An Improved Recommendation Algorithm Based on Two-layer Attention Mechanism

Hye-jin Kim*

*Assistant Professor, Dept. of General Education, Kookmin University, Seoul, Korea

[Abstract]

With the development of Internet technology, because traditional recommendation algorithms cannot learn the in-depth characteristics of users or items, this paper proposed a recommendation algorithm based on the AMITI(attention mechanism and improved TF-IDF) to solve this problem. By introducing the two-layer attention mechanism into the CNN, the feature extraction ability of the CNN is improved, and different preference weights are assigned to item features, recommendations that are more in line with user preferences are achieved. When recommending items to target users, the scoring data and item type data are combined with TF-IDF to complete the grouping of the recommendation results. In this paper, the experimental results on the MovieLens-1M data set show that the AMITI algorithm improves the accuracy of recommendation to a certain extent and enhances the orderliness and selectivity of presentation methods.

▸Key words: Attention, Mechanism, Recommendation, Deep Learning, AMITI Algorithm

[요   약]

인터넷 기술의 발달로 기존의 추천 알고리즘은 사용자나 항목의 심층적인 특성을 학습할 수 없기 때문에 본 논문은 이 문제를 해결하기 위해 AMITI(주의 메커니즘 및 개선된 TF-IDF)에 기반한 추천 알고리즘을 제안했다. CNN(Convolutional Neural Network)에 2중 주의 메커니즘을 도입함으로써 CNN의 특징 추출 능력이 향상되고, 항목 특징에 다른 선호도 가중치가 할당되며, 사용자 선호도와 더 일치하는 권고사항이 달성되었다. 대상 사용자에게 항목을 추천할 때 점수 데이터와 항목 유형 데이터를 TF-IDF와 결합하여 권장 결과의 그룹화를 완료하였다. 본 논문에서 진행한 MovieLens-1M 데이터 세트에 대한 실험 결과는, AMITI 알고리즘이 권장 사항의 정확도를 향상시키고 프레젠테이션 방법의 순서와 선택성을 향상시킨다는 것을 보여준다.

▸주제어: 주의, 메커니즘, 권장, 딥러닝, 아미티 알고리즘

I. Introduction

With the development of Internet technology, data overload has become one of the dilemmas facing today's data explosion era[1-3]. In order to solve the problem of data overload, the recommendation system came into being. The recommendation system aims to filter out the content that users are interested in from the huge data and recommend it to users, so that online users can quickly find personalized information that meets their needs[4]. After years of development, the recommendation performance of the recommendation system has been greatly improved. The current recommendation algorithm is mainly divided into recommendation algorithm based on collaborative filtering[5], content-based recommendation algorithm[6] and hybrid recommendation algorithm[7]. The recommendation algorithm based on collaborative filtering finds similar users by obtaining user's historical behavior and rating data, and by obtaining the preferences of similar users and recommending items that they have not browsed before. The content-based recommendation algorithm is mainly based on the items or ratings that the user has selected, and based on the user's historical behavior information to find similar items for recommendation. The hybrid recommendation algorithm merges different recommendation algorithms before recommending to obtain a better recommendation effect. As data continues to grow, data types are becoming more and more diversified. Traditional recommendation algorithms cannot learn the deep-level characteristics of users or items. How to make full use of multi-source heterogeneous data to improve the performance of the recommender system has become a hot spot in the research of recommender systems[8].

In recent years, Deep Learning (DL) has its own powerful learning ability and has been widely used in image recognition, speech recognition, natural language processing and other fields[9]. Deep learning is good at mining and learning the deep-level features of multi-source heterogeneous data[10-16]. By combining it with the recommendation system, it can obtain more efficient learning of hidden features of users and item attributes[17]. Therefore, more and more researchers apply deep learning to recommendation systems. Although the application of neural networks to the recommendation system can effectively improve the recommendation performance, not all feature interactions can contribute to the prediction results. For example, when learning user or item features, interaction with useless features may introduce noise and affect the performance of the recommendation system[18].

This paper proposes a recommendation algorithm AMITI based on attention mechanism and improved TF-IDF. The attention mechanism is introduced into the Convolutional Neural Network (CNN), and the attention network is added before the convolutional layer to re-weight the pre-processed item text information. The user feature vectors and item feature vectors learned by the multi-layer fully connected neural network are input into the second-layer attention mechanism, and the Multilayer Perceptron (MPL) is used to parameterize the attention score. After completing the recommendation task, by combining user ratings and item categories with TF-IDF, analyzing the weight of different item types in the recommendation results, obtaining the user's preference for different item types, and classifying the recommendation results.

Chapter 2 of this paper summarizes the existing research trends, and Chapter 3 suggests 'Improved Recommendation Algorithm Based on Two-layer Attention Mechanism.' Chapter 4 includes the process of comparing the existing research method and the method proposed in this study through experiments, and Chapter 5 includes conclusions.

II. Research Foundation

1. Attention mechanism

Attention Mechanism was originally used in the field of image processing. When processing the image, the attention mechanism makes the neural network pay more attention to the key content of the input data[19].

MNIH et al.[20] combined the attention mechanism with Recurrent Neural Network (RNN) for image classification, reducing the interference of unnecessary information and improving the accuracy of classification. Attention mechanism is widely used in machine translation, target sentiment analysis, speech recognition and other fields. The attention mechanism is introduced into the neural network to make the network recognize the important information in the sentence when dealing with sentence classification problems, and pay more attention to this information, and at the same time improve the interpretability of the neural network. Literature[21] introduces the attention mechanism to RNN, proposes the ARNN structure for location recommendation, uses the historical check-in data of the target user to capture the user's life pattern, and makes interpretable location recommendation to the user. The AFM model proposed in [22] not only integrates the features of the project through the user's attention content, but also enhances the interpretability of the model.

