Joint Deep Network With Auxiliary Semantic Learning for Popular Recommendation

XINGKAI WANG[iD], **YIQIANG SHENG**[iD], **AND HAOJIANG DENG**

National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China
School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Yiqiang Sheng (shengyq@dsp.ac.cn)

ABSTRACT There is a cold-start problem in the recommendation system field, which is how to profile new users and new items. The popular recommendation algorithm is an important solution to the cold-start problem. In this paper, we propose a new joint deep network model with auxiliary semantic learning for the popular recommendation algorithm (DMPRA). First, we define the items with a large quantity of review data and high ratings as the popular recommended items. Second, we introduce text analysis into the popular recommendation algorithm. We use the optimized CharCNN networks to learn the auxiliary semantic vectors from the users' reviews. Then, we use the Factorization Machine (FM) component and deep component to learn the corresponding vector representations of the items' attribute features. We use convolution to simulate the interaction of hidden latent vectors. This method can make the vectors interact more satisfactorily than traditional interactive representation methods. Finally, we provide the users with a reasonable popular recommendation list. The experimental results show that our algorithm can improve the AUC (area under the ROC curve) and Logloss (cross-entropy) of the popular items' prediction. In addition, we provide relevant explanations for some useful phenomena.

INDEX TERMS Popular recommendation algorithm, joint deep network, semantic learning.

I. INTRODUCTION

Recommendation systems help people obtain desired results more easily. In fact, there are still many difficulties, such as the cold-start problem. It is difficult for the model to learn the representations of a new user. It is also difficult to learn the representations of the recommended items that are rarely scored. One solution is the popular recommendation algorithm. The algorithm can recommend a popular item list to users. Then, the recommendation system will switch to the personalized recommendation module when a certain quantity of user data is collected. There is a long-tailed distribution of rated items applicable to many commercial systems. The small number of most popular items has the majority of ratings [1]. The popular recommendation algorithm can solve parts of the recommendation problem [2]. However, popularly recommended items do not indicate that the majority of users like the item. Only those items that the majority of users really like should be recommended by popular recommendation algorithms. Those items promoted

by businesses for false propaganda should not be evaluated by many people before people realize that they are not worth being widely recommended. It should be realized in a short time that the items are not popular items, which can save time and efficiency for the users. By identifying the real popular recommendation list, we can improve the quality of the popular recommendation algorithm. Otherwise, it will waste the time of the users.

At present, an increasing number of users generate various comments when they use various mobile applications. Social psychology [3] shows that a user's satisfaction will affect the satisfaction of other users. Therefore, the analysis of users' comments has great importance. Researchers have proven that introducing comment information can improve the prediction of the recommendation system. The introduction of comment information explains why users give good or bad reviews. However, review information has not been introduced into the popular recommendation algorithm.

At present, the quality of items provided by the popular recommendation module is unbalanced, which makes it difficult to ensure the effect when new users join. False scores and early-stage merchant promotions lead to an increase in

The associate editor coordinating the review of this manuscript and approving it for publication was Jiankang Zhang[iD].

the rating of items over a short period of time. This introduces more people with different preferences. Eventually, the score of poor items will gradually decrease. For example, junk movies that have famous directors, actors, and many production funds can obtain a high viewing volume by good early promotion in the early stage. They would be proven as a junk movie after being watched by many people. However, it wastes the users' time. In this paper, we introduce text analysis into the popular recommendation algorithm, which can improve the quality of recommendations. The main contributions are summarized as follows:

1) This paper proposes a new joint deep network model with auxiliary semantic learning for the popular recommendation algorithm. First, we use items with a large quantity of review data and high ratings as popular recommended items. It is a more reasonable definition of popular items. We analyze the users' reviews in a fine-grained way by the optimized CharCNN networks. The networks can learn the auxiliary semantic vectors. Then, we select the important attributes of the items. We use the Factorization Machine (FM) component and deep component to learn the corresponding vector representations. We combine the vectors in the shared layer. Finally, we predict whether the item is a popular item. Our experimental results demonstrate that the algorithm can improve the quality of the popular recommendation module.

2) In the traditional neural network, there are several methods of vector interaction, such as element-by-element vector multiplication and vector union policy. However, vectors interact only once in those ways, which has the problem of insufficient vectors' interaction. In the shared layer at the top of the three models, we use the convolution to learn the appropriate representation of the item, which allows the hidden latent vectors to interact more sufficiently than traditional interactive representation methods. Our experiments show that the method is effective.

