International Journal of Smart Business and Technology
Vol.8, No.1 (2020), pp.21-32
http://dx.doi.org/10.21742/ijsbt.2020.8.1.03

A CNN based Personalized Product Recommendation Model

Daniela De Venuto[1] and Francesco Bellotti[2*]

[1,2]Department of Computer Science and Engineering, University of Bologna, Bologna, Italy*
[2]Francesco.bellotti@unibo.it*

Abstract

With the rapid development of information technology and Internet technology, how to dig out products or services that users are interested in from massive information and display them in a personalized manner according to the score prediction results has become a current research hotspot. Collaborative filtering is currently a technology commonly used in personalized recommendation systems. Its basic idea is to use users with the same interests in the past to choose similar products in the future. This paper proposes a personalized product recommendation model based on deep neural networks and dynamic collaborative filtering. This model uses the BERT model and two-way GRU to replace the traditional word vector text processing method, which can effectively reduce the impact of data sparsity and achieve deeply hidden feature extraction. The scoring matrix is predicted by the coupled CNN network, and the product scoring prediction is realized based on the dynamic collaborative filtering of the fusion time series items. The experimental results verify that the model reduces the impact of data sparsity and cold start problems, takes into account the changes in user interest over time, and has obvious advantages in recommendation accuracy.

Keywords: *CNN, Recommendation model, Collaborative filtering*

1. Introduction

With the rapid development of information technology and Internet technology, the scale of data has increased sharply, and information overload is serious. The recommendation system is an important technical means to solve the problem [1][2]. At present, the quality of the recommendation algorithm in the recommendation system of major e-commerce platforms has a profound impact on the development of e-commerce. The personalized recommendation is based on the needs of users, mining products or services that users are interested in from massive amounts of information, and presenting them in a personalized manner according to the score prediction results. The recommendation algorithm is the core of personalized recommendation, which directly determines the recommendation performance of the recommendation system, and has become a hot issue in current research [3][4].

Collaborative filtering is currently a technology commonly used in personalized recommendation systems. Its basic idea is to use users with the same interests in the past to choose similar products in the future. A typical collaborative filtering algorithm is a collaborative filtering recommendation model based on matrix decomposition. The model first characterizes user and product features separately and then uses the dot product of user and product feature vectors to predict the score matrix, which has good recommendation

Article history:
Received (October 16, 2019), Review Result (December 1, 2019), Accepted (February 19, 2020)

Print ISSN: 2288-8969, eISSN: 2207-516X IJSBT

performance, but the score data The sparsity problem has always restricted the bottleneck of traditional collaborative filtering [5]. In recent years, deep learning technology has been used to solve the problem of personalized product recommendation, and hybrid recommendation models based on deep learning have been proposed one after another, such as DeepCoNN [6], DLALSTM [7], NARRE [8] and so on. These models first use the deep learning technology to transform the user's review data of the product into the hidden space and then use the traditional recommendation model to complete the personalized recommendation after the score matrix is obtained, but the static word vector encoding affects the evaluation performance of the score matrix and reduces the accuracy of the recommendation system.

2. Recommended model

The key of the current collaborative filtering product recommendation algorithm is the prediction of product scores, and there are mainly the following three aspects. First, the sparsity of the rating matrix. The sparsity of review data is a key factor that affects the accuracy of the final score prediction of the personalized product recommendation system. This is because the number of products on the platform is far greater than the number of comments by a single user, resulting in a very obvious sparseness of the comment data, which in turn affects the lack of relationship between users and products, and between users and users, which reduces the prediction results of product ratings. accuracy. The second, the cold start problem. There are two main reasons for the cold start of the recommendation system, namely, new users and new products. In the product recommendation platform, new users register every day, and there are also new products on the shelves. The system can't find new products or the level of interest of new users. Finally, the problem of information expiration. Over time, the user's interest preference or the popularity of the product will change, and the traditional recommendation algorithm does not consider the impact of this change.

This paper proposes a personalized product recommendation model based on convolutional neural networks and dynamic collaborative filtering. The model architecture is shown in [Figure 1].