2. Recommendation algorithm based on deep neural network

With the deepening of research, more and more deep neural network models are introduced into the recommendation system. The recommendation method based on deep learning can integrate multi-source heterogeneous data and generate recommendations. These data are divided into user's explicit feedback data such as ratings, likes/dislikes; implicit feedback data such as browsing, clicking and other behavior records; user attribute information Such as gender, age, occupation, etc.; project attribute information such as project name, brief introduction, etc.; social relations, tags, comments and other auxiliary information. By constructing a multi-layer neural network to input project or user-related information, and using regression and other methods to score and predict the input data, it solves the problems of data sparsity and cold start in traditional recommendation algorithms. Literature [23, 24] proposed a recommendation model based on neural collaborative filtering (NCF). The model learns the potential feature vectors of users and items through a parallel neural network, and maps the hidden vectors to predicted values through MLP in the prediction layer. The NCF recommendation model consists of two parallel neural networks, and the rating data of users and items pass through the embedding layer. The embedding layer first inputs the data allocation index, uses the index to construct the characteristic sequence of users and items; then converts the index into a vector with a fixed size. Through the embedding layer, the high-dimensional sparse vector in the input layer can be mapped to the low-dimensional dense vector representation. Both user and item data are sparse vectors obtained through one-hot encoding, and the user embedding vector p_u and item embedding vector q_v are obtained through the embedding layer as the input of multiple fully connected layers. Use MLP to learn the interaction functions between users and items:

$$\hat{y}_{uv} = f_{MPL}(p_u, q_v) \tag{1}$$

Where: $f_{MPL}(\cdot)$ is the activation function in the MLP network. Use the nonlinear activation function to enhance the nonlinearity and flexibility of the model, and define the MLP model as shown in equation (2):

$$y_1 = M(u,v) = p_u \odot q_v$$
$$y_2 = f_2(w_2(p_u \odot q_v) + b_2)$$
$$\vdots$$
$$y_n = f_n(w_n t_{n-1} + b_n)$$

(2)

Where: $\odot$ represents the element product operation; y_n, f_n, w_n, b_n respectively represent the output value, activation function, weight matrix and bias vector of the nth layer.

The NCF model uses MLP to extract high-order feature information to improve the recommendation ability of the model. In the process of feature interaction, not all feature interactions can contribute to the prediction results. Therefore, the attention mechanism is introduced into the neural network to assign personalized weights to the items in the historical interaction sequence, and the improved TF-IDF is used to group the recommendation results and recommend item groups to target users.

III. AMITI Recommendation Algorithm

On the basis of the NCF recommendation model, the user and item attribute information is used as input data $u : \{u_1, u_2, \cdots, u_n\}$, for example, user ID, age, gender, etc.; item attribute information $v : \{v_1, v_2, \cdots, v_n\}$, for example, item ID, type, title, etc. The AMITI model architecture is shown in Fig. 1.

Introduce a two-layer attention mechanism, and one layer is used to build a sub-network in combination with CNN, so that CNN learns the key content in the project text. The other layer takes the user and item feature vectors as input data, and uses the attention mechanism to assign personalized weights to the user's historical interaction items to obtain the effects of different items on the current predicted preferences. Group the recommended results and present the recommended results to users in the form of item groups to enhance the orderliness of the recommended content.
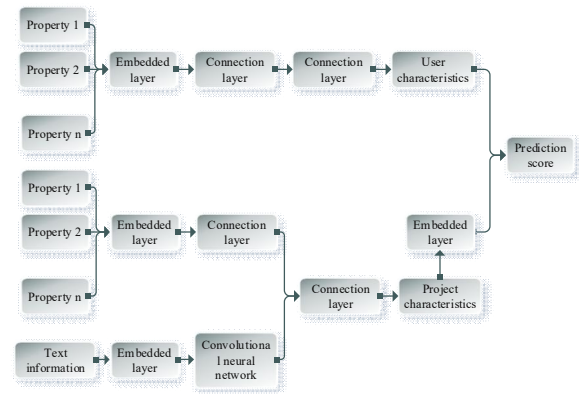

Fig. 1. AMITI model architecture

1. Learn the potential characteristics of users and projects

In order to improve the problem of data sparseness in the recommendation system, the attribute information of users and items is used to make score prediction. After data preprocessing, user and item attribute information is input into the embedding layer to encode the attribute information. The embedding layer maps the input sparse vector to a dense low-dimensional embedding vector to obtain embedding representations of user and item attributes p_u and q_v. At the beginning of training, the embedding is a simple random selection. As the training progresses, each embedding vector will be updated to help the neural network perform its task.

The embedding vectors p_u and q_v of users and items are input into the parallel multi-layer fully connected neural network, and the latent feature vectors of non-text attributes of users and items are learned respectively.

$$\hat{p}_u = f(w_{u2}f(w_{u1}p_u + b_{u1}) + b_{u2})$$

(3)

$$\hat{q}_v = f(w_{v1}p_v + b_{v1})$$

(4)

Where: $f(.)$ is the tanh activation function, w_n and b_n are the weight matrix and bias to be learned, respectively.

2. Convolutional neural network with attention mechanism

For the text information of item attributes such as item titles, in order to enhance the network's ability to learn key content in the text, the attention mechanism is combined with CNN to form a sub-network for extracting text features. The text convolutional neural network consists of an attention layer, a convolutional layer, a pooling layer, and a fully connected layer.