3) The effect of learning the auxiliary semantic vector from the users' reviews can influence the performance of the popular recommendation module. We notice some useful phenomena in the experiments. For example, the improvement on the Douban Dataset is larger than that on the Yelp Open Dataset and Amazon product data. We also provide relevant explanations.

The rest of our paper is organized as follows. Section II presents related work. Section III describes the proposed popular algorithm. Section IV introduces the detail of our experiments. Section V discusses results in this paper. The conclusions are presented in Section VI.

II. RELATED WORK
Recommendation system can be used to predict whether users are interested in a certain item under the current large-scale data flooding. The cold-start problem has always been the attention of the academic circles. In addition to the popular recommendation algorithm, there are many traditional solutions. The authors in [4] believed that the integration of language knowledge represents users' interests more effectively than traditional keyword-based profiles. They combined the machine learning algorithm with the relevance feedback method. They adopted a word sense disambiguation strategy based on vocabulary knowledge stored in the WordNet vocabulary database to obtain a semantic outline. They filled the missing score with context information. The author in [5] considered three cold-start problems: 1) suggestions for new users' existing projects; 2) suggestions for existing users' new projects; 3) suggestions for new users' new projects. They proposed a regression model based on predictive features. It could utilize all available information of users and projects, such as users' demographic information and project characteristics, to solve the cold-start problem. The algorithm was effectively extended to a linear function of the number of observations. The author in [6] suggested that users should fill in the mass of information at the beginning of their login, which could solve the cold-start problem. They developed an iterative optimization algorithm. The experimental results on three benchmark recommendation datasets showed that the proposed algorithm was superior to the existing cold-start recommendation methods. The author [7] used hidden Dirichlet distribution to cluster users, which could solve the cold-start problem in the APP recommended scene. The target users who seek advice were mapped to the potential groups. They estimated the likelihood that users would like the application by using the transfer relationship between the potential groups and the applications. The author in [8] proposed a book recommendation model that combined the datasets of students' course selection and collaborative filtering algorithm. It solved the cold-start problem that the target users did not have the borrowing records. Their model generated book recommendation lists for the target users by making use of the target users' course selection records and the borrowing datasets of the old users. They compared the effects of different parameter settings on the algorithm.

In recent years, many researchers used deep learning methods to solve the cold-start problem in the recommendation systems. The authors [9] proposed a new attention model to unify Collaborative Filtering Recommendation and Content-based Recommendation in both warm and cold scenarios. They proposed a novel cold sampling learning strategy. The authors in [10] proposed the Purchase Intent Session-bAsed algorithm. It is a content-based algorithm that predicts the purchase intent in cold start session-based scenarios. Experiments showed that PISA performed better than the competitive baseline when introducing new projects. The combination of PISA and baseline can further improve performance in non-cold start scenarios. The authors in [11] proposed a new framework of the hybrid recommendation system based on the automatic encoder, which combined information of users and projects.
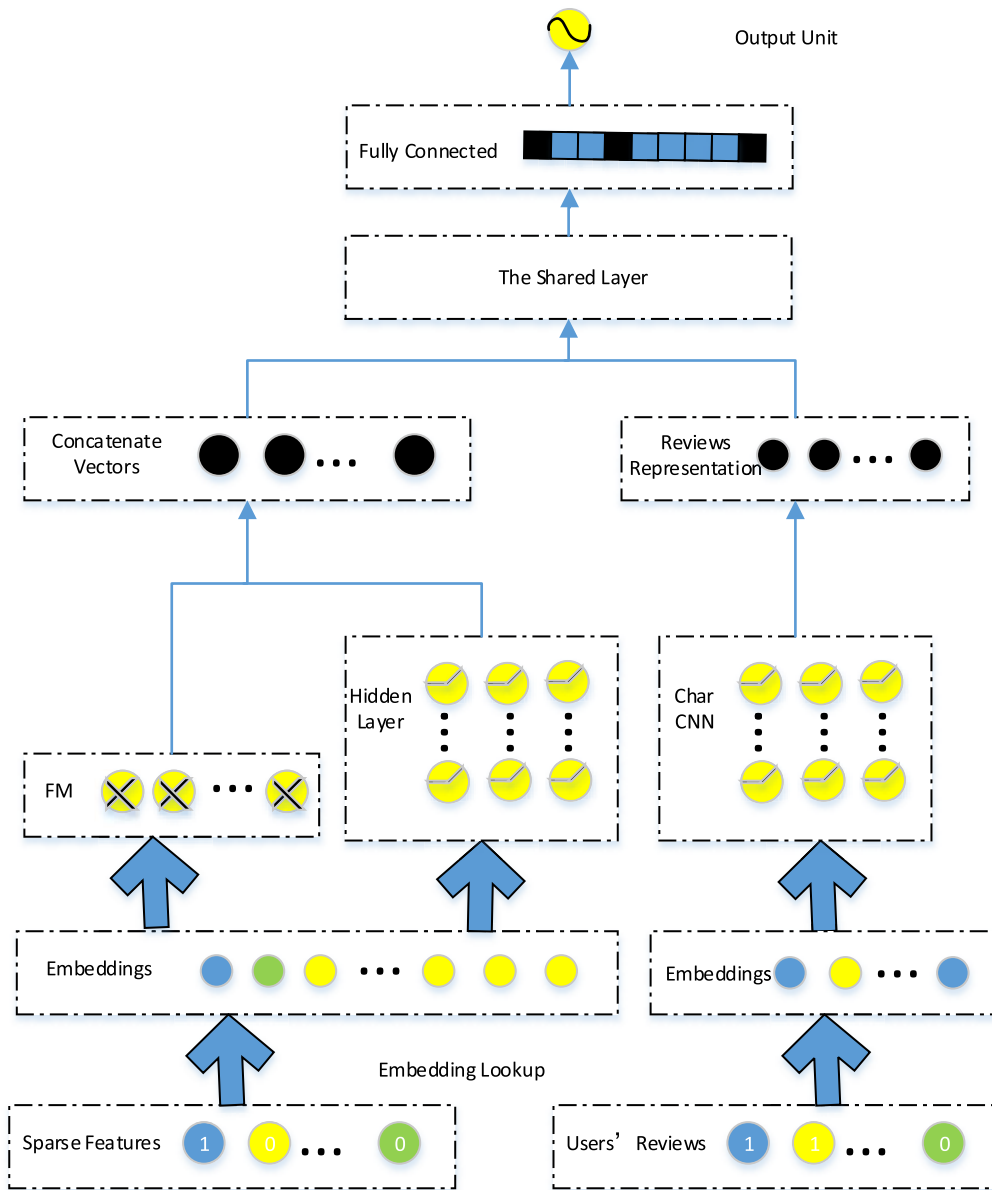
FIGURE 1. The popular recommendation algorithm. We reuse some symbols in [17].

