JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Personalized Product Recommendation Method for Analyzing User Behavior Using DeepFM

Jianqiang Xu*, Zhujiao Hu**, and Junzhong Zou*

Abstract

In a personalized product recommendation system, when the amount of log data is large or sparse, the accuracy of model recommendation will be greatly affected. To solve this problem, a personalized product recommendation method using deep factorization machine (DeepFM) to analyze user behavior is proposed. Firstly, the K-means clustering algorithm is used to cluster the original log data from the perspective of similarity to reduce the data dimension. Then, through the DeepFM parameter sharing strategy, the relationship between low- and high-order feature combinations is learned from log data, and the click rate prediction model is constructed. Finally, based on the predicted click-through rate, products are recommended to users in sequence and fed back. The area under the curve (AUC) and Logloss of the proposed method are 0.8834 and 0.0253, respectively, on the Criteo dataset, and 0.7836 and 0.0348 on the KDD2012 Cup dataset, respectively. Compared with other newer recommendation methods, the proposed method can achieve better recommendation effect.

1. Introduction

E-commerce is a major product of the popularisation and maturity of the Internet. A personalized product recommendation system is of great significance to e-commerce and online shopping [1-4]. The so-called personalized product recommendation system identifies the product set of interest to specific users, thus mining the potential needs of users and helping users to make decisions [5,6]. The performance of a personalized product recommendation system depends on the mining of user behavior relationships. Behavior relationships with personalized information can better reveal useful information hidden in the user's historical behavior [7,8], thus significantly improving the performance of the system [9,10]. Therefore, an excellent recommendation system can help users quickly and accurately find the goods they are interested in and improve the service efficiency of the e-commerce platform, which is of great significance to promote the development of e-commerce.

This study mainly aims at the personalized product recommendation field of electronic commerce and

proposes a personalized product recommendation method using a deep factorization machine (DeepFM) to analyze user behavior. The reliability of the method is verified by experiments. It is worth noting that the recommendation system algorithm described herein is general and can also be applied to other recommendations.

2. Related Works

How to mine effective behavior relationships from large-scale high-dimensional sparse user behavior data and quickly generate accurate personalized recommendations for users has become an urgent problem to be solved in the personalized product recommendation field. In response to these problems, scholars have proposed many methods [11–25]. For example, [11] proposes a personalized product recommendation method based on deep learning and multi-view information fusion, which maps multi-view information into a unified potential space through an automatic encoder, adds a cognitive layer to depict different cognitive styles of consumers, and introduces an integrated module to reflect the interaction of multi-view potential representations, thus improving the accuracy of personalized product recommendation of the model. [12] proposes a personalized product recommendation method on improved fuzzy C-means (FCM). The traditional FCM clustering algorithm is improved through membership adjustment and density function. Individual preferences are divided into different groups and a personalized knowledge base is established according to association rules. [13] proposes a personalized product recommendation method based on the user model and the user-item matrix. By combining the aforementioned, the user attribute data, and the value of user score changing with time are introduced into the traditional recommendation method. The interactive personalized product recommendation method based on hybrid algorithm model proposed in [14] obtains a list of original recommendation results through various recommendation algorithms, determines the weight of the recommendation algorithm by combining interactive mode with customer feedback, and fuses the original results with the comprehensive formula of evidence theory. [15] proposes a hybrid method of personalized product recommendation. By using implicit details and mining click flow paths of users with the same interests, users' preference level is predicted, and the similarity between products is then extracted using a sequence model. [16] proposes a collaborative filtering personalized product recommendation method based on user portraits. By establishing a mathematical model, the discrete volume correlation theory is used to improve the similarity of commodities, and the weighted approximation of user preferences is used to recommend commodities to customers according to the nearest neighbour similarity. However, these methods all have the problem of sparse data [17–21], that is, when the data are sparse, the calculated similarity is not sufficiently accurate.

To solve the problem of data sparseness, scholars have also proposed a variety of methods. For example, [17] proposed a method of personalized recommendation through matrix co-decomposition and tag and time information. According to the assumption that user preferences will change with time, preference and correlation are optimised by adding time influence, thus alleviating the problem of data sparseness. [18] proposes a personalized product recommendation method based on conditional variational automatic encoders (CVAE). It uses the fact that similar users tend to be interrelated in purchasing preferences, learns through tag verification signals, learns tags of different conditional attributes respectively, and then merges results from multiple prediction pools. [22] proposes a

personalized product recommendation method based on a product evaluation concept tree and system filtering. The similarity is optimised through the product evaluation concept tree, and concept similarity is used to replace item similarity, thus improving the sparsity of data. [23] proposes a cross-domain multi-dimension tenant factorization (CD-MDTF) method using multi-dimensional tensor decomposition, which can best balance the influence between domains through cross-domain collaborative filtering and alleviate the problems of data sparsity and cold start. [24] proposes a personalized recommendation method of multi-interest and multi-content hybrid collaborative filtering. By combining commodity- and user-based collaborative filtering, the similarity between target and other commodities is identified, and commodities are recommended according to the similarity between target users and active users. [25] proposes a personalized product recommendation method based on association sets. The implementation and requirements of the system are mapped through the policy module, and the whole system is designed as a standard component, thus adapting to the changes in recommendation policies. Although these methods alleviate the sparseness of data to some extent, they often only consider one factor and ignore other characteristics.

