \tag{10}$$

where B is the result of the feature matrix e_5 of the movie name after the convolution operation. All convolutional layer operations described above can be represented by a uniform function. The specific mathematical representation is in Equation (11).

$$B_j = CNN(W, X_j) \tag{11}$$

W represents the total weight, and X_j represents all the feature representations of the movie name e_5. B_j represents the implicit (potential) feature representation of the movie name after each convolution layer operation.

In summary, this section focuses on the further processing of the name attribute of the movie item. The essence of this module is a convolutional neural network model. First, the module extracts attribute characteristics of different movie item names through deep convolution operations. Second, the size of the dimensional representation of the input data is reduced by subsequent pooling operations. Finally, the name characteristics of the movie items are further explored in this module. The structure of this model is demonstrated in Fig.4

D. SHARE LAYER

After processing the previous modules, we obtained the implicit attribute characteristics of the movie user/item. Next we combine the characteristics of the movie user and the features of the movie item into a unified feature matrix. For the feature matrix of the movie user, the specific merge method is as follows. Its mathematical expression is in Equation (12).

$$U = I_U(M_U(Cat(K_{U_1}, K_{U_2}, K_{U_3}, K_{U_4})) + b_U) \tag{12}$$

Wherein, $K_{U_1}, K_{U_2}, K_{U_3}, K_{U_4}$ respectively represent representations of attribute characteristics of UserID, Age, Gender, and Occupation after the previous step processing. $Cat(.)$ is a connection function that changes in the dimension direction. After connecting the previous four feature vectors, it outputs a feature matrix of fixed dimension values through a hidden layer(shared connection layer). $I_U(.)$ represents some kind of activation function, such as *ReLu* or *Sigmod* function. M_U stands for the weight between the upper and lower layers of the connection network. b_U stands for its corresponding deviation. U represents the final output, which is the attribute representation of the user. Similarly, the attribute characteristics of the movie item have to be similarly processed. The difference is that there are only three attribute features of the movie items, and the method is in Equation (13).

$$V = I_V(M_V(Cat(K_{V_1}, K_{V_2}, K_{V_3})) + b_V) \tag{13}$$

Among them, $K_{V_1}, K_{V_2}, K_{V_3}$ is the result representation of the item attribute feature after the previous step processing. M_V represents the weight between the upper and lower fully connected layers, and b_V represents the bias value. $Cat(.)$ is a connection function that changes in the dimension direction. After connecting the first three vectors, it outputs a feature matrix of a fixed dimension value through a hidden layer(shared connection layer). $I_V(.)$ represents some kind of activation function, such as *ReLu* or *Sigmod* function. After obtaining the attribute feature matrix of the user and the item, the first two feature matrices are merged into a unified matrix representation by the following function. The method is in Equation (14).

$$G = Concatenate(U, V) \tag{14}$$

U is the feature matrix of the user attribute obtained earlier, and V is the feature matrix of the item attribute. $Concatenate(.)$ represents the connection function. G represents the joint feature matrix of the resulting movie user/item.

This section is a shared connection layer with a total of three connection layers (movie user feature connection layer, movie item feature connection layer, and full connection layer). First, the Equation (12) represents the working principle of the user feature connection layer. Through the connection layer, various feature attributes of the user are merged into a representation of a feature matrix. Second, Equation (13) represents the movie item connection layer, which is capable of merging various attribute features of a movie item into a representation of a feature matrix. Finally, as described in Equation(14), the fully connected layer connects the previously obtained user attribute feature matrix and the attribute item matrix of the movie item into a more complete matrix representation, thereby facilitating the deeper auto encoder network to be deeper. The whole graphical fusion structure of this model is demonstrated in Fig.5.

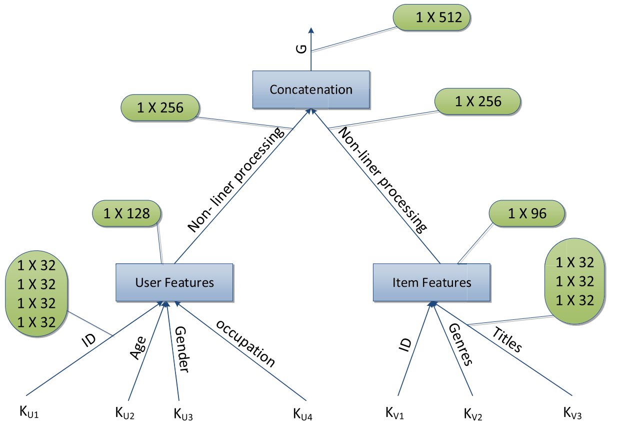


FIGURE 5. Feature matrix fusion of movie user/item by using share layer.

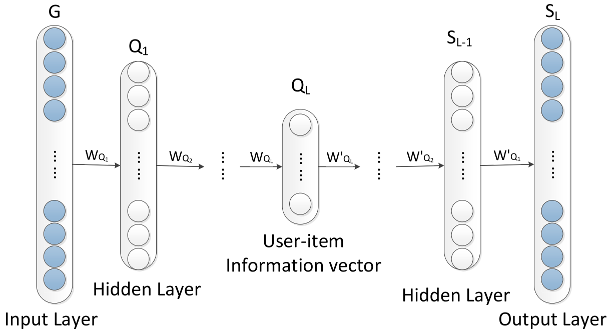


FIGURE 6. Feature matrix of user/item deeply mined by using the deep auto encoder layer.