Numerous scholars have found that text analysis played an important role in determining users' behavior and preferences. In the process of information recommendation, full consideration of users' emotional orientation and emotional state can better meet the needs of users. In 2007, the authors in [12] firstly analyzed the influence of the subjective and objective tendency of reviews. The authors in [13] proposed a recommendation system based on the sentiment analysis of sentences extracted from social networks. It is an algorithm that depends on the adverb found in the sentence. The authors in [14] proposed the Explicit Factor Model. It produced explainable suggestions while maintaining high prediction accuracy. The dominant features and user opinions of products were extracted through the phrase-level emotional analysis of user comments. The authors in [15] designed a recommendation system with semantic feature extensions. They extracted keywords by using jieba word segmentation system and sorted the keywords to expand the dimensions of the video feature vector.

In summary, there are many methods to address the cold-start problem. Many methods need corresponding preconditions. The popular recommendation algorithm is an algorithm without corresponding preconditions. In addition, text analysis technology can improve the quality of recommendation systems. However, text analysis technology has not been introduced into popular recommendation algorithms. We propose a more convenient and direct popular recommendation algorithm. It combines the users' reviews and the important

attributes of items. It can effectively use the wisdom of users and the attributes of items to provide users with a reasonably popular recommendation list.

III. PROPOSED RECOMMENDER ALGORITHM

In this paper, we propose a popular recommendation algorithm that combines the attributes of items and the features extracted from users' reviews. An implementation scheme of the recommendation algorithm is shown in Figure 1. The popular recommendation algorithm model consists of three parts: the FM component, the deep component, and the Char-CNN component. The FM component and deep component use the attributes of items as input. The CharCNN component uses the users' reviews as input. We summarize the stages of the popular recommendation algorithm in Algorithm 1. We discuss the design of the popular recommendation algorithm from scratch in this section.

Algorithm 1 Training Process
***
Input: Item data
Output: Predict whether the item is the popular item
  1: **Step 1:Data processing**
  2: Convert users' reviews to character level and extract attribute features of items;
  3: **Step 2:Create model**
  4: Create the popular recommendation model;
  5: **Step 3:Train and validate model**
  6: **while** stopping criterion is not met **do**
  7:     Train the model until convergence;
  8:     Minimize the training loss based on the output loss;
  9: **end while**
10: **Step 4:Test model**
11: Test fine-tuned model using the test dataset;
12: **return** compute the AUC.