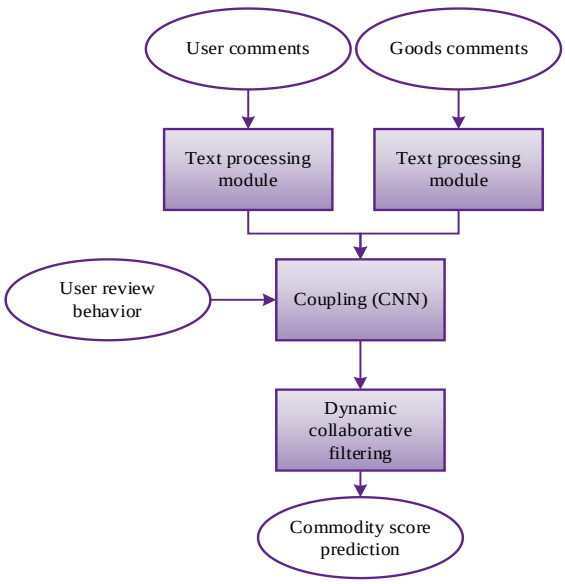

Figure 1. Recommended model architecture

First, the text information of user reviews and product reviews is processed by the BERT model and two-way GRU, and the hidden feature vectors of users and products are extracted using parallel convolutional neural networks; then, the user's scoring behavior of the product is introduced, and the TimeSVD algorithm is used to achieve the final result. Product rating prediction. This model uses convolutional neural networks to capture the hidden feature vectors of users and products to reduce the data sparseness and cold start problems that exist in the process of user's product rating matrix prediction; integrate time into the collaborative filtering process Sequence, reduce the impact of the information process in the product recommendation process.

3. Hidden feature extraction of users and products based on deep learning

3.1. Comment text processing

The bag-of-words model is a typical word segmentation review text processing model, but the model is not sensitive to the order of words in the text and often uses the same word vector to represent the text with opposite meanings [9]. And the noise of the review text is large. The use of network language also seriously affects the text processing performance of word segmentation. To this end, the text uses a two-way GRU text processing model [10], as shown in [Figure 2]. Since user reviews and product reviews are two parallel models with similar structures, only the processing process of product review text is introduced.

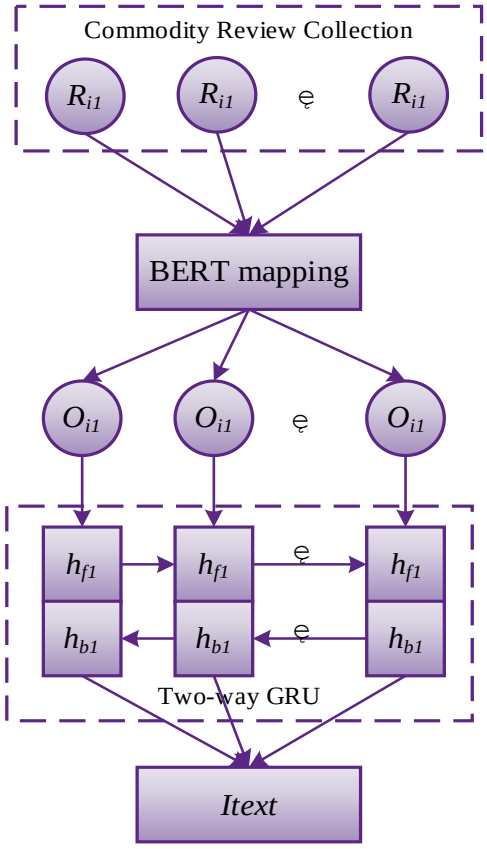

Figure 2. Product review processing process

Let I denote a collection of reviews for a product:

$$I = \{R_{i1}, R_{i2}, \cdots, R_{id}\}$$

where $R_{ij}(j = 1; 2; \cdots; d)$ means a single comment, d represents the maximum number of reviews set by the model. Input product review data into the BERT mapping model [5], and convert a single review data into a vector form.

In the BERT mapping process, if the number of product reviews is greater than d, only the first d reviews of the product are mapped; if the number of product reviews is less than d, then the length of the mapped vector is equal to d by filling in the zero vector.