Based on the above analysis, it can be seen that although depth-based learning methods have strong learning ability and show certain effectiveness in recommendation effect, such methods rely on the quality of datasets to a large extent and ignore some features. Compared with other neural networks, DeepFM can learn low- and high-order features at the same time and can solve the problem of feature combination in large-scale sparse data. To solve the problems of large feature quantity, sparseness, and abnormal data in personalized recommendation methods based on log data, this study proposes a new method, which introduces an embedding layer into the neural network through DeepFM, compresses input vectors into dense vectors with low dimensions, and mines the hidden feature combination relationships in high-dimensional sparse datasets. In addition, so as to improve the recommendation accuracy of the model, the same objects are clustered to obtain a low-dimensional representation for high-dimensional data that can reflect the intrinsic structural characteristics of the original data. The main innovations of the proposed method are summarised as follows:

(1) Aiming at the problem of similarity in log datasets, a data dimensionality reduction method based on K-means object clustering is adopted. Using the number of advertisements shown as the weight of advertisement and query, the advertisement-query matrix is established and clustered to obtain a low-dimensional representation that can reflect the intrinsic structural characteristics of the original data.

(2) To solve the problem of sparse original datasets, the DeepFM depth learning method with parameter sharing structure is adopted. By sharing the vectors that decompose the second-order feature relation matrix with the embedded layer vectors of the neural network, it is easier for the neural network to learn the higher-order feature combination relation through multilayer connection and nonlinear transformation.

3. Problem Description and Algorithm Framework

3.1 Problem Description

A personalized recommendation can be regarded as a click-through rate prediction problem, in which

the relationship between features and feature combinations plays an important role in rate prediction. Among the existing click-through rate models, some only capture low-order feature combination relationships, some only capture high-order feature combination relationships, and some still need feature engineering. In view of the problems existing in these models, the key issue examined in this study is a deep click-through rate prediction model based on feature combination relationship mining.

3.2 Algorithm Framework

This section introduces the proposed personalized product recommendation method for analyzing user behavior by using DeepFM, which is mainly divided into two stages: the model training stage and the online prediction stage. In the model training stage, based on the historical behavior data of active users, by calculating the probability of a user clicking on a commodity in the context environment, since the input vector may contain continuous and discrete features, it is necessary to process the features of different domains to obtain the click rate prediction model. In the online prediction stage, after the user enters the recommendation system, the click rate prediction model obtained through training is used to predict the probability of a specific user clicking on a commodity in the context environment. Based on the predicted click rate, the products are recommended to the users in sequence according to the click rate.

The overall framework of this method is shown in Fig. 1. Firstly, the K-means algorithm is used to reduce the dimension of the original data. The purpose is to obtain a low-dimensional representation from the high-dimensional log data, which can reflect the intrinsic structural characteristics of the original data while reducing noise and sparsity. DeepFM is then used to extract low- and high-order feature combinations from the reduced dimension log data. DeepFM compresses the input vectors into low-dimensional dense vectors by introducing an embedding layer into the deep neural network, so as to further reduce noise and sparsity of the dataset. The extracted features are then trained and learned to form a click rate prediction model. The prediction model generates personalized merchandise recommendations and updates the original log dataset according to user feedback.
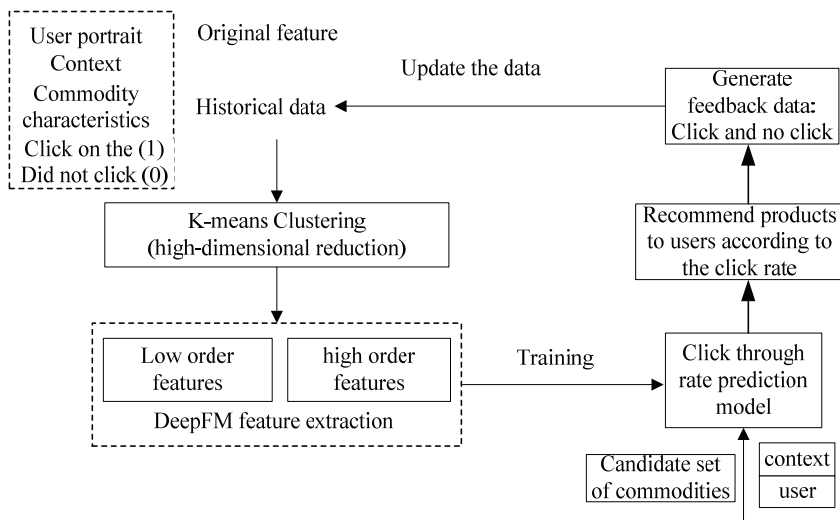

Fig. 1. Overall framework of the method.

4. Proposed Personalized Product Recommendation Method

4.1 Feature Dimension Reduction Based on Object Clustering

The click log data contains many types of objects such as users, queries, advertisements, and so on. The relationships among these objects are very complex. There are relationships within the same object, such as similarity relationships within advertising objects [26]. At the same time, there are complex relationships among different types of objects. For example, given a specific user and the query submitted by the user, it is necessary to estimate whether the user will click on the advertisement and the probability of doing so. There are complex implicit relationships among the user, the query, and the advertisement. According to the characteristics of click log data, this study reduces dimensions from the two angles of similarity relation between the same objects and association relation between different objects. Therefore, this study first reduces the dimensions of users, queries, and advertisements from the perspective of similarity.

The K-means clustering algorithm based on distance partition is used to cluster queries, advertisements, and users. The aim is to aggregate similar objects into the same cluster, and the similarity of objects in the same cluster is as high as possible to obtain initial aggregated data. The text similarity [27] technology is directly used to realise the clustering of queries and even advertisements. Although the clustering task can be completed, this method does not consider the correlation between advertisements and queries contained in actual data, and the obtained clustering results have no reference significance for click rate estimation. Therefore, to better mine and make use of the relationship between advertisements and queries, this study proposes a new method, using the number of ad impressions provided in the experimental data as the weight of advertisement A_i and query Q_J to build an advertisement-query matrix $W_{N_a \times N_q}$, where N_a represents the number of advertisements, N_q represents the number of queries, and ω_{ij} represents the weight between $\langle A_i, Q_J \rangle$. The K-means algorithm is used to cluster the advertising query matrix to obtain a relatively dense dataset. Taking advertisement clustering as an example, Fig. 2 introduces the process of the clustering algorithm, and adopts the same processing method for query clustering.