Besides, the number indicates the dimension of the user features and item features and the dimension of user-item features concatenation after non-linear processing in the share layer. The above features with the proposed dimensions can achieve the best performance through extensive experiments.

E. AUTO ENCODER LAYER

After the output feature matrix G is obtained, the processing of the deep auto encoder can more deeply mine the implicit attribute features of the user/item and the correlation between the two. The following is an example representation of the deep auto encoder module of our proposed model in Fig.6.

Its mathematical representation is as follows in the coding phase Equation (15) and decoding phase Equation (16).

$$\begin{aligned}
Q_1 &= f(W_{Q_1}G + b_1) \\
Q_2 &= f(W_{Q_2}Q_1 + b_2) \\
&\cdots\cdots\cdots \\
Q_L &= f(W_{Q_L}Q_{L-1} + b_L) && (15) \\
S_1 &= f(W'_{Q_L}Q_L + d_1) \\
S_2 &= f(W'_{Q_{L-1}}S_1 + d_2) \\
&\cdots\cdots\cdots \\
S_{L-1} &= f(W'_{Q_2}S_{L-2} + d_{L-1}) \\
S_L &= f(W'_{Q_1}S_{L-1} + d_L) && (16)
\end{aligned}$$

The generated output layer vector S_L is the attribute of the video user/item after being processed by the deep auto encoder model. Here L represents the number of network layers of the corresponding deep auto encoder model. W'_{QL} is the weight generated after the transposition process of W_{QL}. This kind of processing can effectively reduce the number of parameters in the encoding and decoding stages in the network.

This section is a network structure of a deep auto encoder model. It mainly performs nonlinear dimensionality reduction processing on the joint attribute feature matrix of movie users and movie items processed by the shared layer. Under the premise of maintaining the attribute information of the movie user and the movie item, the nonlinear mapping processing of the deep auto encoder can further optimize the representation of the attribute features of the movie user and the movie item.

F. MULTI-LAYER PERCEPTION

As shown, here is a multi-layer feed forward neural network named multi-layer perceptron(MLP). After the processing of the previous deep auto encoder network module, the implicit correlation between movie user/item attribute features has been described more accurately. Therefore, after processing through a MLP network with full connectivity as shown in the following Fig.7, the original high-dimensional movie user/item attribute feature matrix is mapped to a vector of the same dimension as the target rating vector R, and then combined with the original target rating vector data enables the final predictive rating process. Since the MLP network model has full connection characteristics, the attribute characteristics of the movie user/item input by the network can be maintained between the users/items excavated by the deep auto encoder network layer after the dimensionality reduction processing. Its mathematical expression is as follows in Equation (17).

$$
\begin{aligned}
B_1 &= G(W_1 S_L + K_1)\\
B_2 &= G(W_2 B_1 + K_2)\\
&\cdots\cdots\cdot\\
B_L &= G(W_L B_{L-1} + K_L) \qquad (17)
\end{aligned}
$$

Among them, $G(.)$ represents a non-linear activation function, here can be *ReLu* or *Sigmod* function. The output value B_L of the multi-layer perceptron network model is equal to $\hat{R}$, which is the predicted rating vector value of the network model. The processing of the hidden layer can maintain the attribute relationship between the movie user and the movie item. Through the principle of nonlinear mapping of multiple hidden layers, the model can more accurately obtain the prediction rating information of movie users for movie items. The whole structure of this model is in Fig.7 The processing of the hidden layer can maintain the attribute relationship between the movie user and the movie item. Through the principle of nonlinear mapping of multiple hidden layers, the model can more accurately obtain the prediction rating information of movie users for the movie items.

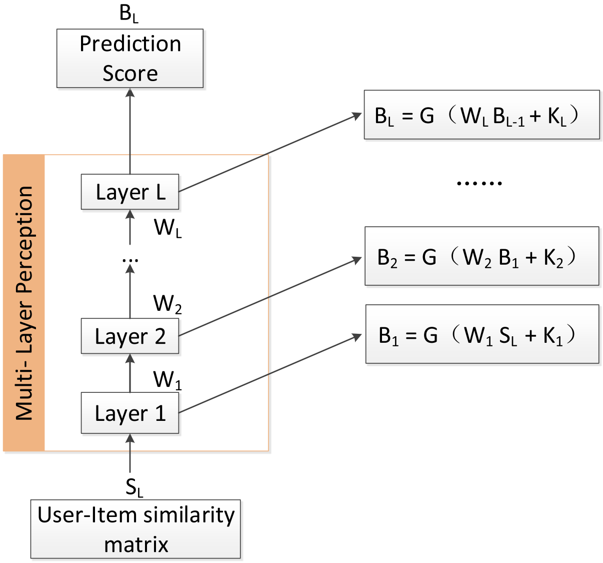

B$_L$
Prediction Score
Multi- Layer Perception
Layer L
W$_L$
...
W$_2$
Layer 2
W$_1$
Layer 1
S$_L$
User-Item similarity matrix
B$_L$ = G（W$_L$ B$_{L-1}$ + K$_L$）
......
B$_2$ = G（W$_2$ B$_1$ + K$_2$）
B$_1$ = G（W$_1$ S$_L$ + K$_1$）

FIGURE 7. **Predicted rating model deeply mined by multi-layer perception networks.**

G. LOSS FUNCTION

Here the predicted loss function defined by the rating model is in Equation (18):

$$\arg \min_{W,V,V',b,c,c'} \sum_{R\in\Omega} \|\hat{R} - R\|_2^2 + l(W, V, V', b, c, c') \qquad (18)$$