Taking into account the internal connection between the comments, the two-way GRU is used to encode the vector memory after the BERT mapping. The encoding process includes the forward and subsequent parts, where the forward is processed in the positive order $O_{i1} \rightarrow O_{id}$), and the backward is processed in the reverse order $O_{i1} \rightarrow O_{id}$), get the encoded sequence and respectively. Next, get the comment vector feature by splicing:

$$h_i = h_{fi} \oplus h_{bi}$$

The feature vector after two-way GRU encoding can be expressed as:

$$H_0 = \{h_1, h_2, \cdots, h_d\}$$

It is known that the content of each product review has its characteristics, that is, it implies the contribution to the different preferences of the product. Therefore, the self-attention mechanism is used to summarize the information of the feature vector, and the attention vector obtained after the aggregation can be expressed as:

$$a = softmax\left(\omega_1 \times tanh(\omega_2 \times H_0^T)\right)$$

Where, ω_1 and ω_2 both represent the weight parameters whose dimensions meet the requirements, and $softmax()$ represents the normalization processing based on the attention mechanism. The attention vector contains the contribution of reviews to product preferences. After the hidden feature vectors of the same product are weighted and summed and followed by a fully connected layer, the final implicit feature vector of the review set u for the product can be obtained.

$$I_{text} = W' \times aH_0 + b'$$

Where W' and b' respectively represent the weight parameter and the bias parameter of the fully connected layer. In the same way, the final implicit feature vector U_{text}. of the review set to the user can be obtained in the same way.

3.2. Score prediction based on convolutional neural network

After obtaining the user and product preference expressions according to the text processing method in the previous section, this section first extracts the deep features of the user and the product based on the convolutional neural network and uses the shared layer to map them to the same feature space for scoring. Matrix prediction.

Input the final hidden feature vector to the convolutional neural network to extract its deep-level features, assuming that the number of neurons in the convolutional layer of the convolutional neural network is m, and the convolution kernel of the jth neuron is $K_j \in R^{c \times d}$, where d is the dimension of the hidden feature vector. For the hidden feature vector (take a product as an example), the output of each convolution kernel is:

$$K_j = G\left(I_{text} * K_j + b_j\right)$$

Where, $G()$ and b_j represent the activation function and bias term of the convolution kernel, respectively. The activation function in the text adopts the linear rectification function ReLU, which is specifically:

$$G(x) = max\{0, x\}$$

ReLU can effectively avoid the phenomenon of gradient disappearance and gradient explosion in the training process, and can significantly improve the convergence speed of the stochastic gradient descent algorithm and reduce the model training time [11].

Next, the feature vector after the convolution operation is pooled, and the maximum pooling method is used to extract the maximum value of the feature vector as the feature of the specific convolution kernel. After the maximum pooling operation, the output result of the convolutional layer is Converted to a fixed-length vector, the pooling operation can be expressed as:

$$R_i = max\{k_1, k_2, \cdots, k_{m-d+1}\}$$

According to the above convolution operation and pooling operation process, it can be seen that after the convolutional layer passes the output feature vector through the maximum pooling operation, the information with the strongest features in the text information is filtered out, and the information with weaker features has been pooled. Was filtered out. To ensure the integrity of the text information, multiple parallel convolutional neural networks are used in the article to obtain various features of the text information. This is because multiple parallel convolutional layers can extract different local information, and the output of the parallel convolutional layer The feature set can be expressed as:

$$R = \{R_1, R_2, \cdots, R_M\}$$

Where M represents the number of convolution kernels in the parallel convolution layer.

The output of the pooling layer is imported into the fully connected layer, and let W denote the weight matrix of the fully connected layer, then the output of the fully connected layer can be expressed as:

$$X_I = f(W \times R + b)$$

Where, $f()$ and b are the activation function and bias term of the fully connected layer, respectively. The above is the deep feature X_I obtained as an example of the product, and the user comment text is also input, and the deep feature X_U of the user comment can be obtained.