A. THE EMBEDDING LAYERS

Our experimental datasets combine the users' reviews and the important attributes of items. We process the users' reviews as a sequence of encoded characters. We use CharCNN to learn deeper hierarchical features from the users' reviews. The important attributes of items are sparse. Another embedding layer is used to compress the input of attributes to low-dimensional vectors.

B. ARCHITECTURE

We aim to predict whether an item is a popular item. We propose a popular recommendation model. The model consists of three parts: the FM component, the deep component, and the CharCNN component. Each part learns the corresponding latent vector of the input. The shared layer on the top of the three parts is used to let the latent vectors interact. Finally, the combined prediction model predicts whether the item is a popular item.

1) FM COMPONENT

The authors in [16] proposed the FM component to learn feature interactions in the recommendation field. The FM component can allow parameter estimation under very sparse data. Since it has linear complexity by optimizing in the primal, it can capture order-2 feature interactions more effectively than previous approaches. The FM component of degree $d = 2$ is defined as follows.

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} <v_i, v_j> x_i x_j \quad (1)$$

Here, $w_0 \in \mathbb{R}$, $w \in \mathbb{R}^n$, $V \in \mathbb{R}^{n \times k}$, and $< ., . >$ is the dot product of two vectors of size k, is defined as follows.

$$<v_i, v_j> := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \quad (2)$$

The dot product of two vectors represents the impact of order-2 feature interactions.

2) DEEP COMPONENT

The deep component in our model is inspired by Cheng's method [18]. They applied a feed-forward neural network on the field embedding vectors. They converted the sparse high-dimensional features into a low-dimensional and dense real-valued vector. They fed the embedding vectors into the forward pass as follows.

$$a^{(l+1)} = f(W^{(l)} a^{(l)} + b^{(l)}) \quad (3)$$

Here, l is the layer number and f is the activation function. $a^{(l)}$, $b^{(l)}$, and $W^{(l)}$ are the activations, bias and model weights at l-th layer.

3) CHARCNN COMPONENT

Compared with other models that extract features from user reviews, the CharCNN component extends more easily to many human languages. We trained the networks by using gradient backpropagation [19]. The architecture of the CharCNN module is shown in Figure 2. We created an alphabet, including 26 English letters and 10 digits: Alphabet = "abcdefghijklmnopqrstuvwxyz0123456789". The frame of the input vectors is equal to the length of the alphabet. We used the alphabet to quantize each character by 1-of-m encoding. As CharCNN can learn simple quantization from textual signals, we fed this input to CharCNN without other normalization. The filter in Convolutional Neural Networks is defined as $w \in R^{(h*k)}$, where h is the number of vertical characters and k is the dimension of a character vector. The feature S_i is generated from a window of characters $a_{i:i+h-1}$. The process is defined as follows.

$$S_i = f(w * a_{i:i+h-1} + b) \quad (4)$$

Here, b is the offset and f is the activation function. Then we use max-pooling to extract the local optimal features.
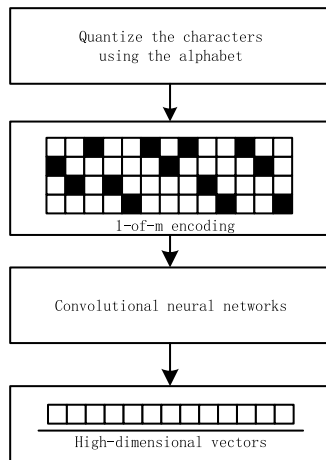
FIGURE 2. The CharCNN component which was inspired by the authors in [20].

The process is defined as follows.

$$f(x) = max(0, x) \tag{5}$$

We use fully connected layers to obtain high-dimensional vectors. We prove that the CharCNN module could extract semantic feature vectors effectively in [21].

4) THE SHARED LAYER

The three outputs of the above three components represent the features of items. Since they can be viewed in different feature spaces, they cannot be comparable. Thus, we map them into the same feature space by the shared layer. We construct the vectors' interaction as interactive representations. Each element produces only one interaction in the traditional interactive representations, such as concatenating x_i and x_j into a single vector $\hat{z} = (x_i, x_j)$.

Some researchers found that they should model all nested variable interactions. The authors in [22] used FM [23] as the estimator of the corresponding rating. They put $\hat{z}$ as the FM input. The authors in [24] added a two-way interaction layer in addition to applying DNNs on the embedding vector. Their model included both bitwise and vectorwise interaction, which is called PNN. We believe that there is insufficient interaction in the current interactive representations.