Wherein, $l(.) = \lambda/2(\| \ W \ \|_2^2 + \| \ b \ \|_2^2 + \| \ V \ \|_2^2 + \| \ V' \ \|_2^2 + \| \ c \ \|_2^2 + \| \ c' \ \|_2^2)$ The parameter values in the model are continuously learned by *Adam* algorithm though minimizing the loss function. $l(.)$ is a regularization function that controls the complexity of the model. Where W represents the sum of the weight parameters of the last fully connected network in the model, and b represents the sum of the corresponding deviations. V represents the sum of the weights of all coding stages in the deep auto encoder network layer, and the corresponding c represents the sum of all the deviations of the coding stage. V represents the sum of the weights of all decoding stages in the deep auto encoder network layer, and the corresponding c' represents the sum of all the deviations of the decoding stage. R represents the true train or test rating matrix, and Ω represents the whole initial input rating matrix.

This module is part of the loss function of the model presented in this paper. By optimizing the loss function, the model is able to derive predictive ratings that are closer to the true rating information. In addition, the loss function uses the L_2 regularization method to further prevent the overfitting phenomenon. At the same time, the model also perfects the L_2 regularization method, and adds auxiliary features such as bias terms b, c and c', so as to improve the accuracy of the prediction model of the algorithm model under the premise of improving the operating efficiency of the entire network model.

IV. EXPERIMENTS AND ANALYSIS

A. DATA DESCRIPTION

Two real datasets are used to test the performance of our model and other models: MovieLens-1M and MovieLens-100K. The MovieLens-1M dataset downloaded from the MovieLens website, and it contains 1000209 rating records from 6040 users rated for 3952 movies. In addition, the number of ratings per use is about 165.6 and the number of ratings per item is about 253.09. The rating sparsity of MovieLens-1M dataset is 95.81%.

The MovieLens-100K dataset is also downloaded from the MovieLens website, but it contains only 100000 rating records from 943 users rated for 1682 movies. In addition, the number of ratings per use is about 106.4 and the number of ratings per item is about 59.45. The rating sparsity of MovieLens-1M dataset is 93.7%.

The user side information of user ID, age, gender and occupation are include in both of the MovieLens-1M and MovieLens-100K data set. Similarly, the item side information of movie ID, title and genres are include in both of the above two datasets too. The data sets used in our experiments is demonstrated as follows in Table 1.

B. EXPERIMENTAL ENVIRONMENT

In this section, the extensive experiments are conducted to verify the prior performance of our proposed model.

1) HARDWARE

All the experiments are conducted on Intell(R)W-2123 CPU@ 3.6GHz, RAM 32GB, and GPU NVIDIA GTX1080Ti platform.

2) SOFTWARE

The operating system used in this experiment is Ubuntu 16.04, and the Python language is used to achieve the program, the specific software version is 3.5. Since the proposed model is based on deep learning theory, Tensorflow 1.4.0 is used to implement the deep learning module.

C. EVALUATION METRIC

The Mean Absolute Error(MAE) and Root Mean Square Error(RMSE) are used to evaluate the prediction performance of all the mentioned algorithms. In detail, the definition of MAE and RMSE are stated in Equation (19) and (20) respectively.

$$MAE = \frac{\sum_{(u,i)\in R}(R_{u,i} - \hat{R}_{u,i})}{|R|} \tag{19}$$

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in R}(R_{u,i} - \hat{R}_{u,i})^2}{|R|}} \tag{20}$$

where R denotes the whole rating matrix, $R_{u,i}$ denotes the rating user u gives to item i, and $\hat{R}_{u,i}$ denotes the rating user u gives to item i as prediction. Smaller values of MAE and RMSE mean better performance.

TABLE 1. Data set descriptions on movieLens-100K and movieLens-1M.

Data Set	MovieLens-100K			MovieLens-1M
No.of users	943			6040
No.of items	1682			3952
No.of ratings	100000			1000209
No.of ratings per user	106.4			165.6
No.of ratings per item	59.45			253.09
Rating Sparsity	93.7%			95.81%
User Features	User ID	Age	Gender	Occupation
Item Features	Movie ID	Title	Genres	

D. COMPARED METHODS

The following famous and state-of-the-art recommendation algorithms are chosen to compare with our method. Average: This approach predicts the missing ratings by analyzing the average historical ratings of users or items, and hence, there are two variants: UserAverage and ItemAverage. top-K IBCF: Collaborative filtering algorithm based on item cluster top-K. top-K UBCF: Collaborative filtering algorithm based on user cluster top-K. SlopeOne: It is an efficient online rating-based collaborative filtering approach, which precompute the average difference between the ratings of one item and another for users who rated both. PMF [24]: Probabilistic Matrix Factorization for recommendation. BPMF [25]: Bayesian Probabilistic Matrix Factorization (BPMF) for recommendation. PRA [26]: Probabilistic Rating Auto-encoder, which uses autoencoder to generate latent user feature profiles. NRR [27]: NRR is a neural rating regression model which captures the user/item specific characteristics. RMB-DNN [17]: A recommendation model based on deep neural network.

E. EXPERIMENTAL RESULTS

In this section, the extensive experiments are conducted to verify the superiority of our proposed model.