Although the deep features X_I and X_U are both derived from review data, the feature representations of two different mapping spaces for users and products are not comparable. They must be transformed into the same feature space before scoring prediction. For this reason, the sharing layer is used to couple the hidden features of user review data and the hidden features of product review data, that is, X_I and X_U are coupled into a single feature vector $Z = (X_I, X_U)$

4. Commodity recommendation model based on dynamic collaborative filtering

The dynamic collaborative filtering algorithm is based on the SVD algorithm and adds time series items, so it is called the TimeSVD algorithm [12]. The TimeSVD algorithm is evolved from a simple factorization model, assuming that the user's rating of the product has been

obtained Prediction matrix, where N and M represent the number of users and the number of products, respectively, and r_{ui} in the matrix represents the predicted value of user u's rating for the product i.

The review text information is processed by the convolutional neural network, and the deep feature vectors of the user and the product are obtained as X_I and X_U, respectively, and the shared layer is coupled and the factorization machine is used to obtain the user's scoring matrix predicted value $\hat{r}_{ui}$. TimeSVD The algorithm first needs to reduce the error between r_{ui} and $\hat{r}_{ui}$ to obtain the best score prediction. The specific process can be expressed as:

$$\hat{r}_{ui} = X_I X_U$$

$$F = \sum_{ru} (v_{ui} - r_{ui})^2$$

Adding a bias term in the scoring prediction process constitutes an SVD model:

$$f_{ui} = X_I X_U + u + b_u + b_i$$

In the formula, u is the average value of the user's rating prediction value of the product during the training process, b_u and b_i represent the user bias item and the product bias item, respectively, and represent the average value of the rating prediction value of a user or a product.

Based on the SVD model, the SVD algorithm adds user interest information through implicit feedback information. That is to say, as long as any user has commented on a certain product, regardless of the predicted value of the comment content, it means that the user is interested in the product, and the degree of interest is expressed by the hidden factor $Y_j = \{Y_{j1}, Y_{j2}, \cdots, Y_{jF}\}$. At this time, the user's rating prediction model for the product is revised as:

$$\hat{r}_{ui} = [X_U + \frac{\sum_{j \in N(u)} Y_j}{\sqrt{|N(u)|}}] X_I + u + b_u + b_i$$

In the formula, $N(u)$ represents the set of all commodities evaluated by user u.

The basic idea of the TimeSVD algorithm is that over time, the user's preference for the product also changes, that is, X_U、b_u and b_i in the above formula are no longer fixed quantities, but a function of time, but the feature vector of the product does not vary. Change of time. It can be seen that the TimeSVD algorithm incorporates time series items, so in the process of predicting the user's rating of the product, it is necessary to divide multiple periods along the time axis and predict the product rating in each period. The divided period is represented by Bin(t). The values of X_U、b_u and b_i are the same in the same period but are different in different periods. The scoring prediction process of the TimeSVD algorithm considering time series items is as follows:

$$\hat{r}_{ui} = [X_U(t) + \frac{\sum_{j \in N(u)} Y_j}{\sqrt{|N(u)|}}] X_I + u + b_u(t) + b_i(t)$$

where, $b_i(t)$ and $b_u(t)$ respectively represent the bias of the product and the user at time t, and both are composed of a static part and a dynamic part, namely:

$$b_i(t) = b_i + b_{i,\ Bin(t)}$$

$$b_u(t) = b_u + \alpha_u \cdot sign(t - t_u) \cdot |t - t_u|^{\beta} + b_{ut}$$

In the formula, $b_{i,\ Bin(t)}$ represents the bias of the product in the period Bin(t), and t_u represents the average value of all ratings given by the user.

After obtaining the predicted value of the score, the TimeSVD algorithm minimizes the error between the predicted value and the true value through a formula:

$$minF = \sum_{r_{ui}} (\hat{r_{ui}} - r_{ui})^2 + \lambda[\| X_U \|^2 + \| X_I \|^2 + \| b_u(t) \|^2 + \| b_i(t) \|^2 + \| Y_j \|^2]$$

5. Experimental analysis

5.1. Experimental conditions

To verify the effectiveness of the personalized product recommendation model of the deep neural network and dynamic collaborative filtering (DEEP-TimeSVD) proposed in this article, the sub-datasets Toys_and_Games and Instant_Video in the Amazon public data set are used, and about 78.3% of the product reviews in the two data sets. The number of product reviews is less than 20, and 67.1% of the product reviews have less than 10, which meets the user's requirement for sparse product reviews.