The attention layer assigns attention weight to the word vector matrix of each item text to obtain the updated word vector matrix. Pass the project text content through the embedding layer to obtain the word vector matrix $EINR^{n \times d}$, where d is the dimension of the word vector, that is, each word is mapped to a d-dimensional vector $x_j INR^d$, n is the number of words Number; $FINR^{3 \times d}$ represents the word vector matrix of the text information carried by all items browsed by the target user, and is the word vector of the first word, which means $x_i INR^d$. Calculate the word vector representation of each vocabulary in the target user's word vector matrix F and the attention score of each vocabulary in the word vector matrix E of all texts of the project.

$$a(x_i, x_j) = v_a^T R(w_a [x_i \oplus x_j]) \tag{5}$$

Where: v_a^T, are training parameters.

The $a(x_i, x_j)$ attention score is normalized by the softmax function, and the attention weight a_{ij} corresponding to each word vector is obtained.

$$a_{ij} = s(a(x_i, x_j)) = \frac{\exp(a(x_i, x_j))}{\sum_{j=1}^{n} \exp(a(x_i, x_j))} \tag{6}$$

Where: $a_{ij} \in A^{s \times d}$ is the value of the attention weight, the attention weight matrix $A^{s \times d}$ and the item original word vector matrix F are vector-joined to obtain the updated item word vector matrix $M_{att}^{s \times d}$, as the input matrix of the convolutional neural network.

$$M_{att} = A \circ F \tag{7}$$

In the convolutional layer, each neuron slides from the left most of the matrix F_{att} to the right by the convolution kernel $F_j INR^{d \times m}$ along the sentence direction. The window size of the convolution kernel F_j is set to m, the feature representation of each word in the sentence is obtained after the convolution operation, and then the feature map is formed through the activation function. The characteristics of the jth neuron are shown in formula (8):

$$C_j = f(M^{s \times d*} F_j + b_j) \tag{8}$$

Where: $*$ is the convolution operation; b_j is the bias term; f is the nonlinear activation function ReLU, and the activation function f is used to enhance the nonlinearity of the convolutional neural network.

$$R(x) = \max(0, x) \tag{9}$$

The maximum pooling is selected to perform a pooling operation on the output result of the convolutional layer, the feature map is divided into several rectangular regions, and the maximum value is output for each subregion. Maximum pooling removes features that are not important or repetitive for the current task in each sub-region, and retains information that can express text features. At the same time, the number of parameters is further reduced, and the efficiency of network feature extraction is effectively improved. The pooling result of the jth convolution kernel is shown in equation (10):

$$c_j' = \max\{c_1, c_2, c_3, \cdots, c_j^{d-m+1}\} \tag{10}$$

Input the pooled output result into the fully connected layer, multiply it with the weight matrix of the fully connected layer and add the bias

summation, and then classify the output through the ReLU activation function to obtain the hidden features of the project text information, as in the formula (11) Shown:

$$q_{text} = R(w_j c_j' + b_j) \tag{11}$$

Where: w_j is the weight coefficient of the fully connected layer; b_j is the corresponding bias term. The item feature $\hat{q}_j$ can be obtained by vector connection of the item non-text attribute feature vector $\hat{q}_v$ and the text feature vector q_{text}.

$$\hat{q}_j = c(\hat{q}_v, q_{text}) \tag{12}$$

3. Predictive score generation

The traditional neural network recommendation model usually implements the interaction between the implicit representation $\hat{p}_u$ of user characteristics and the implicit representation $\hat{q}_j$ of item characteristics in the prediction layer to obtain the final prediction score. Due to the lack of customization and optimization of recommendation tasks in this model, the equal treatment of all historical items of the user will limit its ability to express. Traditional movie recommendation methods usually recommend all movies in the user's playback history as context, which does not match the user's actual preferences. The traditional neural network recommendation model ignores that different items in the user's historical items play different roles in predicting the next item, so the accuracy rate is low. In the prediction layer of the AMITI model, the neural attention network is used to distinguish the importance of historical items to overcome the limitations of the traditional neural network recommendation model. The implicit representation $\hat{p}_u$ of learned user characteristics and the implicit representation $\hat{q}_j$ of item characteristics are used as the input of the attention layer. The degree of attention of the target user to different items is learned, and different degrees of attention play different roles in predicting the next item. The attention score of user u_i on item v_j is shown in equation (13):

$$s_c(\hat{p}_u, \hat{q}_j) = R_e(w_1(\hat{p}_u \odot \hat{q}_j) + b_1) \tag{13}$$

Where: w_1 and b_1 are the weight matrices and bias items that need to be learned, and the non-linear relationship between the current item and the predicted next item is obtained through the R_e activation function. $s_c(\hat{p}_u, \hat{q}_j)$ means that the larger the attention score, the more attention the user u_i has to the item v_j, and the greater the role of the item v_j in predicting the next item. Use the softmax function to normalize the attention score $\hat{a}_{ij}$.

$$\hat{a}_{ij} = \frac{\exp(s_c(\hat{p}_u, \hat{q}_j))}{\sum_{j \in INR(u)} \exp(s_c(\hat{p}_u, \hat{q}_j))} \tag{14}$$

where: $\hat{a}_{ij}$ is the contribution degree of project v_j to the user u_i preference curve; $R(u)$ is the historical interactive item set of user, u_i and the weight is redistributed through the implicit representation of the item feature, $\hat{q}_j$ as shown in equation (15):

$$\hat{q}_i = \sum_{j \in INR(u)} \hat{a}_{ij} \hat{q}_j \tag{15}$$

The user implicit feature $\hat{p}_u$ and the item implicit feature $\hat{q}_j$ are calculated as the inner product to obtain the predicted score, as shown in equation (16):

$$\hat{y}_{ui} = \hat{p}_u^T \hat{q}_i \tag{16}$$

The Mean Square Error (MSE) is used as the loss function to minimize the gap between the real

score and the predicted score in the process of training the model.

$$L_{sqr} = \sum_{(u,i)INR(u)} \left(y_{ui} - _{ui}\right)^2 \tag{17}$$

Where: y_{ui} is the user's actual rating of the item; y_{ui} is the user's predicted rating of the item. Stochastic gradient descent method is used to minimize the loss function to optimize the objective function, and the back propagation algorithm is used to optimize the learning weight w_n and bias b_n of each layer. After completing the training of the neural network through the above algorithm, the model is used to predict users to score unrated items, and the target users are recommended according to the predicted score size, and the resulting recommendation results are used for subsequent grouping to achieve item group recommendation.