1) RECOMMENDATION PRECISION COMPARISONS ON DIFFERENT ALGORITHM MODELS

In this section, the recommendation precision comparisons are conducted among the state-of-the-art algorithms and the proposed algorithm model in this paper. The RMSE and MAE are as the evaluation metrics to measure the performances among the different algorithms. Table 2 shows that the proposed algorithm has the best performance whether on RMSE or MAE measure among the different algorithms. Specially, The RMSE and MAE values of the model implemented on the MovieLens-100K data set are 0.914 ± 0.002 and 0.715 ± 0.003, respectively, while the RMSE and MAE values of the model implemented on the MovieLens-1M data set are 0.859 ± 0.001 and 0.665 ± 0.003, respectively. That is to say, the proposed algorithm has the highest recommendation accuracy compared with the mentioned state-of-the-art algorithms.

TABLE 2. Prediction performance comparison by the evaluation metrics of RMSE and MAE.

Algorithms	MovieLens-100K		Algorithms	MovieLens-1M	
	MAE	RMSE		MAE	RMSE
UserAverage	0.839 ± 0.004	1.047 ± 0.004	UserAverage	0.830 ± 0.002	1.036 ± 0.002
ItemAverage	0.825 ± 0.005	1.035 ± 0.005	ItemAverage	0.782 ± 0.001	0.978 ± 0.001
topK-IBCF	0.868 ± 0.001	0.934 ± 0.002	topK-IBCF	0.775 ± 0.002	0.881 ± 0.001
topK-UBCF	0.900 ± 0.001	0.949 ± 0.001	topK-UBCF	0.827 ± 0.003	0.909 ± 0.001
SlopOne	0.737 ± 0.001	0.935 ± 0.001	SlopOne	0.710 ± 0.002	0.899 ± 0.002
PMF	0.788 ± 0.028	0.975 ± 0.030	PMF	0.684 ± 0.003	0.874 ± 0.006
BPMF	0.725 ± 0.003	0.927 ± 0.003	BPMF	0.678 ± 0.001	0.867 ± 0.002
PRA	0.759 ± 0.004	0.964 ± 0.005	PRA	0.715 ± 0.001	0.900 ± 0.001
NRR	0.717 ± 0.005	0.909 ± 0.003	NRR	0.691 ± 0.002	0.875 ± 0.002
RMB-DNN	0.696 ± 0.002	0.987 ± 0.029	RMB-DNN	0.658 ± 0.001	0.935 ± 0.015
Ours	0.715 ± 0.003	0.914 ± 0.002	Ours	0.665 ± 0.003	0.859 ± 0.001

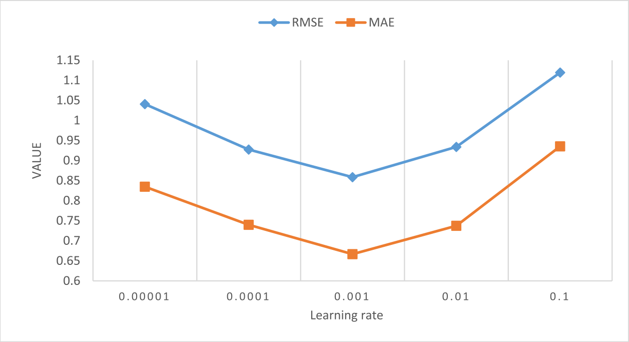


FIGURE 8. The prediction rate of the algorithm model's learning rate on the movieLens-1M dataset (RMSE and MAE).

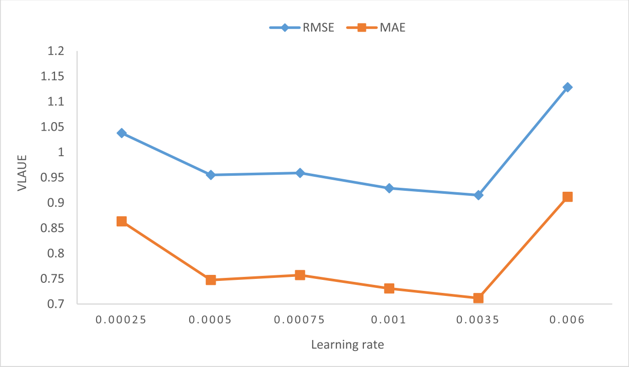


FIGURE 9. The prediction rate of the algorithm model's learning rate on the movieLens-100K dataset (RMSE and MAE).

2) MODEL LEARNING RATE ON MOVIELENS DATASET

Fig.8 reveals that as the learning rate of the algorithm model is different, the RMSE and MAE results of the final model's prediction rating will also change. From Fig.8 we can clearly see that in the MovieLens-1M data set, when the learning rate is 0.001, the test results (RMSE and MAE values) on the MovieLens-1M data set have been optimized. The performance of the algorithm is optimal.

Similarly, Fig.9 also reveals the changes in experimental measurements (RMSE and MAE) on the movieLens-100K dataset as the learning rate of the network model differs. Unlike the MovieLens-1M data set, because the data volume of the Movielens-100K data set is relatively small, a relatively large learning rate can get the best training model, so that the best RMSE and the best MAE value can be obtained in the test data set. Specifically, Fig.9 reveals that when the learning rate value is 0.0035, the RMSE and MAE values of the test data set reach a minimum value, and the performance of the model algorithm at this time is optimal.

As an important parameter of the deep learning model, learning rate plays a crucial role in the performance of the proposed model. In theory, the learning rate of the model is too small, which not only leads to the increase of the time complexity of the model algorithm, but also the overfitting phenomenon of the model. When the learning rate of the model is too large, although the model can be faster converges, but the model will be under-fitting, which will directly affect the actual performance of the model. From the actual test results of the experiment, the algorithm model does have an optimal learning rate, and the experimental results directly confirm the correctness of the theoretical analysis.