In subsequent experiments, the mean square error (MSE) is used as an indicator of the accuracy of the evaluation model for predicting product ratings. The indicator is defined as follows:

$$MSE = \frac{1}{N} \sum_{u,i} (\hat{r_{ui}} - r_{ui})^2$$

In the formula, N represents the number of samples in the experiment. The smaller the MSE of the model, the better the prediction performance of the model's product score.

The specific parameters of the experimental environment in the article are shown in [Table 1].

Table 1. Experimental environment parameters

Category	Parameter
The operating system	Windows 10
Programming platform	Pycharm 2019
A programming language	Python
CPU	Core i7
memory	8GB
storage	2TB

5.2. Text processing performance experiment

Aiming at the problem of scoring prediction of review text, the paper proposes to use BERT mapping combined with two-way GRU (BERT-GRU) to propose review text features. This section compares it with different text processing models, including pre-trained word vector combined with CNN model (Word-CNN), pre-trained word vector combined with CNN model (Word-GRU), BERT mapping combined with CNN model (BERT-CNN) The performance comparison results of the four text processing models are shown in [Table 2].

According to the text processing performance results of the four models, the text processing performance of the text processing model with BERT mapping has been further improved compared with the traditional pre-trained word vector model. This is because the traditional

word vector model cannot accurately give the true meaning of the same word in different environments under the influence of polysemous words, and the use of BERT mapping can effectively solve this problem and improve the semantic understanding of the review text. Comparing BERT-CNN and BERT-GRU, it can be seen that the two-way GRU model has better text processing performance than the traditional CNN model. This is because the review set processing of a certain product is essentially a sequence processing problem, and the RNN recursive structure in the two-way GRU is more suitable for characterizing sequence features than the CNN structure, which can effectively extract the links between review texts and improve text processing performance.

Table 2. Comparison results of text processing performance

Models	Toys_and_Games	Instant_Video
Word-CNN	0.86	0.93
Word-GRU	0.76	0.94
BERT-CNN	0.74	0.94
BERT-GRU	0.73	0.93

5.3. Model parameter selection experiment

This section tests the hidden factors, period division, and learning rate involved in the model, and analyzes the influence of different parameter values on the experimental results. The test range of the hidden factor is [5;10;15;20], and the test range divided by the period is [10;15;20;25]. [Figure 3] shows the change of the MSE index of the DEEP-TimeSVD algorithm proposed in this paper under different hidden factors. The results show that when the hidden factor value is 15, the MSE of DEEP-TimeSVD algorithm for Toys_and_Games and Instant_Video are 0.85 and 0.93, respectively, which both reach the minimum. Comparing Figure 3, we can see that when the hidden factors are 5, 10, and 20, the MSE of the DEEP-TimeSVD algorithm does not change very much. It shows that the DEEP-TimeSVD algorithm has low sensitivity to the setting of the hidden factor. Considering the recommended performance, the hidden factor a is set to 15 in the subsequent experiments.

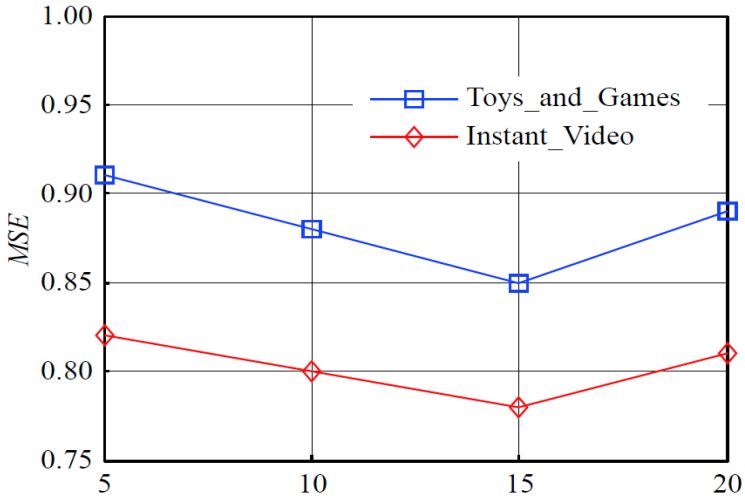

Figure 3. The impact of hidden factors on algorithm performance

[Figure 4] shows the changes in the DEEP-TimeSVD algorithm MSE indicators divided by different periods. The results show that the Toys_and_Games $Bin(t)$ and Instant_Video data sets have the smallest MSE when it is 20. For this reason, $Bin(t)$ is selected as 20 in this article.