Based on the advertisement-query matrix $W_{N_a \times N_q}$, the clustering of advertisements/queries is completed through the clustering algorithm shown in Fig. 2, so that the similarity of advertisements/queries in the same cluster is as high as possible. The specific method in this study is to cluster advertisements and queries based on the same advertisement-query matrix. The two clusters are independent of each other, and the clustering order does not affect the subsequent calculation. For clustering of user dimensions, considering the similarity of users with similar query requirements, this study directly groups the users corresponding to queries in the same cluster into a user cluster according to the query clustering results obtained previously.

The number of users, queries, and advertisements in the initial data is represented by N_u, N_q, and N_a, respectively. After objects of the same type are internally clustered, objects belonging to the same cluster are represented by the same ID, and the clustered number of users, queries, and advertisements are represented by K_u, K_q, and K_a, respectively. In this way, the number of users, queries, and advertisements in the initial dataset is reduced from the original N_u, N_q, and N_a, to K_u, K_q, and K_a, respectively.

Assuming that T_q and T_a, respectively represent the number of iterations required to perform the clustering task by calling a clustering algorithm on the query and advertisement, and K represents the
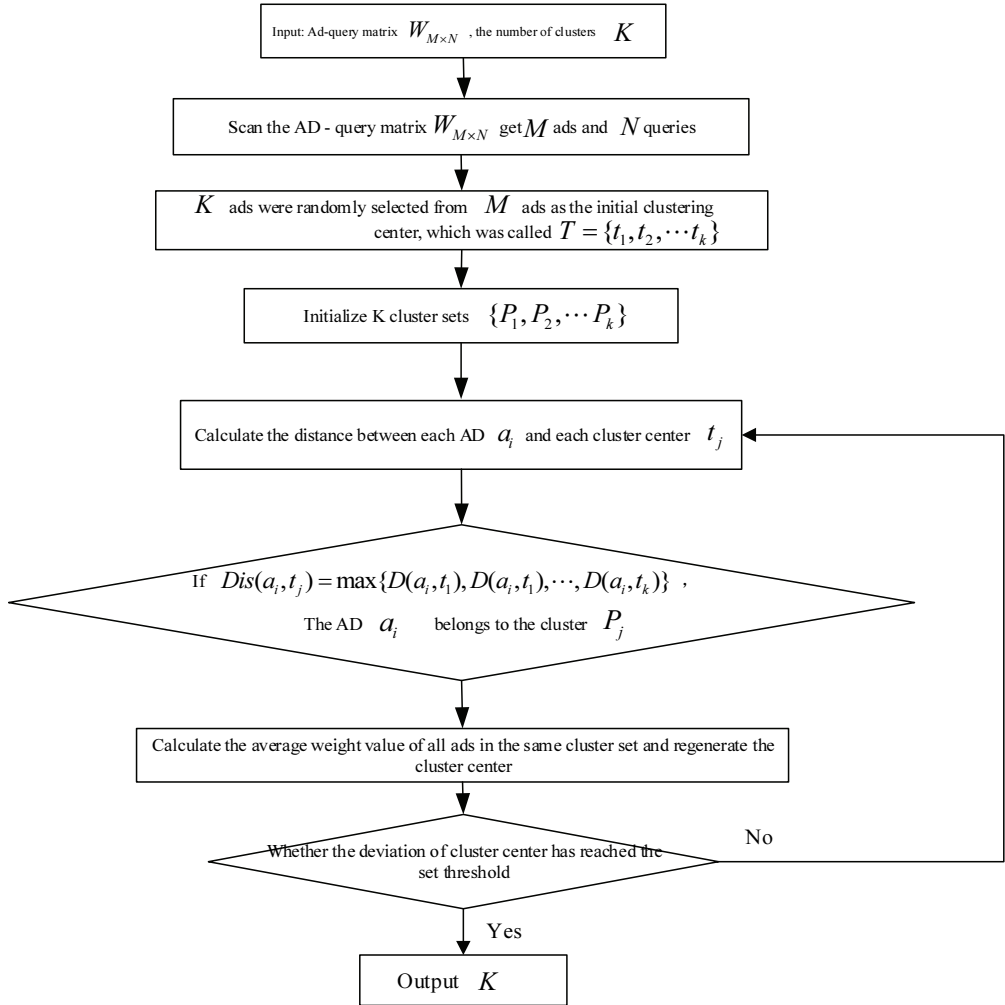
Fig. 2. The process of the clustering algorithm.

number of clusters, the time complexity of completing the clustering of the query is $O\left(KT_qN_q\right)$, and the time complexity of clustering is $O\left(KT_aN_a\right)$. Since the user's clustering is directly obtained according to the clustering result of the query instead of calling the algorithm of Fig. 2, and its complexity is $O\left(N_q+N_u\right)$, let $T=\max\left\{T_q,T_a\right\}$, $N=\max\left\{N_u,N_q,N_a\right\}$, then the complexity of the clustering link is expressed as

$$O\left(KT_qN_q\right)+O\left(KT_aN_a\right)+O\left(N_q+N_u\right)=O\left(KTN\right) \tag{1}$$

4.2 Personalized Recommendation Based on DeepFM

DeepFM is an end-to-end learning framework, which can be used not only for click rate prediction but also for other machine learning tasks. It is composed of two parts, namely, the decomposer and the deep neural network. For a certain feature i, w_i represents the weight of the i-th feature, and V_i represents the

k-dimensional implicit vector of the i-th feature. The V_i vector is input into the decomposition machine part and the deep neural network part. (1) In the decomposition machine part, V_i and other feature vectors interact with each other, and the inner product is used to measure the importance of the corresponding feature combination relationship, that is, fitting the second-order feature combination characteristic relationship. (2) Vector is input to the neural network, and the high-order feature combination relation is learned through a neural network connection and nonlinear transformation.