3) AUTO ENCODER AND ACTIVATION FUNCTION ON MOVIELENS DATA SET

Through a large number of reading research papers and previous experimental results, different types of activation functions do have different experimental effects for the same algorithm model. The role of the auto encoder is to mine deeper into the implicit attribute characteristics of the input data. It can be clearly seen from Fig.10 and Fig.11 that the algorithm model is affected by five different activation functions, such as *ReLu*, *swish*, *softplus*, *selu*, and *elu*, in the case of the Movielens-1M data set with or without the auto encoder model. The difference in experimental results is very obvious. We can draw the following conclusions. First, under the same experimental conditions on the Movielens-1M dataset, the addition of the deep auto encoder model does cause the RMSE and MAE values of the algorithm model prediction rating to be reduced to different degrees, so that the algorithm model is added to the deep auto encoder. The performance of the algorithm has been improved to varying degrees. Second, Fig.10 and Fig.11 reveal that under the same experimental conditions described above, the *ReLu* activation function is most effective for the performance of the algorithm model, both in terms of the RMSE of the model prediction rating and in the MAE value.

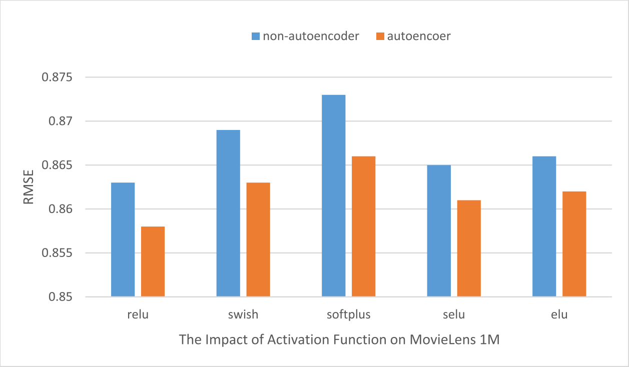


FIGURE 10. **Effect of various types of activation functions on the RMSE of test ratings in the movieLens-1M dataset with or without auto encoder layers.**

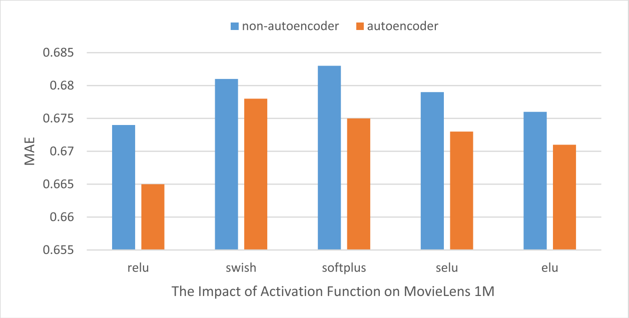


FIGURE 11. **Effect of various types of activation functions on the MAE of test ratings in the movieLens-1M dataset with or without auto encoder layers.**

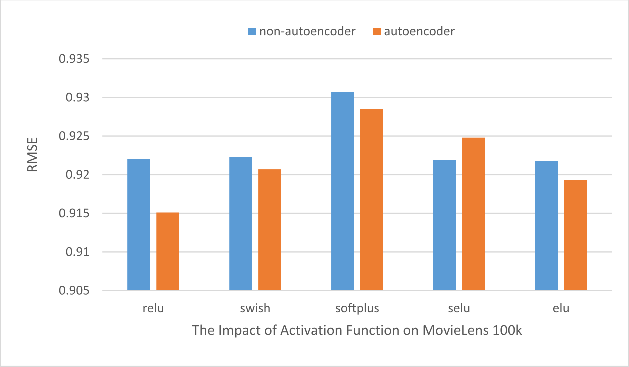


FIGURE 12. **Effect of various types of activation functions on the RMSE of test ratings in the movieLens-100K dataset with or without auto encoder layers.**

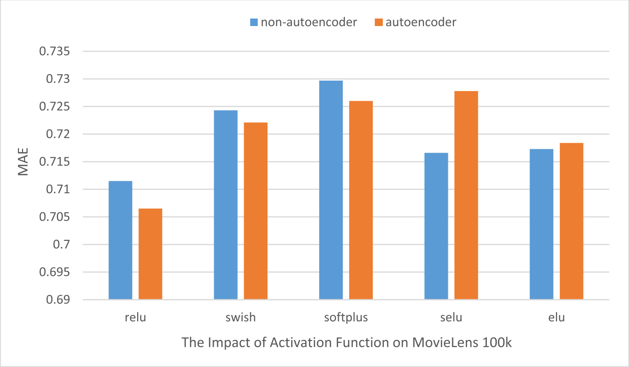


FIGURE 13. **Effect of various types of activation functions on the MAE of test ratings in the movieLens-100K dataset with or without auto encoder layers.**