4. Improvement of TF-IDF method

TF-IDF is often used for text classification and information retrieval, usually only considering the number of documents and the frequency of keywords appearing in the document, and when the word has score data, it cannot make full use of the score data to calculate the TF-IDF value more accurately. For example, when using TF-IDF to calculate the TF-IDF value of the comedy type in a movie watched by the user, only the frequency of the comedy type gj in the user browsing history movie collection and the entire data set is substituted into the calculation. Rating data of similar movies. When the comedy type g_i appears frequently in the user's browsing history, but the user has a higher rating for this type of movie, the traditional TF-IDF method cannot accurately obtain the user's preference for the comedy type g_i.

The scoring data is introduced into the TF-IDF method to avoid losing words with higher scores while evaluating the importance of words. The improved TF-IDF method is shown in formula (18):

$$S_{u_j w_i} = \frac{\sum_{s=1}^{n} r_{u_j v_s^{w_i}}}{\sum_{s=1}^{n} r_{u_j v_s}} 1b \frac{|D|}{\left|\{j : w_i INd_j\}\right|} \tag{18}$$

Among them: $S_{u_j w_i}$ is the importance of the word w_i to the user u_j, and the value is 0~1.

$\sum_{s=1}^{n} r_{u_j v_s^{w_i}}$ is the score sum of the file containing the word w_i. $\sum_{s=1}^{n} r_{u_j v_s}$ is the score sum of all files. $|D|$ is the total number of files in the corpus. $\left|\{j : w_i INd_j\}\right|$ is the number of files containing the word w_i. In formula (18), the score data $r_{u_j v_s}$ is determined according to the data set where the item is located. When performing statistics on the historical items of a specified user in the training set, S_{u_j, w_i} is calculated using the user's rating data of the historical items. Use the improved TF-IDF to calculate the TF-IDF value of the types contained in the items in the recommendation results, and achieve item group recommendation by obtaining the user's preference for different types. The $r_{u_j v_s}$ in formula (18) is the predicted score of the target user u_j on the item v_s in the candidate set obtained by the AMITI model. The first part of formula (18) is the term frequency calculation of TF-IDF (Term frequency, TF). The predicted score is used to calculate the item score of the file containing w_i and the proportion in the total file score. When the proportion is larger, it means that the higher the score of the file containing the word w_i, it reflects the importance of the word w_i to a certain extent. The second part is the Inverse document frequency (IDF) of TF-IDF, which means that among the total number of documents $|D|$ in the corpus, the more documents containing the word w_i, the stronger the weakening of the importance of TF, the word w_i. The $S_{u_j w_i}$ value obtained by multiplying the two parts indicates the user's preference for the word w_i. Based on the improvement of the TF-IDF algorithm, the item types in the recommendation

results are analyzed, the importance of different item types to the user is obtained, and the automatic grouping of the recommendation results is realized. Similar items in the Top-N recommendation results are put into the same group. The user's favorite items will be recommended first, so that users can quickly find content that meets their interests. The specific implementation steps of the improved TF-IDF algorithm are as follows:

Step 1 Calculating the word frequency is the proportion of the item scores containing the word w_i in the specific file and the sum of the scores of all items in the user's browsing history.

$$TF_{j,i} = \frac{\sum_{s=1}^{n} r_{u_j, v_s^{w_i}}}{\sum_{s=1}^{n} r_{u_j, v_s}} \qquad (19)$$

Step 2 Calculating the frequency of the reverse file refers to dividing the number of items in the data set by the number of items containing the word w_i, and then taking the logarithm of the obtained quotient, as shown in equation (20):

$$IDF_i = 1lb\frac{|D|}{|\{j : t_i INd_j\}|} \qquad (20)$$

Step 3 Multiply the word frequency and the word frequency of the reverse document to obtain the TF-IDF value S_{u_j, w_i} of the word w_i in the file d_j.

$$S_{u_j, w_i} = TF_{j,i} \times IDT_i \qquad (21)$$

The improved TF-IDF uses scoring data to reflect the proportion of word w_i in file d_j when calculating word frequency. The traditional TF-IDF word frequency calculation obtains the word frequency of the word w_i by comparing the number of occurrences of the word w_i in the file d_j to the sum of the number of occurrences of all words in the file d_j.

5. Recommended results grouping

Use the improved TF-IDF method to get the user's preference for different project types. Take movie recommendation as an example, choose MovieLens data set for algorithm test. The movie type word frequency information is shown in Table 1. First, recommend N movies to user u_i. Secondly, establish a movie information document $M = \{m_1, m_2, \cdots, m_N\}$ for the movies in the recommendation result, extract the included movie types from the movie information document, and establish a movie type information document $G = \{g_1, g_2, \cdots, g_N\}$. Finally, the word frequency statistics are performed on the movie type documents, and the word frequency information of the movie type in the recommendation result is obtained.