In the shared layer at the top of the three models, we convolve two vectors to learn the appropriate representation of the items' feature. We use the output of the CharCNN component as the input of convolution. We concatenate the output of the FM component and the output of the deep component as the convolution kernel of the convolution. The convolution process is shown in Figure 3. This approach has two advantages. First, it does not increase the scale of the parameters. Second, each element interacts many times during the interactive representations. Therefore, we can extract more diverse interactive representations by this approach. Since the step length of the convolution is 1, the dimension of convolution interactive representations is equal to the dimension

of convolution input. We use a fully connected layer to map the convolution output to the final output, which can predict whether the item is a popular item. We use the differentiation chain rule to compute the parameters in different layers.

IV. EVALUATION
A. DATASETS AND DATA PREPROCESSING
In this section, we experimented with the popular recommendation algorithm on the following three datasets.

1) YELP OPEN DATASET
The Yelp Open Dataset [25] has been widely used in the recommendation field. We used the tool – Yelp's Academic Dataset Examples [28] to convert the dataset from JSON to CSV format. We sorted the records in the review table by the key '*date*'. We recorded the number of reviews in the review table according to the key '*business_id*'. Since the business table had the key '*stars*', we deleted the column of keys '*stars*' in the review table. We merged the information into the business table and the review table by associating the primary key '*business_id*'. Then, we stored them in the same table structure to assure that each line of data combined the users' reviews and the important item attributes. To match as much data in the real scene as possible, we defined the proportion of positive samples to approximately 0.05. We chose the items for which the value of the key '*stars*' was more than 4, and the value of the number of reviews was more than 68 as the popular items in the Yelp Open Dataset.

2) AMAZON PRODUCT DATA
The Amazon product data [26] contain product reviews and metadata from Amazon. We chose some subsets for experimentation. We used '*reviews_Video_Games.json.gz*' and '*metadata.json.gz*' (3.1GB). We converted the two datasets from JSON to CSV format. First, we sorted the records in the review by the key '*reviewTime*'. We recorded the number of the items' reviews according to the key '*asin*'. Then, we computed the average rating. We extracted the digital information from the key '*salesRank*', which was used as the new key '*rank*' representing the sales ranking. Next, we merged the two tables by associating primary keys '*asin*' to ensure that each line of data combined the users' reviews and the item attributes. We also defined the proportion of positive samples as approximately 0.05. We chose the items for which the value of the key '*rank*' was less than 2,000, the value of the number of reviews was more than 40 and the value of the key '*overall*' was more than 4 as the popular items in the Amazon product data.

3) DOUBAN DATASET
The Douban Dataset [27] was collected from the Douban website. It is an important social app in China. The user group included professional film critics and ordinary film viewers. The datasets that we collected from the website included name, director, actor, screenwriter, rating, and users' reviews.
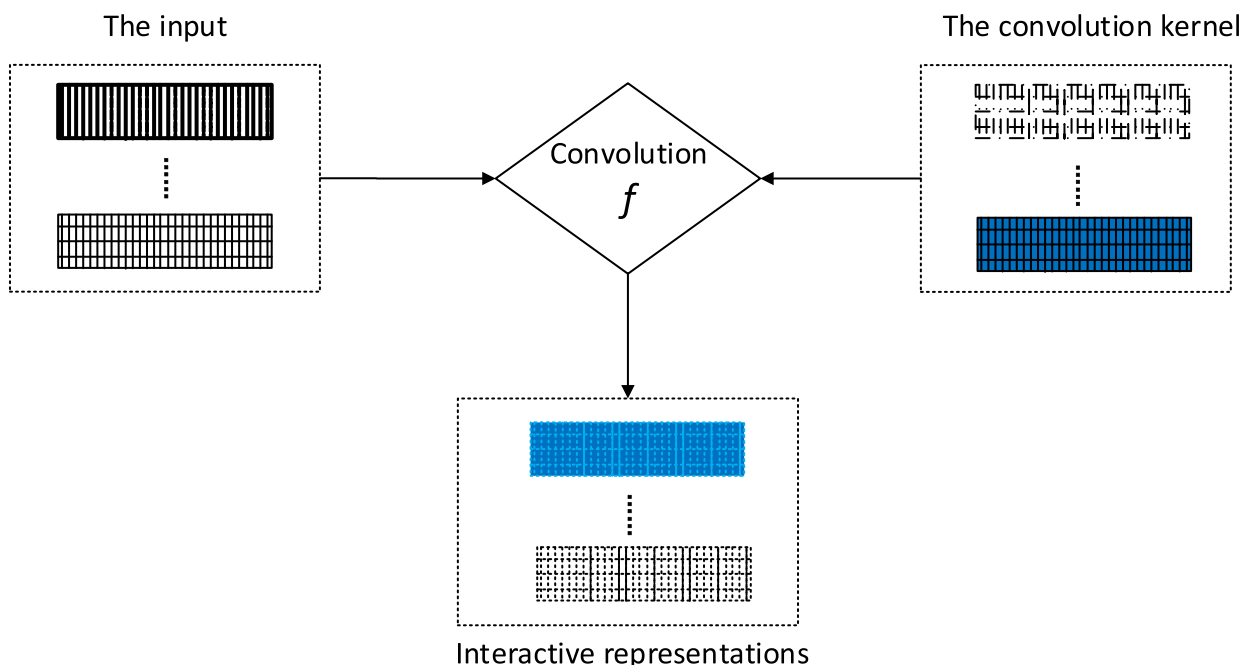
The input   The convolution kernel