Parameters of neural network ($W^{(l)}$、$b^{(l)}$) and decomposer are optimised through joint learning. Formally, the DeepFM framework can be expressed as:

$$\hat{y}(x) = sigmoid \left(y_{FM}(x) + y_{Deep}(x) \right) \tag{2}$$

In the above formula, $\hat{y}(x) \in (0,1)$ is the click rate prediction value, $y_{FM}(x)$ is the output of the decomposer part, and $y_{Deep}(x)$ is the output of the deep neural network part.

The decomposer can be used to learn the second-order feature combination relation and the first-order feature weight. In the decomposer model, the second-order combined features can be easily captured by fitting the weights of the feature combination relations corresponding to the two features through the inner product of the implicit vectors corresponding to the two features. When the data are very sparse, the decomposer can achieve better results.

The decomposer decomposes the matrix of the second-order feature combination relation [28] and uses the implicit vector V_i and V_j of the feature to calculate the inner product as the weight of the feature combination relation. Specifically, whenever one of the features appears, the implicit vector corresponding to that feature will be trained. Even if the characteristic sum appears in certain test data at the same time, the decomposer can obtain a reasonable prediction value. Therefore, in the case of high-dimensional sparse data, those feature combination relationships that never or seldom appear in the training set can be better learned by the decomposer. The output of the decomposer is the sum of an additional unit and a series of multiplication units:

$$y_{FM}(x) = \langle w, x \rangle + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \langle V_i, V_j \rangle x_i \cdot x_j \tag{3}$$

Among them, for the sake of simplicity, the constant term is omitted, $w \in R^d, V_i \in R^k$ (k is a hyper parameter, designated by the user), the addition unit $\langle w, x \rangle$ reflects the weight of the first-order feature, and the multiplication unit represents the weight of the second-order feature. As shown in Eq. (2), the output of the decomposer $y_{FM}(x)$ module is part of the predicted click rate.

DeepFM's deep neural network part is a forward neural network, which is used to learn high-order feature combination relations [29]. The original data of click rate prediction contains many discrete features, which are often high-dimensional and sparse, resulting in difficulty in parameter training of deep neural networks. To solve this problem, DeepFM uses the embedding layer to compress the high-dimensional sparse input vector into a low-dimensional and dense real vector and then connect it to the hidden layer of the neural network.

There are two points that should be paid special attention in the embedding layer. (1) Although different

domains contain different feature dimensions, the embedding vectors of each domain have the same size. (2) The implicit vector in the decomposer is shared with the neural network, that is, as the network weight of the embedding layer, the input is compressed into the embedding vector. DeepFM does not need to train the decomposer first and then use the embedded layer parameters of the initialization neural network. In DeepFM, the decomposer will be used as a part of the model and trained together with other parameters. The input to the embedding layer is expressed as:

$$a^{(0)} = \left[e_1, e_2, \ldots, e_m \right]$$
(4)

where $e_i = V_{\text{field}_i} \cdot x_{\text{field}_i}$ is the embedding vector of the i-th domain, m is the number of domains, and V_{field_i} is the network parameter between the embedding layer and the input layer, where the i-th one-hot input vector of the second domain is x_{field_i}.

The DeepFM decomposer and deep neural network share the same embedding vector, which brings two advantages. (1) The embedding vector can learn the information of low- and high-order feature combination from the original data. (2) DeepFM does not need to specially design feature engineering to limit the structure of the neural network.

The logarithmic loss function is equivalent to relative entropy and represents the difference between the two distributions. Therefore, click rate prediction generally uses the logarithmic loss function as the prediction target:

$$e(x, y) = -\left(y \cdot \log\left(\hat{y}(x) \right) + (1 - y) \cdot \left(1 - \log(1 - \hat{y}(x)) \right) \right)$$
(5)

$$E = \sum_{(x,y) \in T} e(x, y)$$
(6)

Among them, (x, y) is a data sample, x is the feature vector, y is the class label of the sample, $\hat{y}(x)$ is the predicted value of the input vector, $e(x, y)$ is the log loss function of the predicted value and the correct result, and E is the log loss function of the entire dataset T.

To reduce training time, layer normalisation (LN) is adopted when DeepFM is deployed. For each sample, LN calculates the mean and variance for all inputs of neurons in a layer of the sample. The mean-variance is used to compress the inputs into distribution with a mean of 0 and a variance of 1. Then, to ensure the characteristics of different neurons, each neuron is assigned a rescaling and retranslating variable, and the input normalized in the previous step is rescaled and translated [30].

The detailed calculation flow based on DeepFM framework is shown in Fig. 3. Step 1, initialising parameters. Step 2, randomly disrupting the sample sequence to ensure the randomness of the data. Step 3, traversing the data by batch size. Step 4, complete the DeepFM prediction value. Step 5, calculating a logarithmic loss function. Step 6, calculate the gradient and update it.