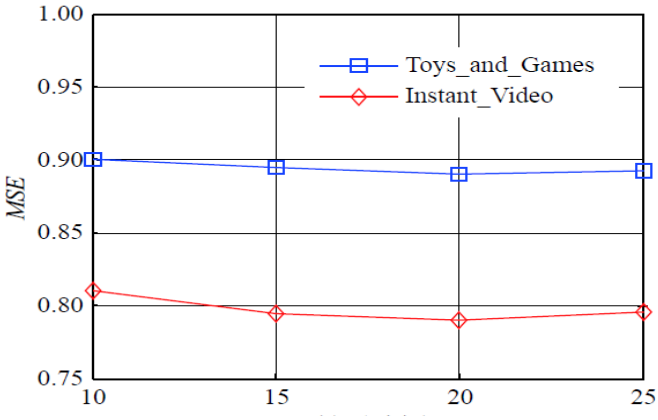

Figure 4. The impact of period division on algorithm performance

5.4. Comparison experiment of scoring prediction effect

This section still uses the Toys_and_Games and Instant_Video subsets in the Amazon public data set to evaluate the accuracy of scoring prediction recommendations. First, the BERT-GRU scoring prediction model proposed in the article is used to process the review text, and then personalized product recommendations are made according to the prediction results of the scoring matrix. The factorization machine (FM), SVD, and SVD models were used for comparative testing. In addition, the comparison method also includes the product recommendation algorithm proposed in the literature [13] that combines the edge noise reduction autoencoder and the FM model (mSDA-FM). In addition to MSE, the recommended evaluation index also uses average absolute error (MAE), which is defined as follows:

$$MAE = \frac{1}{N}\sum_{u,i}(\hat{r}_{ui} - r_{ui})$$

The DEEP-TimeSVD algorithm proposed in this article and the comparison algorithms are used to process Toys_and_Games and Instant_Video respectively, and the results are shown in Figure 5 and Figure 6.

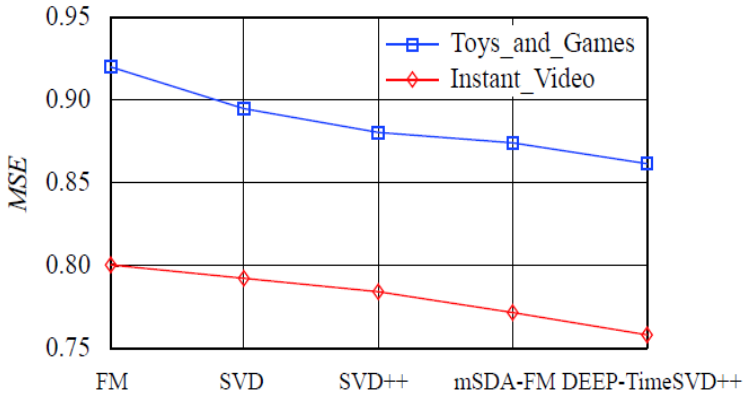

Figure 5. Comparison of MSE results

The experimental results show that the DEEP-TimeSVD algorithm proposed in this paper is better than the comparison method on the two performance indicators of MSE and MAE. From Figure 5, it can be seen that on the MSE indicator, the DEEP-TimeSVD algorithm has better performance on Toys_and_Games and Instant_Video. mSDA-MF increased by 1.4% and 3.8%, respectively. It can be seen from Figure 6 that the DEEP-TimeSVD algorithm has improved the MAE index by 2.5% and 2.2%, respectively, on the SVD algorithm with better performance compared to Toys_and_Games and Instant_Video. The above experimental results confirm The DEEPTimeSVD algorithm has higher accuracy in product recommendation.