FIGURE 3. Convolution the latent vectors.

TABLE 1. The statistics of the datasets.

class	Positive	Negative	Total
Yelp Open Dataset	10818	181788	192606
Amazon product data (selected)	509	10163	10672
Douban Dataset	379	6390	6769

Ten was the full rating, and 6 was the passing rating of a movie in the Douban score. If the average rating of a film was lower than 6, it indicated that the film was not in line with the expectation of the users. In the off-line experiment, we selected films with scores greater than 7 and more than 20,000 reviews as popular films. Other films were negative samples in our off-line experiments. The statistics of the datasets are shown in Table 1.

We wanted to prove that our algorithm improves the quality of the popular recommendation module by analyzing the comments of a small number of users. Since later reviews could cause a posterior probability problem, we selected and analyzed the early reviews of the corresponding films according to the time. Additionally, it matched the usage in real scenes. We selected one-tenth of the total reviews. At the same time, we chose the important item attributes. The training set used 80% of each dataset. We used 10% of each dataset as the validation set. The rest were used for the test set. We predicted whether the item was a popular item.

B. METRICS
We used two metrics to measure the quality of the recommendation algorithm: AUC (area under the ROC curve) and Logloss (cross-entropy). The AUC was used to measure the quality of the binary classification model. The classifier with a larger AUC is more effective. Logloss was used to measure the distance between the predicted result and the true label.

C. BASELINES
We compared the performance of our popular recommendation algorithm (DMPRA) with three models: FM, DNN and DeepFM. Researchers [17] compared the performance of their DeepFM model with several models: LR, FM, FNN, PNN (three variants) and Wide&Deep. Moreover, they found that the DeepFM performed better than the others in the AUC and Logloss for CTR prediction, which could prove that the model was a highly competitive baseline. Therefore, we used it as a baseline in this paper's scene for comparison. In addition, to prove the effectiveness of the joint model, we also showed the individual results of the two components: FM and DNN.

D. EXPERIMENT
The best experimental results of the popular recommendation algorithm were compared with the baselines in this section. We also tuned the effects of the baselines to their best performance. The length of the CharCNN model's input was set according to the experimental datasets and the experimental environment. The length l was 3,000 in the Yelp Open Dataset. The length l was 3,500 in the Amazon product data. The length l was 7,000 in the Douban Dataset. It was ignored when the length of the CharCNN's input exceeded l. We used the zeros to fill the input when the length was less than l. The results are shown in Table 2. It shows the performance of AUC and Logloss on the three datasets with

TABLE 2. Performance on the popular recommendation algorithm.

Model name	AUC	Logloss
Yelp Open Dataset		
FM	0.9026	0.2561
DNN	0.9162	0.2312
DeepFM	0.9198	0.2283
DMPRA	**0.9278**	**0.2156**
Amazon product data (selected)		
FM	0.7682	0.2193
DNN	0.7763	0.1928
DeepFM	0.7801	0.1786
DMPRA	**0.7886**	**0.1761**
Douban Dataset		
FM	0.6376	0.1189
DNN	0.6595	0.1025
DeepFM	0.6669	0.0883
DMPRA	**0.7346**	**0.0656**

TABLE 3. Comparing AUC of the model's variants.