5. Experimental Results and Analysis

To verify the effectiveness of the proposed method, experiments are carried out on the Criteo and KDD2012 Cup datasets. The computer processor model is 2.5 GHz Intel Core i5, with 8 GB of memory,
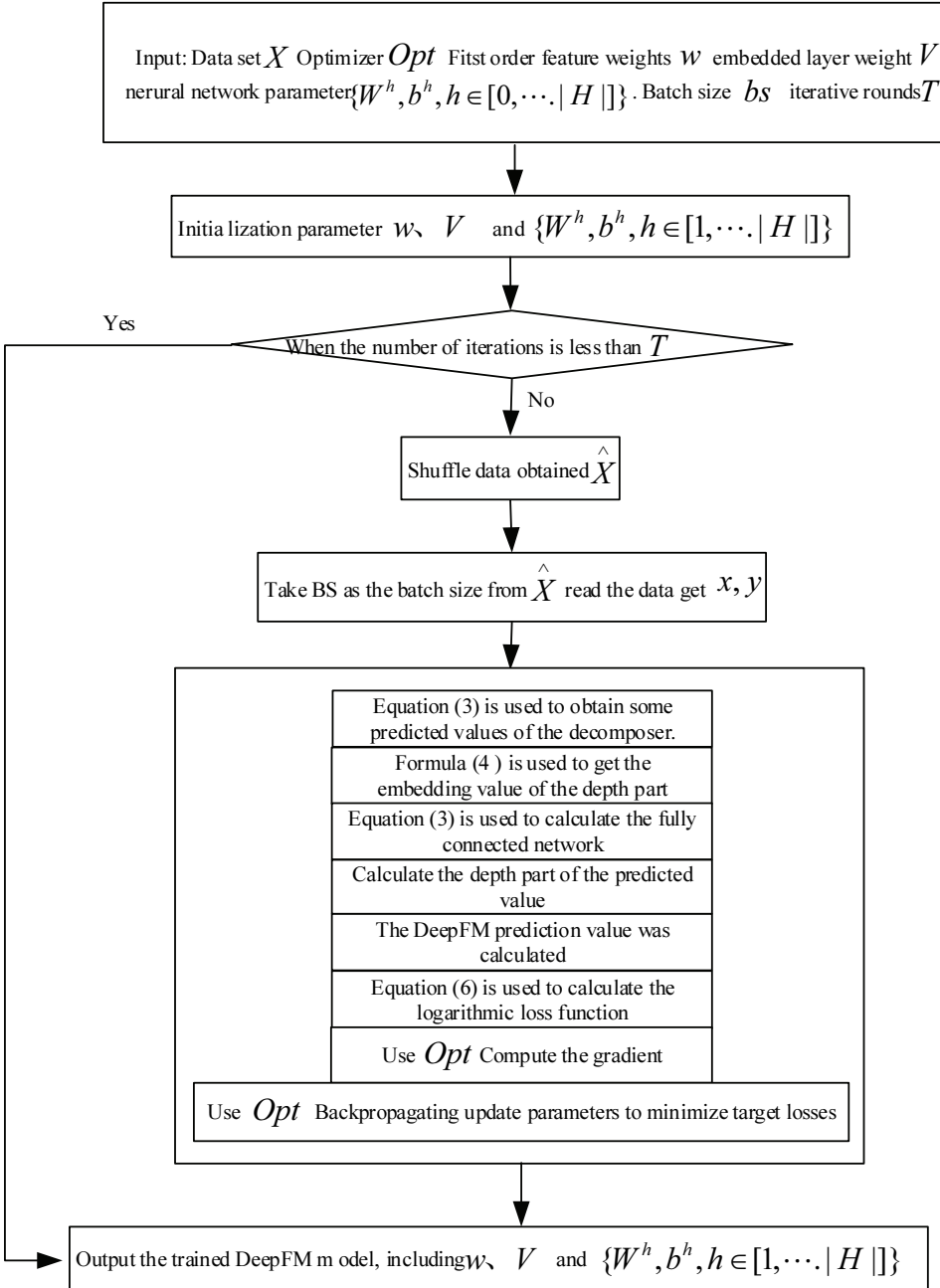

Fig. 3. Calculation flow chart based on DeepFM framework.

GPU is implemented on Win10 operating system of NIVIDIA GeForce GTX 1050, hard disk 1T, the programming language is Python 3.0, and deep learning system is TensorFlow.

The comparison method in this study is as follows:

  (i) CVAE: Proposed by [18].

  (ii) CD-MDTF: Proposed by [23].

    (iii) DeepFM+CVAE: Apply the proposed DeepFM model to the method proposed in [18].

    (iv) DeepFM+CD-MDTF: Apply the proposed DeepFM model to the method proposed in [23].

5.1 Experimental Dataset

The Criteo dataset includes about 45,000,000 user history records. The dataset has 13 continuous feature domains and 26 discrete feature domains. Sequence partitioning is used to segment the dataset, which makes the distribution of test set and training set more similar. A total of 90% are training sets and the remaining 10% are test sets.

The KDD2012 Cup dataset is the advertisement click log data provided by Tencent's search engine Soso (merged into Sogou in September 2013). The training dataset has 149,639,105 records, 987 GB in size. The test dataset is consistent with the training set except for the number of clicks and displays, with a total of 20,257,594 records, 1.26 GB in size. A record in the dataset represents all the information contained in an advertisement among those displayed by a user's retrieval behavior, which is also called an instance.

5.2 Evaluation Indicators

The experiment adopts two evaluation indexes, namely area under the curve (AUC) and Logloss function (logarithmic loss function). AUC and Logloss are two commonly used evaluation indicators in recommendation tasks. Among them, AUC means the probability that a positive sample and a negative sample are randomly selected, and a classifier ranks the positive sample in front of the negative sample. The calculation formula is:

$$AUC = \frac{\sum_{\text{ins}_i \in positive-class} rank_{ins_i} - \frac{M \times (M+1)}{2}}{M \times N} \tag{7}$$

where M is the positive sample number, N is the negative sample number, $rank_i$ is the sample i in a given model fruit in ascending order. Logloss measures the distance between the real and the predicted distribution. It is also the objective function of the click rate prediction model, as shown in Eq. (6).