Table 1. Word frequency information of movie genres

Movie	Type			
	g_1	g_2	...	g_i
m_1	R_{m_1, g_1}	R_{m_1, g_2}	...	R_{m_1, g_i}
m_2	R_{m_2, g_1}	R_{m_2, g_2}	...	R_{m_2, g_i}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
m_s	R_{m_s, g_1}	R_{m_s, g_2}	...	R_{m_s, g_i}

When the recommended movie contains movie type g_i, R_{m_s, g_t} is 1, and vice versa. Use the improved TF-IDF to analyze the word frequency information of movie genres, and predict the user's preference for different movie genres, as shown in equation (22):

$$S_{u_j, g_i} = \frac{\sum_{s=1}^{n} r_{u_j, m_s} \times R_{m_s, g_t}}{\sum_{s=1}^{n} r_{u_j, m_s}} 1b\frac{N}{\sum_{s=1}^{n} R_{m_s, g_i}} \qquad (22)$$

Where: r_{u_j, m_s} is the user u_j's rating of the movie m_s; R_{m_s, g_t} is whether the movie m_s contains the type g_t; N is the number of movies recommended as a result; S_{u_j, g_i} is the user's rating of different movies The degree of preference for the type.

Sort the movie types in descending order according to their preference S_{u_j,g_i}, and take the first K types as the group name of the item group to be recommended, namely $L = \{L_1, L_2, \cdots, L_k\}$, where L represents the item group to be recommended gather. Add D movies of the same type to each project group. The movies in are derived from the recommendation results generated by the AMITI model. The movies in the project group are sorted in descending order according to the predicted score. In the end, it is composed of K project groups, and each project group contains D project group recommendations of the same type of movies.

6. AMITI algorithm description

Using a deep neural network combined with an attention mechanism can effectively improve the recommendation algorithm's ability to extract potential features of users and items, and alleviate the problem of data sparsity. According to the project type, the recommendation results are recommended to users. The overall implementation steps of the AMITI algorithm are as follows:

Input user attribute information $u : \{u_1, u_2, \cdots, u_n\}$, u_n represents the nth attribute of the user; project attribute information $v : \{v_1, v, \cdots, v_n\}$, represents the nth attribute of the project.

Output Generate K project groups, each project group contains D projects of the same type:

Step 1 Perform data preprocessing on user attributes and item attributes, and convert them into digital types.

Step 2 Input the user attributes and item ID and type attributes into the embedding layer to obtain low-dimensional dense embedding vectors p_u and q_v. Input p_u and q_v into the parallel multi-layer fully connected layer for feature learning, and obtain user features $\hat{p}_u$ and item non-text attribute vector $\hat{q}_v$.

Step 3 Perform convolution processing on the item name, and re-weight the word vector matrix of the movie name through the attention mechanism to obtain the updated word vector matrix $M_*^{s \times d}$.

Step 4 Input the word vector matrix $M_*^{s \times d}$ into the convolutional neural network to extract the item name feature and generate the feature, and then use the tf.concat() function to merge the attributes of the item to obtain the final item feature $\hat{q}_j$.

Step 5 Use the attention mechanism to assign a personalized weight $\hat{a}_{ij}$ to each item, and get the updated item feature $\hat{q}_i$.

Step 6 Calculate the predicted score $\hat{y}_{ui}$ by calculating the inner product of the user's implicit feature $\hat{p}_u$ and the item's implicit feature $\hat{q}_i$.

Step 7 Enter the top-N recommendation results generated by the specified user, and use the improved TF-IDF to analyze the user's preference value S_{u_j,g_i} for different item types.

Step 8 Sort the item types in descending order according to the preference degree S_{u_j,g_i} value, and take the top K types as the group name of the item group to be recommended, K is

Step 9 In each project group, add D movies of the same kind corresponding to the group name, and arrange them in descending order of the predicted score of each movie. The final process consists of K item groups, and each item group contains D items of the same type of film for recommendation.

Compared with the traditional collaborative filtering recommendation algorithm, the AMITI algorithm takes user and item attribute information as input data, and generates predictive scores through deep neural network feature extraction, which solves the problem of sparseness of score data in the traditional collaborative filtering recommendation algorithm. Compared with the NCF recommendation model, the AMITI algorithm introduces an attention mechanism in the deep

neural network. On the one hand, it enhances the ability of CNN to extract important content in the project text; on the other hand, it distinguishes the different effects of historical items on predicting the next item. Can effectively improve the recommendation accuracy.

IV. Experimental Results

1. Introduction to the experimental environment and data set

The computer used in the experiment is configured with an Intel Core i5-4210U CPU, 8 GB memory, and Windows 7 Ultimate 64-bit operating system. The programming language uses python language, the version is python3.6, and the compilation environment is implemented in Jupyter Notebook of Anaconda.

The data set used in the experiment is the movie data set MovieLens-1M provided by Group Lens[25] in the United States to verify the effectiveness of the above recommendation algorithm. This data set is widely used in the experiment of recommendation system, which contains 1000 209 rating records of 3 952 movies by 6,040 users, with a rating range of 0-5. Each of the 6,040 users rate at least 20 movies. In the movie training set, the metadata of each movie (movie ID, movie title, release time, and genre of the movie) are included. Included in the score training set (user ID, movie ID, corresponding user score, time stamp). Include demographic data (age, gender, occupation) about the user in the user training set. The sparsity of the data set is about 94%. In the experiment, 80% of the data is randomly selected as the training set and 20% as the test set. The model is trained through the training set data, and the performance of the algorithm is evaluated with the test set data.