Model	Yelp Open Dataset	Amazon product data (selected)	Douban Dataset
FM&DNN(concat)	0.9198	0.7801	0.6669
FM&DNN(FM)	0.9152	0.7756	0.6636
FM&DNN(CNN)	**0.9233**	**0.7846**	**0.7173**
DMPRA(concat)	0.9278	0.7886	0.7346
DMPRA(FM)	0.9236	0.7847	0.7317
DMPRA(CNN)	**0.9316**	**0.7931**	**0.7437**

different methods. We can see that the performance of our popular recommendation algorithm (DMPRA) is better than those methods. We can see that the improvement on the Douban Dataset is larger than that on the Yelp Open Dataset and Amazon product data. A possible explanation is that many restaurants have too few reviews, and many video games have too few reviews. The users' review information was introduced into the popular recommendation algorithm by extracting text features. Since the zeros could introduce more useless information, we chose the small length of the CharCNN's input on the Yelp Open Dataset and Amazon product data, which could influence the effect of learning auxiliary semantic vectors from the users' reviews.

To test the effect of the vector interaction, we also experimented with the different methods of interaction in the FM&DNN model, which named them FM&DNN (concat), FM&DNN (FM), and FM&DNN (CNN). In addition, we also distinguished three variants of DMPRA, which named them DMPRA (concat), DMPRA (FM), and DMPRA (CNN). We concatenated the output of the FM component and the output of the DNN component in FM&DNN (concat). We concatenated the output of three components in DMPRA (concat). FM&DNN (FM) and DMPRA (FM) were inspired by Zheng's work [22]. We concatenated the corresponding output. Then, we used it as the FM input. FM&DNN (CNN) and DMPRA (CNN) used the convolution interaction for vector interaction. Table 3 shows the experimental results. We can see that the performance of our convolution interaction method is better. The results show that our methods allow the hidden latent vectors to interact with each other more sufficiently.

V. DISCUSSION

It is difficult for the recommendation model to learn the representations of the new users and the representations of the recommended items that are rarely scored. Most solutions to solve the cold-start problem need corresponding preconditions. One solution is the popular recommendation algorithm without corresponding preconditions. However, the traditional popular recommendation algorithms provide

the items with unbalanced quality. It is difficult to ensure the effect when new users join. The users' comments will explain why users like the item or dislike the item. The text analysis can play a better role to improve the quality of the popular recommendation algorithms. Therefore, introducing the text analysis into the popular recommendation algorithm could provide a novel idea for the development of recommendation systems. The experimental results show progress made in the popular recommendation algorithm. This paper focuses on optimizing the popular recommendation algorithm by analyzing users' reviews. We proposed a new joint deep network model with auxiliary semantic learning for the popular recommendation algorithm. We used optimized CharCNN networks to analyze the users' reviews. We used the FM component and deep component to learn the corresponding vector representations of the items' attribute features. We combined the vectors in the shared layer. We predicted whether the item was a popular item. However, some challenges remain. The types of movies and the viewing habits of users in different regions and ages have not been fully considered. Considering more evaluation indicators will make the recommendation algorithm more powerful. For example, efficiency is a key factor in a large-scale recommendation scene. In addition, the detection of false comments can be taken into account.

VI. CONCLUSION AND FUTURE WORK

In order to recommend a better popular item list to users, we have proposed the popular recommendation algorithm. We introduced the text analysis into the popular recommendation algorithm. The optimized CharCNN networks were used to learn the auxiliary semantic vector from the users' reviews. We used FM component and deep component to learn the corresponding vector representations of the items' attribute features. We used the shared layer at the top of the three components to model all variable interactions. The shared layer used the convolution to let the hidden latent vectors interact with each other. In addition, we have provided relevant explanations for some useful phenomena in our experiments.

Our method performed better than the baseline methods in the popular recommendation field. The future work will research on choosing appropriate users' review.

REFERENCES

[1] C. Anderson, *The Long Tail: Why the Future of Business is Selling Less of More*. New York, NY, USA: Hachette Books, 2006.

[2] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, 2010, pp. 1–11.

[3] J. Masthoff and A. Gatt, "In pursuit of satisfaction and the prevention of embarrassment: Affective state in group recommender systems," *User Model. User-Adapted Interact.*, vol. 16, nos. 3–4, pp. 281–319, Aug. 2006.

[4] M. Degemmis, P. Lops, and G. Semeraro, "A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation," *User Model. User-Adapted Interact.*, vol. 17, no. 3, pp. 217–255, Mar. 2007.

[5] S.-T. Park and W. Chu, "Pairwise preference regression for cold-start recommendation," in *Proc. 3rd ACM Conf. Recommender Syst. (RecSys)*, 2009, pp. 1–8.