5.3 Parameter Performance Analysis

To verify the values of the implicit vector dimension and the number of embedded layers of the proposed method, experiments are carried out on Criteo and KDD2012 Cup datasets. In the experiments, the values of the implicit vector dimension and the number of neurons range from 1 to 11. The experimental results are shown in Figs. 4 and 5.

As can be seen from Figs. 4 and 5, when the implicit vector dimension and the number of embedded layers are 6 and 8, respectively, the recognition rates on Criteo and KDD2012 Cup datasets are close to the highest values, and AUC and Logloss tend to be stable thereafter. Therefore, in the following experiments, the implicit vector dimension is set to 6 and the number of embedded layers is set to 8.
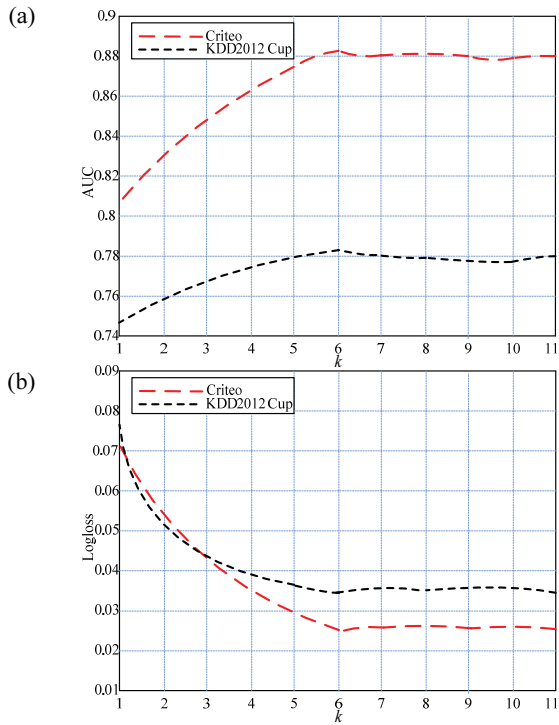
(a)

(b)

Fig. 4. Implicit vector dimension value analysis: (a) AUC and (b) Logloss.
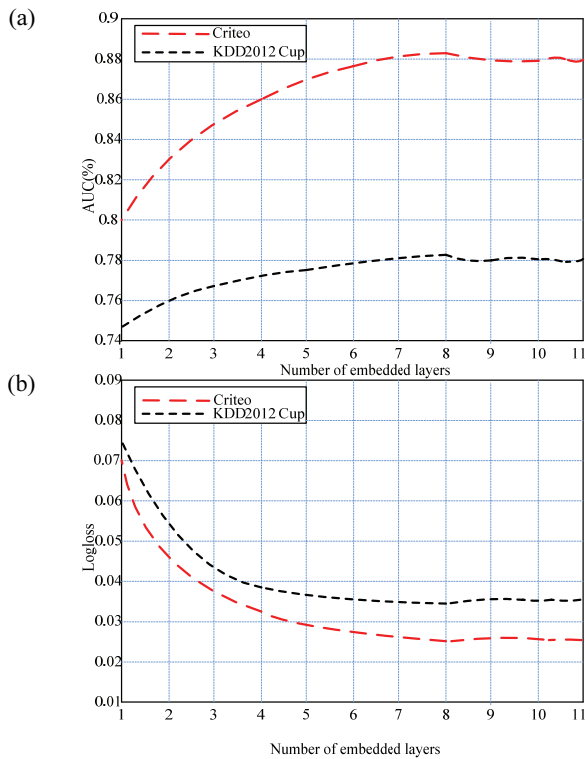
(a)

(b)

Fig. 5. Analysis of number of embedded layers: (a) AUC and (b) Logloss.

5.4 Comparison and Analysis of Recommended Results

AUC and Logloss results obtained by several recommended models on Criteo and KDD2012 Cup datasets are shown in Figs. 6 and 7.

As can be seen from Figs. 6 and 7, compared with other methods, the proposed method obtains a better recommendation effect, and the AUC and Logloss on Criteo and KDD2012 Cup datasets are (0.8834, 0.0253) and (0.7836, 0.0348), respectively, which are better than the comparison algorithm. This is because the proposed method uses K-means clustering to make the similarity of objects in the same cluster as high as possible, to better mine and utilise the correlation between advertisements and queries. Furthermore, DeepFM reduces the noise point and sparsity of the dataset and improves the recommendation effect of the model. CVAE and CD-MDTF have achieved poor results on Criteo and KDD2012 Cup datasets because the two datasets are divided by time, and the test set is likely to have feature combination relations that the training set does not have. In addition, the feature vector dimensions of the two models are too high, and the parameters are not easy to train. DeepFM+CVAE and DeepFM+CD-MDTF have better model effects than CVAE and CD-MDTF because DeepFM compresses the input vectors into low-dimensional dense vectors, reducing the noise points and sparsity of datasets.
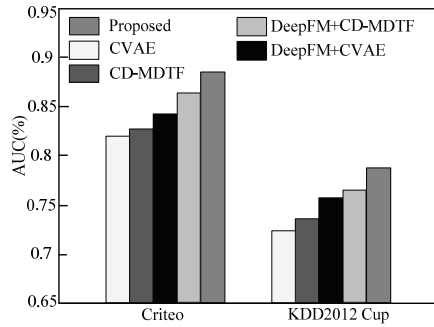

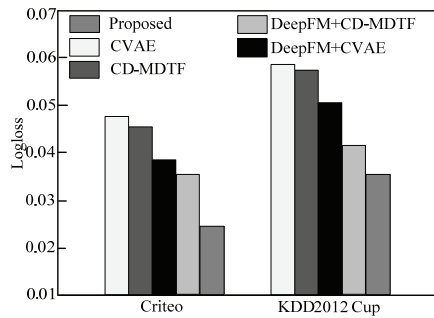**Fig. 6.** Comparison of AUC recommended by different models.