2. Evaluation Index

In this paper, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are used as indicators to evaluate the performance of the model. For users in the test set, the true score of user u for item I is r_{ui}, and the predicted score calculated by the recommendation algorithm is r_{ui}, then RMSE (RMSE) and MAE (MA) are calculated as equations (23) and (24) Shown:

$$R_{MSE} = \sqrt{\frac{\sum_{u,i\int} \left(r_{ui} - \hat{r}_{ui}\right)^2}{|T|}} \tag{23}$$

$$M_A = \frac{\sum_{u,iINT} \left|r_{ui} - \hat{r}_{ui}\right|}{|T|} \tag{24}$$

Where: T is the number of items in the test set with scoring records.

After verifying the accuracy of the recommendation results, the reliability of the final item group recommendation results is verified according to the retention rate formula, as shown in equation (25):

$$R_{Retention\ Rate} = \frac{\sum_{uINT} |R(u) \cap F(u)|}{\sum_{uINT} |F(u)|} \tag{25}$$

Where: R(u) is the recommendation result before grouping; F (u) is the final recommendation result.

3. Comparative experiment

In order to verify the accuracy and effectiveness of the recommendation results obtained by the project team recommendation algorithm, the following four classic recommendation algorithms are selected in the unified data set Movies Lens to compare with the project team recommendation algorithm.

1. Item CF: Item-based collaborative filtering recommendation algorithm, which generates a recommendation list for users based on the

previously calculated item similarity and user historical behavior[26].

2. CDL: Collaborative deep learning is a model that uses stacked denoising self-encoding to extract the features of a description document and combines user historical scoring data to generate recommendations.

3. NAIS: Project MLP-based collaborative filtering recommendation model fused with attention mechanism.

4. NCF: Neural collaborative filtering recommendation algorithm, which uses the MLP neural network to learn the interaction function between the user and the item from the hidden vector of the user and the item to generate a predictive score.

4. Analysis of experimental results

The proposed AMITI algorithm is evaluated experimentally on the movie data set of MovieLens-1M, and RMSE and MAE are used as evaluation indicators to measure the accuracy of the prediction score. Under the condition that the experimental data set and the experimental environment are the same, the influence of the following three hyperparameters on the recommended performance of the AMITI model is studied.

1. The number of iterations: 1 number of iterations means that the neural network is trained once using all the samples in the training set.

2. Dropout rate: In the case of more model parameters and fewer training samples, the trained model is prone to overfitting. Use the discard rate to randomly erase some neurons to reduce the number of parameters, thereby preventing the occurrence of over-fitting.

3. Learning rate: Determine the convergence result and efficiency of the model.

As the number of iterations increases, the number of weight updates in the neural network also increases. Excessive number of iterations can easily cause over-fitting of the model, resulting in

reduced model accuracy. When the number of iterations is greater than 10, MAE and RMES increase. The influence on MAE and RMSE when the number of iterations is different is shown in Fig. 2.
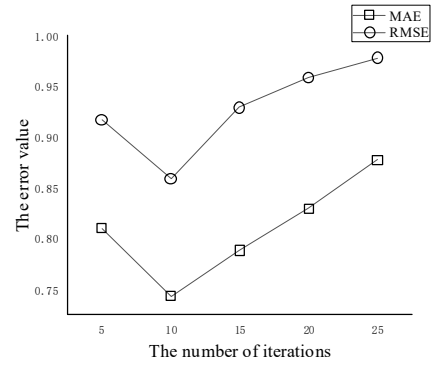

Fig. 2. Comparison of experimental results when the number of iterations takes different values

The discard rate in the hyperparameters can prevent the model from overfitting. The discard rate causes some hidden layer units to stop working during training, which improves the generalization ability of the network. It can be seen from Figure 6 that when the discard rate is 0.5, the model effect is optimal, that is, half of the nerves are discarded during training. As the discard rate increases, more neurons are discarded, which will decrease. The ability of neural network to learn features. Fig. 3 shows the impact on MAE and RMSE when the discard rate takes different values.
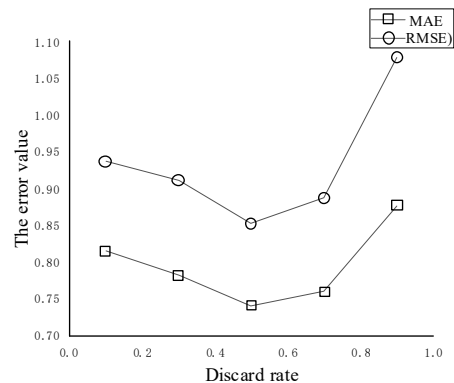

Fig. 3. Comparison of experimental results when the discard rate takes different values

The influence of different values of learning rate on model performance is shown in Fig. 4. The

model works best when the learning rate is 0.0001. As the value of the learning rate increases, the MAE and RMSE of the model increase accordingly. The larger the value of the learning rate, the more likely it is to miss the local minimum, which makes it difficult to fit the model.
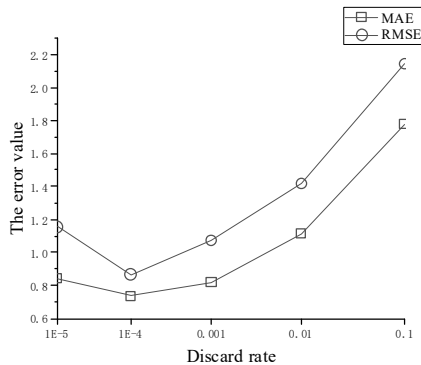

Fig. 4. Comparison of experimental results when the learning rate takes different values

It can be seen from Figures 2 to 4 that when the discard rate, learning rate, and training size of each batch are 0.5, 0.0001, and 256, respectively, the number of iterations is set to 10, and the root mean square error and average absolute error of the model are the lowest.

