Pair-Wise Convolution Network with Transformers for Sequential Recommendation

Jiangpeng Shi[1], Xiaochun Cheng[2], and Jianfeng Wang[3(✉)]

[1] School of Life Sciences, Shanxi Datong University, Datong, China
jiangpeng_shi@163.com
[2] Department of Computer Science, Middlesex University, London, UK
xiaochun.cheng@gmail.com
[3] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
wjf739@gmail.com

Abstract. Sequential recommendations seek to employ the sequence of interactions between users and commodities to predict their next behavior based on the behavior they have recently made. Previously, some recommendation systems have been built on Markov chains and recurrent neural networks (among others). However, these methods have many limitations that they emphasize too much sequence change to fully emphasize the correlation between adjacent items; Besides, they generally ignore the influence of contextual information. To solve the shortcomings of the existing sequential recommendations, we try to model the relationship between items, get an effective representation of sequential features, and capture complex sequence correlations. Specifically, we propose a pair-wise convolution network with transformers for the sequential recommendation. The two-dimensional convolution networks encodes the sequence into a three-dimensional tensor and learns the relationships of features between the sequences. We adopt a residual connection to prevent the gradient from disappearing and solve the loss of feature information. The experimental results show that our method is superior to various advanced sequential models on sparse and dense data sets and different evaluation indicators.

Keywords: Sequential recommendation · Convolutional attention network · Pair-wise convolution

1 Introduction

In the present information explosion era, there are more and more data on the Internet, which greatly enrich the content of the Internet. Due to the extensive use of internet technologies, social media platforms, and e-commerce systems (such as Tiko, Amazon, or Netflix) are used more and more frequently in people's lives. People's habit of obtaining information from nothing, from

Y. Xiang et al. (Eds.): SocialSec 2020, CCIS 1298, pp. 433–445, 2020.
https://doi.org/10.1007/978-981-15-9031-3_38

quantitative change to qualitative change, has undergone tremendous changes. At the same time, Internet users not only put forward higher requirements for the speed and real-time information acquisition but also have an increasing demand for personalized and precise search. Therefore, a large number of clicks, purchase interactions, and other user feedback are also generated in these systems. Taking Tiko as an example, users may watch thousands of short videos every week. Faced with massive amounts of information, they have become dazzled and unable to make decisions. It is difficult for people to find the information that is most suitable for them from a large amount of information. The recommendation system has been widely applied in online service to improve the quality of search engines and the accuracy of their result lists. The main purpose of the recommendation system is to infer users' preferences for items based on user interaction behavior and assist users to filter out their desired products. Therefore, the recommendation plays an important role in thereby improving user search efficiency and improving information overload, such that this helps the decision-making process of selecting the best product design for manufacture.

In the past decade, many efforts have been made to develop general recommendation methods, such as model-based methods, deep learning-based methods include Markov models and recurrent neural network models. For example, Liu et al. [7] proposed a new short-term attention/memory priority model, which can obtain the user's general interest from the long-term memory of the conversational context, while considering The user's current interest comes from the short-term memory of the last click. However, this model ignores the dynamic changes in users' long-term preferences over time and often uses static features to model users' long-term preferences. They integrate user-item or item-item interactions in a linear fashion, thereby limiting the capabilities of the model. Ying et al. [13] proposes a two-layer attention network that combines long-term and short-term user interests for comprehensive modeling. Yuan et al. [16] proposed a CNN model to learn item sequence features, add a residual mechanism to the deep CNN, and stitch with user embedding in the last layer, and finally input MLP for prediction. However, the user's interest pattern is personalized. Most of the existing sequence recommendation models usually only consider the user's recent interaction behavior independently, without obtaining the real internal connection between the user's continuous behavior, and most algorithms can only model the user's historical behavior linearly, and it is difficult to have a flexible sequence. Therefore, how to obtain the potential dependencies between users' historical behaviors and how to accurately obtain user interests have become key research issues in personalized recommendation systems.

To solve the above limitations, we propose a pair-wise convolution network with transformers for a sequential recommendation (Fig. 1), which considers the relative relationship between any two items. Specifically, the Transformer-based semantic layer models the user's historical sequential behavior and decodes the target product to obtain the user's preferences; the fusion layer captures the linear and non-linear correlation between the user's preferences and the target

product. To understand user preferences more dynamically, the model introduces users' long-term preferences and current users' short-term interests. This paper believes that each user has different preferences for each commodity. The attention mechanism in transformer automatically assigns weights to items to dynamically capture long-term and short-term interests. At the same time, the layer is designed to adaptively combine long-term and short-term preferences, which pays attention to the different effects of different users on each product. The experimental results show that the performance of this method exceeds that of all baseline models.