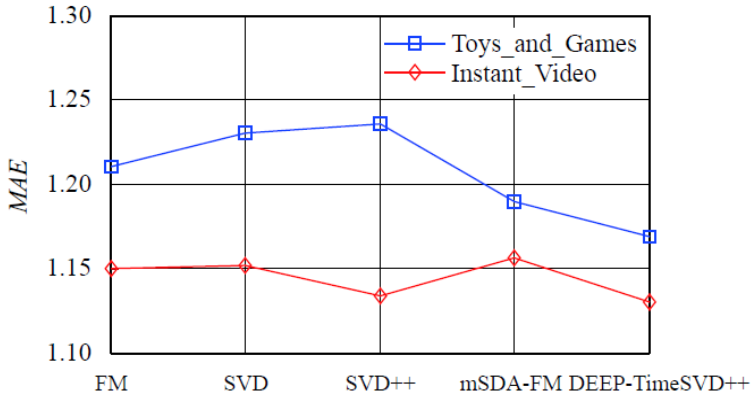

Figure 6. Comparison of MAE results

6. Conclusion

This paper proposes a personalized product recommendation model based on deep neural networks and dynamic collaborative filtering. This model uses the BERT model and two-way GRU to replace the traditional word vector text processing method, which can effectively reduce the impact of data sparsity and achieve deeply hidden feature extraction. The scoring matrix is predicted by the coupled CNN network, and the product scoring prediction is realized based on the dynamic collaborative filtering of the fusion time series items. The experimental results verify that the model reduces the impact of data sparsity and cold start problems, takes into account the changes in user interests over time, and has obvious advantages in recommendation accuracy.

Reference

[1] V. Subramaniyaswamy, G. Manogaran, and R. Logesh, "An ontology-driven personalized food recommendation in IoT-based healthcare system," J Supercomput, vol.75, pp.3184-3216, **(2019)**, DOI: 10.1007/s11227-018-2331- 8

[2] X. Feng, X. U. Yixiong, and Y. Zeng, "Dual-channel CNN recommendation algorithm combining user and product reviews," Modern Electronics Technique, **(2019)**

[3] P. Zhang, Z. Zhang, and T. Tian, "Collaborative filtering recommendation algorithm integrating time windows and rating predictions," Applied Intelligence, **(2019)**

[4] B. Cheng, D. O. E-Commer Ce, "Research on e-commerce personalized recommendation algorithm based on collaborative filtering," Modern Electronics Technique, **(2019)**

[5] Y. Han, C. Han, and L. Liu, "Collaborative filtering recommendation algorithm based on score matrix filling and user interest," Computer Engineering, **(2016)**

[6] F. Yang, Y. Zheng, and C. Zhang, "Hybrid recommendation algorithm based on probability matrix factorization," Journal of Computer Applications, **(2018)**
[7] G. Liu, K. Meng, and J. Ding, "An entity-association-based matrix factorization recommendation algorithm," Computers, Materials and Continua, vol.58, no.1, pp.101-120, **(2019)**
[8] K. Lee and Suk, "Quantitation of hypoechoic lesions for the prediction and Gleason grading of prostate cancer: A prospective study," World Journal of Urology, **(2018)**
[9] J. L. Che, L. W. Tang, and S. J. Deng, "Chinese word segmentation based on bi-directional GRU-CRF model," Fire Control and Command Control, **(2019)**
[10] W. Wang, Y. Sun, and Q. Qi, "Text sentiment classification model based on BiGRU-attention neural network," Application Research of Computers, **(2019)**
[11] F Kokkinos and A. Potamianos, "Structural attention neural networks for improved sentiment analysis," **(2017)**
[12] O. Xi, Z. Pan, and H. L. Cheng, "Sentiment analysis using convolutional neural network," IEEE International Conference on Computer and Information Technology Ubiquitous Computing and Communications Dependable, IEEE, **(2015)**
[13] M. Frans, M. Siti, and P. Setia, "Sentiment analysis: A comparison of deep learning neural network algorithm with SVM and nave Bayes for Indonesian text," Journal of Physics Conference, 971:012049, **(2018**)

This page is empty by intention.