The comparison of the root mean square error and average absolute error between the project team recommendation and the other four recommendation algorithms is shown in Fig. 8. In the MovieLens-1M data set, compared with Item CF, CDL, NAIS, and NCF, the AMITI algorithm has increased RMSE by 14.09%, 4.46%, 2.37%, 2.04%, and MAR by 14.38%, 3.65%, respectively, 2.77%, 2.47%. Item CF is inferior to other algorithms, MAE and RMSE are larger; CDL algorithm, NAIS algorithm and NCF algorithm have similar indicators. Compared with other algorithms, the AMITI algorithm is improved by more than 2%. The neural network based on the attention mechanism has improved the accuracy of the recommendation system to a certain extent after extracting the hidden features of users and items.
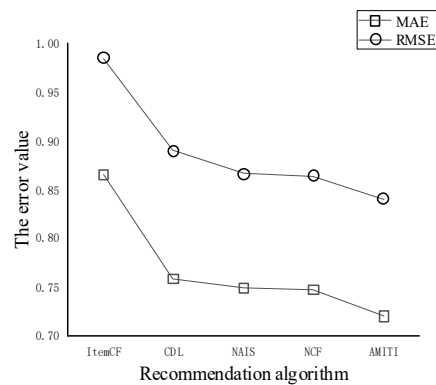

Fig. 5. Comparison of RMSE and MAE results of different recommendation algorithms

When calculating the user's movie genre preference, it is found that each user has 2 to 4 special preferences, and it is determined that the number of final recommended item groups is set to have the best recommendation effect. The effect of different values of Top-N on the retention rate is shown in Fig. 6. Set the number of item groups to 2, 3, and 4 respectively. Each item group contains 4 movies of the same type. At the same time, different Top-Ns are selected to compare the changes in the corresponding retention rates in these three cases. The retention rate increases with the increase in the number of project groups. When the value of Top-N is the same, the smaller the number of project groups, the higher the retention rate, and the retention rate also reaches 100% faster. However, if only 2 item groups are recommended in the final recommendation, it is difficult to cover the needs of most users for movie genre preference, and the number of final recommended item groups is determined as 3.
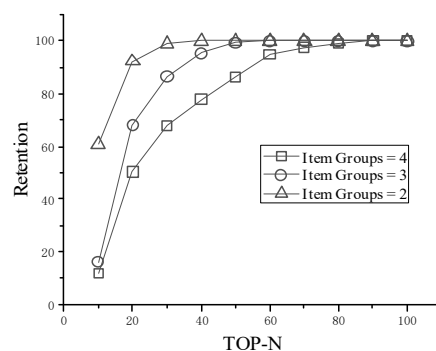

Fig. 6. The influence of different values of Top-N on retention rate

Movies with higher predicted scores are first recommended to users. When Top-N is selected, the value of N should be as small as possible and the retention rate should be as large as possible. According to Figure 6, in order to appropriately increase the probability of unpopular items, when the number of item groups is 3, N=50, and at the same time ensure that 1 to 2 items are randomly recommended for each recommendation.

V. Conclusion

Aiming at the problem that the traditional recommendation system mainly relies on the user's rating data of items and cannot learn the deep-level characteristics of users and items, a recommendation algorithm (AMITI) based on the attention mechanism and improved TF-IDF is proposed. By introducing the dual-layer attention mechanism into the parallel neural network recommendation model, the model's ability to mine important features is improved. Improve TF-IDF based on user ratings and item categories, and classify recommendation results according to item category weights to construct different types of item groups and complete recommendations. The experimental results show that the AMITI algorithm can increase the attention to important content in the text and the attention weight assigned to items, effectively improve the recommendation accuracy and improve the recommendation effect after the item group recommendation is realized.

REFERENCES

[1] Jung-Ha Park, "Comparison between eLearning video and Smartphone Application for Information Technology Use in Nursing Education", Asia-pacific Journal of Convergent Research Interchange, Vol. 5, No. 4, pp.39-47, December 2019. DOI: http://dx.doi.org/10.21742/apjcri.2019.12.05

[2] Ashwini L. Kadam, Hoon Lee, Mintae Hwang, "Implementation of IoT Application using Geofencing Technology for Protecting Crops from Wild Animals", Asia-pacific Journal of Convergent Research Interchange, Vol. 6, No. 6, pp.13-23, June 2020. DOI: http://dx.doi.org/10.21742/apjcri.2020.06.02

[3] Weizhen Fang, Shanyue Jin, "A Study on Equity Incentive Schemes of Wangsu Technology Enterprises", Asia-pacific Journal of Convergent Research Interchange, Vol. 6, No. 10, pp.65-76, October 2020. DOI: http://dx.doi.org/10.47116/apjcri. 2020.10.05

[4] ADOMAVICIUS G., TUZHILIN A., "Toward the next generation of recommender systems : a survey of the state-of-the-art and possible extensions", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, pp.734-749, April 2005. DOI: 10.1109/TKDE.2005.99

[5] Xiaofeng Li and Dong Li, "An Improved Collaborative Filtering Recommendation Algorithm and Recommendation Strategy", Mobile Information Systems, Vol. 2019, pp.1-11, May 2019. DOI: https://doi.org/10.1155/2019/3560968

[6] Yan Li, Hanjie Wang, Hailong Liu, Bo Chen, "A study on content-based video recommendation", Proceedings of 2017 IEEE International Conference on Image Processing, pp.4581-4585, 2017.