A person has a mature and stable value system to see and understand the world, and he has his preferences and judgments about what he comes into contact with, and the behavioral decisions that people usually make are often based on long-term preferences. However, in the real world, many factors affect user decisions. Long-term preferences and short-term preferences together determine people's decision-making behavior. Therefore, the recommendation is based on the long-term preference and the short-term preference as the contextual information. The contributions of this article are summarized as follows: 1. Considering the context information in the sequence table, we make full use of the attention mechanism to make similar items more weighted. The weighted results make them closer and closer in space, modeling users' short-term hobbies, and making a more accurate user experience. Products Recommended. 2. This paper adopts CNN-based network results to capture the relationship between items through the window sliding of the convolution kernel, models the commodity sequence that best reflects what customers need, and analyzes the user's search intention understanding. 3. Considering the time sequence of the user sequence, we use the transformer to model the user's click sequence and make recommendations.

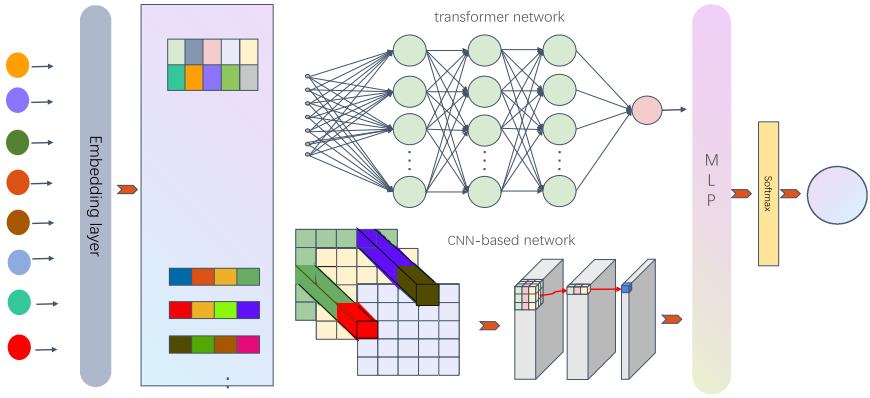

Fig. 1. The framework of the proposed method

The remainder of the paper is organized as follows. In Sect. 2, gives an overview of the literature and a critical review of research in the area of this work. In Sect. 3, we present the methods for predicting potential needs and wants, the different aspects and improvements are presented as sections in this chapter, which are value chain for predicting potential customer needs and wants. In Sect. 4, the data analysis and results for each dataset are shown, and the obtained results are summarized. In Sect. 5, the conclusion and future work are discussed.

2 Related Work

2.1 Sequential Recommendation

More recent attention has focused on the provision of sequential recommendations. The most common one is based on Markov's recommendation, which can effectively capture the local sequence model. Rendle et al. [9] proposed a Factorized Personalized Markov Chains based Markov chain to predict the next item. However, this approach performs best on sparse data sets. He et al. [2] proposed high-order Markov Chains to model pairwise user-item and item-item interactions. Tang et al. [10] models the sequence as a two-dimensional space and adopts 2D CNN to convolute the embedding items to predict the next item. However, the higher-order MC based methods need to be specified rather than being chosen adaptively. In many research methods, many of the methods are based on RNN to predict the next items. Yu et al. [14] proposed a dynamic recurrent basket model based on RNN for the next-basket recommendation. HidasiK et al. [4] proposed to use Gated Recurrent Unit (GRU) to model sequential behavior for the session-based recommendation. Since these RNN-based methods take the state from the last step and current action as their input, these dependencies make RNN-based methods less efficient (i.e., higher model complexity). It may be because these complex models require large amounts of data to capture long-term patterns, i.e., easily overfitting in high-sparsity settings.