Fig. 7. Comparison of Logloss recommended by different models.

5.5 Comparative Analysis of Running Time

In practical application scenarios, besides the recommendation accuracy, the running time of the model is also crucial. This study records the running time of the five methods on two datasets. Fig. 8 shows the

running time of the model on the scale of 200,000, 400,000, and 600,000 data, respectively.

As can be seen from Fig. 8, there is still a significant difference in the running time of the model among the five methods. The running time of CVAE and CD-MDTF models is only in the pre-estimation model training stage. The feature extraction process relies on domain knowledge and the number of features is relatively small, so the running time of the models is relatively short. However, DeepFM+CVAE and DeepFM+CD-MDTF involve complex operations of tensor decomposition, and the computation cost is also very high when input vectors are compressed into low-dimensional dense vectors through DeepFM. In addition, they involve a large number of features, so the models of DeepFM+CVAE and DeepFM+CD-MDTF run for a long time. The model running time of the proposed method is shorter than DeepFM+CVAE and DeepFM+CD-MDTF. This is because the proposed method clusters advertisements and queries before compressing input vectors into low-dimensional dense vectors through DeepFM. The two clusters are independent of each other, and the clustering order does not affect subsequent calculations, thus achieving the purpose of feature dimensionality reduction. In addition, the most time-consuming advertisement click rate estimation model is based on massive data training, and its calculation process is carried out in an offline environment, thus reducing the running time of the model.
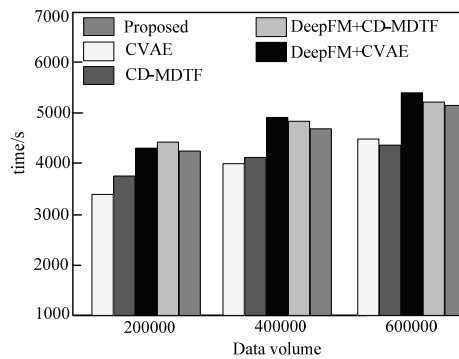

Fig. 8. Comparison of running times of different models.

6. Conclusion

Due to the difference in interaction modes, the recommendation system adopts different recommendation strategies for different types of users. Furthermore, the data of different types of users in the system are also different. Therefore, this study proposes a personalized product recommendation method using DeepFM to analyse user behavior, which is mainly divided into a model training phase and an online prediction phase. In the training phase, the hidden vector of DeepFM is shared with the embedded layer of the deep neural network so that the input vector is compressed into low-dimensional dense vectors, and the purpose of reducing noise and sparsity of the dataset is achieved. In addition, before feature extraction, K-means clustering is carried out on the same object so that high-dimensional data can obtain low-dimensional representation, which can reflect the intrinsic structural characteristics of the original data. In the testing phase, the trained click rate prediction model is used to recommend commodities to customers according to the click rate.

With the further development of mobile Internet, how to recommend personalized products based on

users' behavior has important research significance and application value. Although this study has achieved good results in examining click rate estimation from the perspective of high-dimensional sparse data, there are still some deficiencies. There are many elements for recommendation system modelling, including not only the interaction data between users and projects but also the spatio-temporal sequence patterns of user behaviors, the influence of social relationships, the dynamic evolution of user preferences, and the dynamic changes of project characteristics. Modelling more elements can improve the performance of the recommendation system. Therefore, the future research direction is to study a new deep learning framework that can express and integrate various elements.

Acknowledgement

References

[1] M. Li and L. Hou, "Welfare effects of network neutrality in mobile Internet market," *Enterprise Information Systems*, vol. 14, no. 3, pp. 352-367, 2020.

[2] D. Hidalgo-Mazzei, V. L. Nikolova, S. Kitchen, and A. H. Young, "Internet-connected devices ownership, use and interests in bipolar disorder: from desktop to mobile mental health," *Digital Psychiatry*, vol. 2, no. 1, pp. 1-7, 2019.

[3] I. Baako, S. Umar, and P. Gidisu, "Privacy and security concerns in electronic commerce websites in Ghana: a survey study," *International Journal of Computer Network and Information Security*, vol. 11, no. 10, pp. 19-25, 2019.

[4] C. Burgos, J. C. Cortes, I. C. Lombana, D. Martinez-Rodriguez, and R. J. Villanueva, "Modeling the dynamics of the frequent users of electronic commerce in Spain using optimization techniques for inverse problems with uncertainty," *Journal of Optimization Theory and Applications*, vol. 182, no. 2, pp. 785-796, 2019.

[5] Y. Huang, N. N. Wang, H. Zhang, and J. Wang, "A novel product recommendation model consolidating price, trust and online reviews," *Kybernetes*, vol. 48, no. 6, pp. 1355-1372, 2019.

[6] Y. H. Li, Z. P. Fan, and G. H. Qiao, "Product recommendation incorporating the consideration of product performance and customer service factors," *Kybernetes*, vol. 46, no. 10, pp. 1753-1776, 2017.

[7] Y. Cui, L. Zhang, Q. Wang, P. Chen, and C. Xie, "Heterogeneous network linkage-weight based link prediction in bipartite graph for personalized recommendation," *Procedia Computer Science*, vol. 91, pp. 953-958, 2016.

[8] M. Mpinganjira and D. K. Maduku, "Ethics of mobile behavioral advertising: antecedents and outcomes of perceived ethical value of advertised brands," *Journal of Business Research*, vol. 95, pp. 464-478, 2019.

[9] M. S. Islam and S. A. Eva, "Electronic commerce toward digital Bangladesh: business expansion model based on value chain in the network economy," *Studies in Business & Economics*, vol. 14, no. 1, pp. 87-98, 2019.