It can be clearly seen from Fig.12 and Fig.13 that the algorithm model is affected by five different activation functions, such as *relu*, *swish*, *softplus*, *selu*, and *elu*, in the case of the Movielens-100K data set with or without a auto encoder model. The difference in experimental results is very obvious. Unlike the Movielens-1M dataset, Fig.12 reveals that when the activation function is *selu*, the addition of the deep auto encoder model reduces the performance of the algorithm model, thereby increasing the RMSE value of its prediction rating. For the remaining activation functions, the addition of the deep auto encoder model increases the performance of the algorithm model to varying degrees, thereby reducing the RMSE value of the prediction rating. Fig.13 reveals that when the activation functions are *selu* and *elu*, the addition of the deep auto encoder model reduces the performance of the algorithm model, thereby increasing the MAE value of the prediction rating. For the remaining activation functions, the addition of the deep auto encoder model increases the performance of the algorithm model to varying degrees, thereby reducing the MAE value of the prediction rating. We can also draw the following conclusions. First, under the same experimental conditions on the data set of Movielens-100K, the addition of deep auto encoder model in most cases makes the RMSE and MAE values of the algorithm model prediction ratings are reduced to different degrees, so that the performance of the algorithm model has been improved to varying degrees after adding a deep auto encoder model. Second, Fig.12 and Fig.13 reveal that under the same experimental conditions described above, the *ReLu* activation function is most effective for the performance of the algorithm model, both in terms of the RMSE of the model prediction rating and the MAE value.

4) DROPOUT RATIO ON MOVIELENS DATA SET

When the neural network model has a more complex network structure, the algorithm model tends to be over-fitting phenomenon. To solve this problem, researchers often use the dropout method to prevent over-fitting. The model proposed in this paper also shows different degrees of over-fitting. It can be seen from Fig.14 and Fig.15 that training different experimental data sets (such as Movielens-100K and Movielens-1M data sets) and adopting different dropout methods have different effects on the performance of the algorithm model. Specifically, Fig.14 depicts the experimental results of the data set for MovieLens-1M. When the ratio of dropout in the line graph reaches 0.5, the RMSE and MAE values of the predictive rating performance indicators

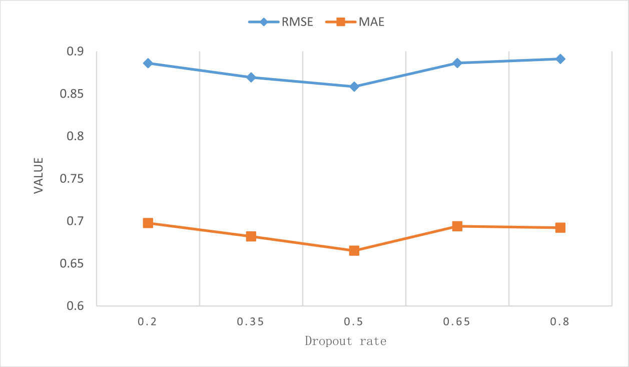


FIGURE 14. Impact of the dropout method on the prediction rating performance indicators RMSE and MAE of the algorithm model on the movielens-1M Dataset.

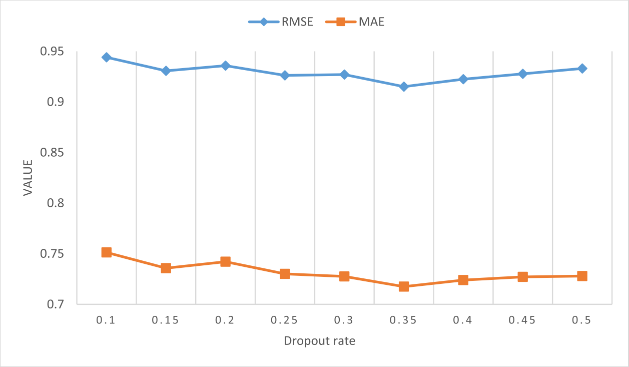


FIGURE 15. Impact of the dropout method on the prediction rating performance indicators RMSE and MAE of the algorithm model on the movielens-100K Dataset.

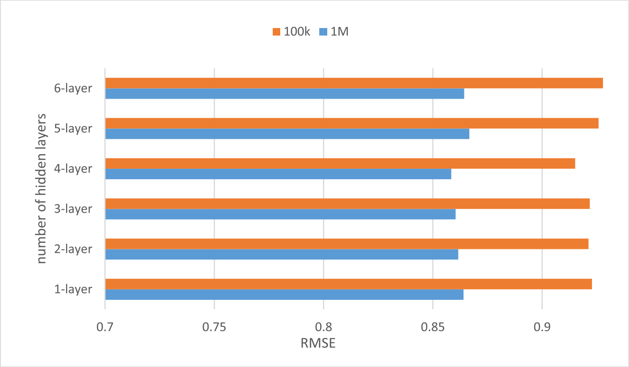


FIGURE 16. Impact of the number of hidden layers in MLP on the prediction rating performance indicators RMSE of the algorithm model on the movielens dataset.

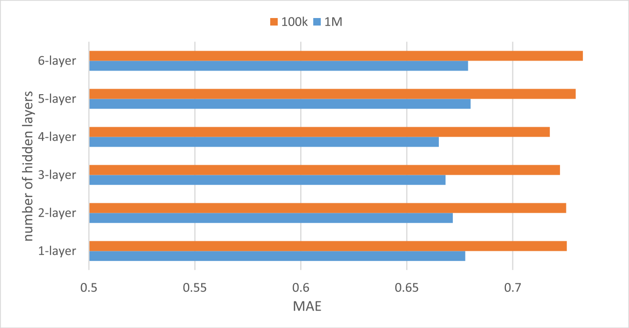


FIGURE 17. Impact of the number of hidden layers in MLP on the prediction rating performance indicators MAE of the algorithm model on the movielens dataset.

of the algorithm model reach the lowest point. Similarly, when the data set is MovieLens-100k, the fluctuation range of the broken line in Fig.15 is relatively small. Among them, the measurement result of the MAE value fluctuates between 0.7 and 0.75, and the measurement result of the RMSE value fluctuates between 0.9 and 0.95. When the ratio of dropout is 0.35, the RMSE and MAE values of the predictive rating performance indicators of the algorithm model reach the lowest point. In summary, because Movielens-100K has a relatively small amount of data in the dataset, the required learning parameters are less than the learning parameters in the Movielens-1M dataset. Therefore, the ratio of the dropout is 0.35, which makes the predictive performance of the proposed algorithm model optimal.