2.2 Deep Learning Models

Many studies focus on predicting the next item as well as improve services by providing reliable insight into what customers need and want. To date, several studies have investigated the search recommendation algorithm. These studies have made important progress in modeling user behavior sequences. The main research work is based on the convolutional neural network and attention mechanism, such as Caser [10] and NextItNet [15]. Caser [10] as the most typical CNN, however, the standard CNN architecture and max-pooling operation of Caser were not well-suited to model long-range user sequences. NextItNet [15] proposed to model the user interaction-item sequence by stacking CNN layers to increase the receptive field of higher layer neurons, The results show that the NextItNet is more effective than the RNN model to recommend the top-n

items. Based on this, many methods are on the extension. At the same time, the good results are shown by the self-attention model based on the transform in the field of SRS. DIN considers the influence of different items in the behavior sequence on the current predicted item by introducing an attention mechanism; DIEN solves the shortcoming that DIN cannot capture the dynamic changes in user interests. DIN [18] and DIEN [17] do not consider the session information in the user's historical behavior, because the behavior in each session is similar, and the difference between different sessions is very large. The BST [1] model uses the Transformer model to capture the associated features of each item in the user's historical behavior sequence. However, the computational cost of the self-attention mechanism is higher than the CNN structure of the superimposed expansion.

3 Method

This paper proposes the pair-wise convolution network with the transformers model, which combines long-term preferences and short-term preferences to provide users with recommendations. This article combines long-term and short-term interest to extract relevant information from users' recent historical interaction behaviors (click, browse, purchase). The system can automatically collects the required data in the implicit feedback scenario, which not only effectively alleviates the problem of data dispersion, but also provides users with a smoother and more comfortable experience.

3.1 Behavior Sequence Layer

At present, deep learning algorithms have made great progress in the recommendation system, and all have good effects such as FM [8], DeepFM [6], Wide & Deep [3,11]. The general method is converted discrete high-dimensional features to fixed-length continuous features through feature embedding operations, and then perform feature extraction through multiple connections, and finally activate function to obtain the predicted recommendation probability. The user's interest characteristics have obvious diversity and local activation. Due to the related characteristics of user interests, if users have different tendencies for different commodities. Therefore, the attention mechanism is introduced to model interest. The most basic interest extraction method is the same as the traditional deep model method, which mainly includes a feature embedding layer, a multilayer perceptron, and an pooling layer and connection layer.

Embedding, as a commonly used vector representation of deep learning, can extract out the multi-dimensional latent features of commodities. The user (ID and its attributes), items (ID and its attributes), and contextual information are uniformly expressed as feature vectors as input to predict the target score value. The data is represented as a feature vector to the target value, each feature vector is represented as a hidden vector, and the interaction between all non-zero features is considered in the hidden space. The neural network can be used

directly. We use word2vec to make a queue with historical data of behavior to form label expansion to get similar items.

In this sequence, we elaborate on the sequence recommendation problem as follows. We mark the set of users as U and the set of products at I. For each user, there will be a sequence of items S corresponding to the user. We embed the user and the product into two matrices, respectively, where d is the latent dimension representation, then the embedding matrix is expressed as follows:

$$E^{(i,t)} = \begin{bmatrix} e_{S^i_{t-L}} \\ \vdots \\ e_{S^i_{t-2}} \\ e_{S^i_{t-1}} \end{bmatrix} \tag{1}$$

Where $e_{S^i_{t-L}}$ is the embedding vector. The items and users were embed into two matrices $E_I \in \mathbb{R}^{|I| \times d}$ and $E_U \in \mathbb{R}^{|U| \times d}$, where d is the latent dimensionality. For user u, we retrieve the input embedding matrix by looking up the previous L items in the item embedding matrix. The role aims to transform the original ID behavior sequence into an embedding behavior sequence. Inspired by transformer [1] and the 2D-CNN network [12], we convolve the three-dimensional pair-wise sequence to extract features. The convolution process is as follows,

$$c^k_i = \phi_c \left(E_{i:i+h-1} \odot F^k \right) \tag{2}$$

where F^k is the convolution kernel. The residual network was used to fuse the k features and the final features are expressed as follows,

$$c^k = \left[c^k_1 c^k_2 \cdots c^k_{L-h+1} \right] \tag{3}$$

We concatenate the outputs of the convolutional layers and feed them into a fully-connected neural network layer to get more high-level and the features $\tilde{c}^k$:

$$\tilde{c}^k = \sum_{l=1}^{L} \tilde{F}^k_l \cdot c^k \tag{4}$$

3.2 Interest Extraction Layer

The interest extraction layer is used to extract user interest by simulating the process of user interest migration. It contains a semantic layer and a feature extraction layer, the semantic layer is the main structure of the model, which mainly includes the self-attention based on the multi-head attention mechanism. First, the network learns the dependence of each item in the behavior sequence through self-attention to obtain user semantic characteristics. Then, the feature of the target is decoded to the user's semantic feature to obtain the user's

semantic preference by the attention. Next, the semantic layer will be described in detail, such as the following formula.