[10] N. Dridi and M. Hadzagic, "Akaike and Bayesian information criteria for hidden Markov models," *IEEE Signal Processing Letters*, vol. 26, no. 2, pp. 302-306, 2019.

[11] Z. Wang, M. Wan, X. Cui, L. Liu, Z. Liu, W. Xu, and L. He, "Personalized recommendation algorithm based on product reviews," *Journal of Electronic Commerce in Organizations*, vol. 16, no. 3, pp. 22-38, 2018.

[12] J. Wu and Z. Wu, "Improved fuzzy C-means clustering for personalized product recommendation," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 6, no. 3, pp. 393-399, 2013.

[13] Y. Ning, L. Liu, and Z. Xu, "Research on personalized recommendation algorithm based on user model and user-project matrix," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, Harbin, China, 2011, pp. 2400-2402.

[14] Y. Guo, M. Wang, and X. Li, "An interactive personalized recommendation system using the hybrid algorithm model," *Symmetry*, vol. 9, no. 10, article no. 216, 2017. https://doi.org/10.3390/sym9100216

[15] V. S. Dixit, S. Gupta, and P. Jain, "A propound hybrid approach for personalized online product recommendations," *Applied Artificial Intelligence*, vol. 32, no. 9-10, pp. 785-801, 2018.

[16] X. Lai, L. He, and Q. Zhou, "Personalized product service recommendation based on user portrait mathematical model," in *Proceedings of 2018 International Symposium on Communication Engineering & Computer Science (CECS)*, Hohhot, China, 2018, pp. 328-333.

[17] L. Luo, H. Xie, Y. Rao, and F. L. Wang, "Personalized recommendation by matrix co-factorization with tags and time information," *Expert Systems with Applications*, vol. 119, pp. 311-321, 2019.

[18] B. Pang, M. Yang, and C. Wang, "A novel top-n recommendation approach based on conditional variational auto-encoder," *Advances in Knowledge Discovery and Data Mining*. Cham, Switzerland: Springer, 2019, pp. 357-368.

[19] S. T. Cheng, C. L. Chou, and G. J. Horng, "The adaptive ontology-based personalized recommender system," *Wireless Personal Communications*, vol. 72, no. 4, pp. 1801-1826, 2013.

[20] W. Hong, L. Li, and T. Li, "Product recommendation with temporal dynamics," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12398-12406, 2012.

[21] L. Zhang, J. Li, Q. Zhang, F. Meng, and W. Teng, "Domain knowledge-based link prediction in customer-product bipartite graph for product recommendation," *International Journal of Information Technology & Decision Making*, vol. 18, no. 1, pp. 311-338, 2019.

[22] M. Hao and B. Q. Zhang, "Study on recommendation method based on product evaluation concept tree and collaborative filtering algorithm," *Applied Mechanics and Materials*, vol. 519, pp. 401-404, 2014.

[23] A. Taneja and A. Arora, "Cross domain recommendation using multidimensional tensor factorization," *Expert Systems with Applications*, vol. 92, pp. 304-316, 2018

[24] L. Yu, L. Liu, and X. Li, "A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce," *Expert Systems with Applications*, vol. 28, no. 1, pp. 67-77, 2005.

[25] J. Gao, "Research on goods recommendation strategy based on decision tree," *Applied Mechanics and Materials*, vol. 687, pp. 2718-2721, 2014.

[26] C. B. Jiang, I. H. Liu, Y. N. Chung, and J. S. Li, "Novel intrusion prediction mechanism based on honeypot log similarity," *International Journal of Network Management*, vol. 26, no. 3, pp. 156-175, 2016.

[27] J. P. McCrae and P. Buitelaar, "Linking datasets using semantic textual similarity," *Cybernetics and Information Technologies*, vol. 18, no. 1, pp. 109-123, 2018.

[28] X. Kang and X. Deng, "An improved modified Cholesky decomposition approach for precision matrix estimation," *Journal of Statistical Computation and Simulation*, vol. 90, no. 3, pp. 443-464, 2020.

[29] M. Yavari and A. Nazemi, "Fractional infinite-horizon optimal control problems with a feed forward neural network scheme," *Network: Computation in Neural Systems*, vol. 30, no. 1-4, pp. 125-147, 2019.

[30] S. Zhang, C. Zhao, and F. Gao, "Incipient fault detection for multiphase batch processes with limited batches," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 103-117, 2017.

Jianqiang Xu https://orcid.org/0000-0003-0449-4506

He has got Master of Business and Administration, Graduated from East China University of Science and Technology in 2009. Now he is studying for PhD of control science and engineering at East China University of Science and Technology. His research interests include Internet of Things, artificial intelligence and cloud computing.

Zhujiao Hu https://orcid.org/0000-0002-5600-1833

She has got Master of Science in Engineering, Graduated from Wuhan University in 2005. She is studying for PhD of microelectronics and solid-state science at Fudan University. Also she worked in Shanghai Fine Electronics Co. Ltd. Her research interests include Internet of Things, artificial intelligence, and big data.

Junzhong Zou https://orcid.org/0000-0001-8119-8503

He has got received B.S. degree in Electrical Engineering from Chongqing University, Sichuan, China, in 1982, and received M.S. degree and Ph. D. degree in Advanced Systems Control Engineering, Saga University, Japan, in 1995 and 1998, respectively. He joined East China University of Science and Technology, Shanghai, China, in 2001, and he is currently a Professor and Thesis Adviser of M.S. and Ph. D graduate Students in ECUST. He is also a Director of Texas Instrument Digital Signal Processing Associated Lab in ECUST. His research interests are dynamical control and automation, biomedical signal processing, robotics and mechatronic Servo systems.