[6] K. Zhou, S.-H. Yang, and H. Zha, "Functional matrix factorizations for cold-start recommendation," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. - SIGIR*, 2011, pp. 1–10.

[7] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua, "Addressing cold-start in app recommendation: Latent user models constructed from Twitter followers," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2013, pp. 1–10.

[8] J. Qi, S. Liu, Y. Song, and X. Liu, "Research on personalized book recommendation model for new readers," in *Proc. 3rd Int. Conf. Inf. Syst. Eng. (ICISE)*, May 2018, pp. 78–81.

[9] S. Shi, M. Zhang, Y. Liu, and S. Ma, "Attention-based adaptive model to unify warm and cold starts recommendation," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2018, pp. 127–136.

[10] M. Shekasta, G. Katz, A. Greenstein-Messica, L. Rokach, and B. Shapira, "New item consumption prediction using deep learning," 2019, *arXiv:1905.01686*. [Online]. Available: https://arxiv.org/pdf/1905.01686.pdf

[11] L. Yu, W. Shuai, and M. S. Khan, "A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering," *Big Data Mining Anal.*, vol. 1, no. 3, pp. 211–221, 2018.

[12] A. Ghose and P. G. Ipeirotis, "Designing novel review ranking systems: Predicting the usefulness and impact of reviews," in *Proc. 9th Int. Conf. Electron. Commerce (ICEC)*, 2007, pp. 1–7.

[13] R. Guimaraes, D. Z. Rodriguez, R. L. Rosa, and G. Bressan, "Recommendation system using sentiment analysis considering the polarity of the adverb," in *Proc. IEEE Int. Symp. Consum. Electron. (ISCE)*, Sep. 2016, pp. 71–72.

[14] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2014, pp. 83–92.

[15] X. Wang, Y. Sheng, H. Deng, and Z. Zhao, "Design of video recommendation system with extended attributes of chinese semantic feature," *J. Netw. New Media*, vol. 7, no. 39, pp. 51–55, 2018.

[16] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 995–1000.

[17] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," 2017, *arXiv:1703.04247v1*. [Online]. Available: https://arxiv.org/pdf/1703.04247v1.pdf

[18] H.-T. Cheng, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, H. Shah, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, and W. Chai, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst. (DLRS)*, 2016, pp. 7–10.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[20] X. Zhang and Y. LeCun, "Text understanding from scratch," 2015, *arXiv:1502.01710*. [Online]. Available: https://arxiv.org/pdf/1502.01710.pdf

[21] X. Wang, Y. Sheng, H. Deng, and Z. Zhao, "CharCNN-SVM for Chinese text datasets sentiment classification with data augmentation," *Int. J. Innov. Comput., Inf. Control*, vol. 15, no. 1, pp. 227–246, 2019.

[22] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2017, pp. 425–434.

[23] S. Rendle, "Factorization machines with libFM," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, p. 57, 2012.

[24] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based neural networks for user response prediction," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1149–1154.

[25] *Yelp Open Dataset*. Accessed: Jun. 18, 2019. [Online]. Available: https://www.yelp.com/dataset/download

[26] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. - SIGIR*, 2015, pp. 43–52.

[27] *Douban Movie Dataset*. Accessed: Jun. 18, 2019. [Online]. Available: https://movie.douban.com/

[28] *Yelp's Academic Dataset Examples*. Accessed: Jun. 18, 2019. [Online]. Available: https://github.com/Yelp/dataset-examples

XINGKAI WANG was born in Shandong, China, in 1992. He is currently pursuing the Ph.D. degree with the National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China. His research interests include machine learning, natural language processing, and the mainly application area is recommendation.

YIQIANG SHENG received the master's degree from Nankai University, Tianjin, China, in 2003, and the Ph.D. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 2014. He was with Sanyo Electric Company, Ltd., Japan, from 2003 to 2008, and with Nokia, Japan, from 2008 to 2010. He conducted research on communications and integrated systems at Osaka University, from 2010 to 2011 and at Tokyo Institute of Technology, from 2011 to 2014. He is currently an Academic Researcher with the Chinese Academy of Sciences. His current research interests include machine learning, distributed big data, the future Internet, intelligence-enabled networking, multimedia service, optimization algorithms, and integrated smart systems.

HAOJIANG DENG received the M.S. degree from the 510 Institute of China Academy of Space Technology, Beijing, China, in 1998, and the Ph.D. degree from the Institute of Semiconductors, Chinese Academy of Sciences, Beijing, in 2001. He is a Full Professor. His current research interest includes digital signal processing in audio and video and broadband multimedia communication.

• • •