5) MULTI-LAYER PERCEPTRON ON MOVIELENS DATA SET

Fig.16 and Fig.17 depict the effect of the number of layers of the fully connected hidden layer of the model on the final predictive rating performance of the model. Fig.16 and Fig.17 reveal that when the number of layers in the hidden layer is 4, the predicted rating performance index RMSE and MAE of the algorithm model are optimal in the Movielens-1M and Movielens-100K data sets. From this we can know that the increase of the number of hidden layers of the algorithm model at any time can effectively improve the predictive rating performance. However, not the more hidden layers of the networks, the better. This experiment also confirmed the theoretical fact that the number of layers in the deep neural network has an optimal value.

6) EMBEDDING SIZE ON MOVIELENS DATA SET

Fig.18 and Fig.19 show the effect of the size of the word embedding dimension in the network model on the predictive rating performance of the algorithm model. When the data set is MovieLens-1M, the fluctuation range of the predicted rating performance indicators RMSE value and MAE value of the algorithm model is small and relatively flat. However, when the data set is MovieLens-100K, the fluctuations of the predicted rating performance indicators RMSE value and MAE value of the algorithm model are steep. This fully demonstrates that the difference in the embedded dimensions of the word not only affects the performance of the algorithm model, but also the performance of the algorithm model has a direct relationship with the size of the data set. The size of the data set and the size of the embedded dimension of the word directly affect the mining breadth and depth of its implicit relationship, so the difference in its expression effect ultimately leads to the difference in the overall predictive

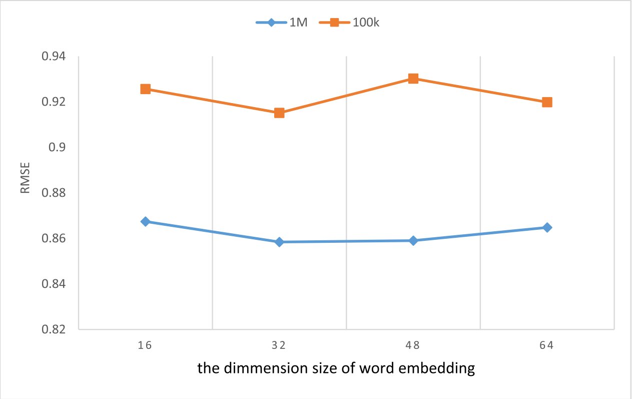


FIGURE 18. Impact of the embedding size on the prediction rating performance indicators RMSE of the algorithm model on the movielens dataset.

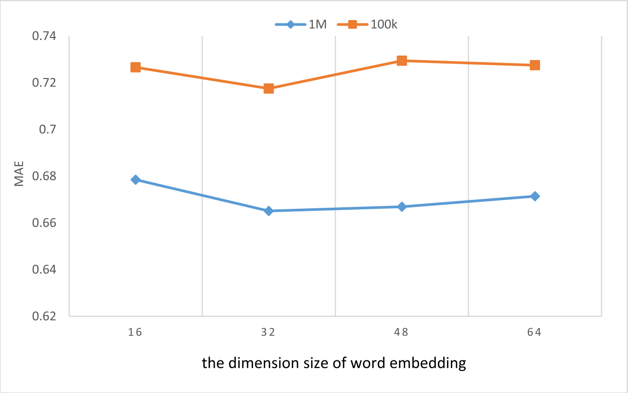


FIGURE 19. Impact of the embedding size on the prediction rating performance indicators MAE of the algorithm model on the movielens dataset.

rating performance of the algorithm model. Fig.18 and Fig.19 reveal that when the size of the word embedding dimension is 32, the prediction performance of the algorithm model in the data sets of Movielens-100K and Movielens-1M is optimal.

V. CONCLUSION

In this paper, a novel deep auto encoder model with convolutional text networks are proposed for solving the sparsity and cold start video recommendation problems by fully using the user features and item features of the video datasets. The extensive experiments are conducted to verify the superiority of the prosed algorithm on RMSE and MAE measure compared with the start-of-the-art algorithms, such as UserAverage, ItemAverage, topK-UBCF, topK-IBCF, SlopOne, PMF, BPMF, PRA, NRR and RMB-DNN algorithm. Besides that, the mode learning rate, auto encoder and activation function effect, dropout method and multi-layer perceptron of the proposed model are deeply studied based on the proposed model. The core part of our model are the deep auto encoder and convolutional text network sub-model. We can conclude that the deep auto ender and convolutional text network indeed can deeply mine the user features and the video item features, and hence, the rating prediction of our proposed model is sufficiently improved. In the future, the deep auto encoder and matrix factorization model maybe combined in order to improve the precision of the rating prediction algorithm. We hope other efficient models can also be studied and used in the personal recommendation area.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers with their instructive comments for improving the quality of this paper.