[7] YANG J. R., YANG C., HU X. W., "A study of hybrid recommendation algorithm based on user", Proceedings of International Conference on Intelligent Human – Machine Systems & Cybernetics, pp.261-264, 2016.

[8] ELKAHKY A. M., SONG Y., He X., "A multi-view deep learning approach for cross domain user modeling in recommendation systems", Proceedings of the 24th International Conference on World Wide Web, pp.278-288, 2015.

[9] HE X. N., CHUA T. S., "Neural factorization machines for sparse predictive analytics", Proceedings of the 40th International ACM SIGIR Conference, New York, pp.355-364, 2017.

[10] Daehyon Kim, "Deep Learning Neural Networks for Automatic Vehicle Incident Detection", Asia-pacific Journal of Convergent Research Interchange, Vol.4, No.3, pp.107-117, September 2018. DOI: http://dx.doi.org/10.14257/apjcri.2018.09.11

[11] Daehyon Kim, "Normalization of Input Vectors in Deep Belief Networks (DBNs) for Automatic Incident Detection", Asia-pacific Journal of Convergent Research Interchange, Vol.4, No.4, pp.61-70, December 2018. DOI: http://dx.doi.org/10.14257 /apjcri.2018.12.07

[12] Lei Li, "An Extensive Review on Recent Deep Learning Applications", Asia-pacific Journal of Convergent Research Interchange, Vol.5, No.3, pp.221-231, September 2019. DOI: http://dx.doi.org/10.21742/apjcri.2019.09.22

[13] Daehyon Kim, "Application of Deep Neural Network Model for Automated Intelligent Excavator", Asia-pacific Journal of Convergent Research Interchange, Vol.6, No.4, pp.13-22, Apr, (2020), DOI: http://dx.doi.org/10.21742/apjcri.2020.04.02

[14] Jennefer Mononteliza, "Research on EIoT Reservation Algorithm

Based on Deep Learning", Asia-pacific Journal of Convergent Research Interchange, Vol.6, No.9, pp.191-205, September 2020. DOI: http://dx.doi.org/10.47116/apjcri.2020.09.16

[15] B. Harini, N.Thirupathi Rao, "An Extensive Review on Recent Emerging Applications of Artificial Intelligence", Asia-pacific Journal of Convergent Research Interchange, Vol.5, No.2, pp.79-88, June 2019. DOI:http://dx.doi.org/10.21742/apjcri.2019.06.09

[16] Won-seok Bang, Sun-Hwa Kim, Kuk-Hoan Wee, "A Study on the Effect of CSV on the Performance of Partner Companies: An Analysis of Artificial Neural Networks", Asia-pacific Journal of Convergent Research Interchange, Vol.6, No.7, pp.125-132, July 2020. DOI:http://dx.doi.org/10.47116/apjcri.2020.07.12

[17] Neeru Narang, Michael Martin, Dimitris Metaxas, Thirimachos Bourlai, "Learning Deep Features for Hierarchical Classification of Mobile Phone Face Datasets in Heterogeneous Environments", Proceedings of IEEE International Conference on Automatic Face & Gesture Recognition, pp.1120-1127, 2017.

[18] Tsioptsias N., Tako A. A., Robinson S., "Can We Learn From Wrong Simulation Models? A Preliminary Experimental Study on User Learning", Proceedings of The Operational Research Society Simulation Workshop, pp.1-11, 2018.

[19] Mandy Korpusik, Zachary Collins, James Glass, "Semantic mapping of natural language input to database entries via convolutional neural networks", Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pp.5685-5689, 2017.

[20] Volodymyr Mnih, Nicolas Heess, Alex Graves, Koray Kavukcuoglu, "Recurrent models of visual attention", Proceedings of Conference on Neural Information Processing Systems. pp.2204-2212, 2014.

[21] Brglez F, Pyun Y J. On the Use of Isomorphs to Enhance the Teaching and the Grading Methods in a Data Structures Course[J], jupiter3.csc.ncsu.edu, 2010.

[22] Aguado M., Asorey M., Ercolessi E., F. Ortolani, S. Pasini, "DMRG Simulation of the SU (3) AFM Heisenberg Model", Physical review, B, Condensed matter Vol.79, No.1, pp.35-44, 2009. DOI:10.1103/PhysRevB.79.012408

[23] Gabriele Sottocornola, Fabio Stella, Markus Zanker, Francesco Canonaco, "Towards a deep learning model for hybrid recommendation", Proceedings of the International Conference on Web Intelligence, pp.1260-1264, 2017.

[24] Athanassopoulos C. A., "Recommender System Framework combining Neural Networks and Collaborative Filtering", Proceedings of the 5th WSEAS international conference on Instrumentation, measurement, circuits and systems, pp.285−290, 2006.

[25] F. Maxwell Harper, Joseph A. Konstan, "The MovieLens Datasets: History and Context", ACM Transactions on Interactive Intelligent Systems, Vol.5, No.4, pp,1-19, 2015. https://doi.org/10.1145/2827872

[26] Ammar Jabakji, Hasan Dag, "Improving item-based recommendation accuracy with user's preferences on Apache Mahout", Proceedings of IEEE International Conference on Big Data, pp.1742-1749, 2016

Authors

Dr. Hye-jin Kim received the B.S. and M.Edu. degree in Wooseok University, Jeonju, Korea. She received the Ph.D. in Computer Science and Education from University of Bristol in 2017.

Dr. Kim worked as a lecturer in Jeonju Vision University Continuing Remote Education Center, Korea for 3 years. Now she is working as an Assistant Professor at Dept. of General Education, Kookmin University, Korea. Her research interests include U-learning, Education Technology, Artificial Intelligence, IoT, Remote Education Management Technology.