REFERENCES

[1] L. Yao, Q. Z. Sheng, A. H. H. Ngu, J. Yu, and A. Segev, "Unified collaborative and content-based Web service recommendation," *IEEE Trans. Services Comput.*, vol. 8, no. 3, pp. 453–466, May/Jun. 2015.

[2] J. Son and S. B. Kim, "Content-based filtering for recommendation systems using multiattribute networks," *Expert Syst. Appl.*, vol. 89, pp. 404–412, Dec. 2017.

[3] Z. Yang, B. Wu, K. Zheng, X. Wang, and L. Lei, "A survey of collaborative filtering-based recommender systems for mobile Internet applications," *IEEE Access*, vol. 4, pp. 3273–3287, 2016.

[4] A. Gogna and A. Majumdar, "A comprehensive recommender system model: Improving accuracy for both warm and cold start users," *IEEE Access*, vol. 3, pp. 2803–2813, 2015.

[5] J. Ding, Y. Huang, W. Liu, and K. Huang, "Severely blurred object tracking by learning deep image representations," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 26, no. 2, pp. 319–331, Feb. 2016.

[6] J. Hu, J. Lu, and Y.-P. Tan, "Deep metric learning for visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 11, pp. 2056–2068, Nov. 2016.

[7] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [Review Article]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.

[8] P. Zhou, H. Jiang, L. R. Dai, Y. Hu, and Q. F. Liu, "State-clustering based multiple deep neural networks modeling approach for speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 4, pp. 631–642, Apr. 2015.

[9] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 791–798.

[10] K. Georgiev and P. Nakov, "A non-IID framework for collaborative filtering with restricted Boltzmann machines," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, Feb. 2013, pp. 1148–1156.

[11] K. Patil and N. Jadhav, "Multi-layer perceptron classifier and paillier encryption scheme for friend recommendation system," in *Proc. Int. Conf. Comput., Commun., Control Automat. (ICCUBEA)*, Pune, Inida, Aug. 2017, pp. 1–5.

[12] M. Alfarhood and J. Cheng, "DeepHCF: A deep learning based hybrid collaborative filtering approach for recommendation systems," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Orlando, FL, USA, Dec. 2018, pp. 89–96.

[13] S. Zhang, L. Yao, and X. Xu, "Autosvd++: An efficient hybrid collaborative filtering model via contractive auto-encoders," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Tokyo, Japan, Aug. 2017, pp. 957–960.

[14] H. Xue, X. Dai, and J. Zhang, "Deep matrix factorization models for recommender systems," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2017, pp. 3203–3209.

[15] L. Zheng, V. Noroozi, and P. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Cambridge, U.K., Feb. 2017, pp. 425–434.

[16] H. Wu, Z. Zhang, K. Yue, K. Zhang, and R. Zhu, "Content embedding regularized matrix factorization for recommender systems," in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, Boston, MA, USA, Jun. 2017, pp. 209–215.

[17] L. Zhang, T. Luo, F. Zhang, and Y. Wu, "A recommendation model based on deep neural network," *IEEE Access*, vol. 6, pp. 9454–9463, 2018.

[18] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc.40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Shinjuku, Tokyo, Japan, Aug. 2017, pp. 355–364.

[19] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, Melbourne, Mornington, Australia, Aug. 2017, pp. 1725–1731.

[20] Q. Yuan, W. Liu, W. Rong, and Z. Xiong, "Autoencoder-based collaborative filtering," in *Proc. Int. Conf. Neural Inf. Process.*, Kuching, Malaysia, 2014, pp. 284–291.

[21] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, May 2015, pp. 111–112.

[22] Y. Wu, C. DuBois, A. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-N recommender systems," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, San Francisco, CA, USA, Feb. 2016, pp. 153–162.

[23] F. Strub, R. Gaudel, and J. Mary, "Hybrid recommender system based on autoencoders," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Boston, MA, USA, Sep. 2016, pp. 11–16.

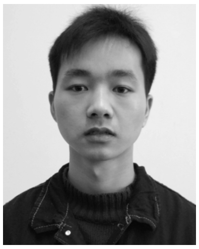
[24] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2008, pp. 1257–1264.

[25] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, Jul. 2008, pp. 880–887.

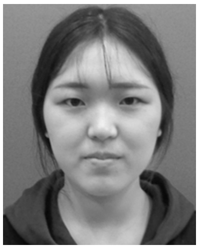
[26] H. Liang and T. Baldwin, "A probabilistic rating auto-encoder for personalized recommender systems," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Melbourne, Mornington, Australia, Oct. 2015, pp. 1863–1866.

[27] P. Li, Z. Wang, Z. Ren, L. Bing, and W. Lam, "Neural rating regression with abstractive tips generation for recommendation," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, Aug. 2017, pp. 345–354.

WENJIE YAN received the Ph.D. degree from the Harbin Institute of Technology. He is currently an Associate Professor with the School of Artificial Intelligence, Hebei University of Technology. His current research interests include machine learning and data mining.



DONG WANG received the bachelor's degree from the Hubei University of Technology. He is currently pursuing the master's degree with the School of Artificial Intelligence, Hebei University of Technology. His current research interests include machine learning and data mining.



MENGJING CAO received the bachelor's degree from Hebei Agriculture University. She is currently pursuing the master's degree with the School of Artificial Intelligence, Hebei University of Technology. Her current research interests include machine learning and data mining.

JING LIU received the Ph.D. degree from the Communication University of China. She is currently a Professor with the School of Artificial Intelligence, Hebei University of Technology. Her current research interests include data mining and fault analysis.