$$\tilde{m} = \text{self-attention}\left(m_{in}\right) \tag{5}$$

where $\tilde{m}$ represents user semantic features, self-attention represents self-attention mechanism, and m_{in} represents user's input features. The attention decodes target item features and user semantic features to obtain user semantic preferences m_o, as follows:

$$m_o = \text{Attention}\left(I_t, \tilde{m}, \tilde{m},\right) \tag{6}$$

$$\tilde{m} = \text{Attention}\left(m_{in}, m_{in}, m_{in}\right) \tag{7}$$

Among them, $\tilde{m}$ represents the user's semantic preference. I_t represents the features of the target item, The attention is formula as:

$$U = \mathit{Attention}\left(m_t, \tilde{m}, \tilde{m}\right) = \text{Softmax}\left(\frac{I_t \tilde{m}^T}{\sqrt{D}}\right)\tilde{m} \tag{8}$$

But there is a multi-head parallel mechanism in the transformer. We connect the attention of each head into multi-head attention, and then the multi-head attention is the following:

$$U = \text{Concat}\left(\mathit{head}_1, \ldots, \mathit{head}_i \ldots, \mathit{head}_{\text{n}}\right) \tag{9}$$

In order to further increase the nonlinearity of the model, we adopt a feedforward neural network. It is defined as follows:

$$\begin{aligned} m_o &= FFN\left(U_o\right) \\ &= \text{Normalize}\left(\text{Conv1D}(\text{ConvlD}\left(m_o\right)) + m_o\right) \end{aligned} \tag{10}$$

Normalize means normalization to solve the problem of vanishing gradient, Conv1D means a one-dimensional volume network. The two-layer convolutional neural network performs two nonlinear mappings of m_o. At the same time, to prevent the loss of original information, a residual connection method is adopted.

3.3 Interest Fusion Layer

In order to provide better generalization capabilities for the entire model, this article adds a fusion layer to the semantic layer. The fusion layer learns the correlation between the target product features generated from the embedding layer and the user's semantic preferences obtained from the semantic layer and merges the two models through the output of the last hidden layer of the multi-layer perceptron. Therefore, at the fusion layer, this article uses a simple linear function to capture the interaction between user preferences and item features:

$$\begin{aligned} z_0 &= \text{Concat}\left(m_o, I_t\right) \\ z_1 &= \mathit{ReLU}\left(W_1^T z_0 + b_1\right) \\ &\cdots \\ h_2 &= z_l = \mathit{ReLU}\left(W_l^T z_{l-1} + b_l\right) \end{aligned} \tag{11}$$

Among them, W_l and b_l are respectively the weight matrix and offset vector of the hidden layer of the i-th layer. Combine the feature vector of the product and the output of the multilayer perceptron as the input of the output layer:

$$h = Concat\,(h_1, h_2)\,\hat{y} = W' \begin{bmatrix} h \\ m_o \end{bmatrix} + b' \tag{12}$$

The main function is to stimulate the interest evolution process related to the current target advertisement by adding an attention mechanism based on the interest extraction layer. Recently developed methods focus on designing structures of MLP for better information extraction. The objective function used in the base model is the cross-entropy loss function defined as:

$$L = -\frac{1}{N} \sum_{(x,y)\in S} (y \log \hat{y} + (1-y) \log(1-\hat{y})) \tag{13}$$

where S is the training set of size N, $y \in 0, 1$ as the label, $\hat{y}$ is the output of the network after the softmax layer, representing the predicted probability of sample being clicked.

4 Experiment

4.1 Datasets and Experiment Data

To verify the effective performance of our method, we adopted two basic data sets for the experimental method. **Gowalla**: this website is a social network website with time frame sequence. This data set contains implicit feedback through user-venue check-ins. **MovieLens**: this data set is a collaborative filtering algorithm widely used in recommendation systems. We use the ml1m data set,which consist of one million user ratings to verify the performance of our algorithm.

The method of processing data is similar to the previous method [12]. For all data sets, we regard user comments and ratings as implicit feedback of users, and divide behavior sequences according to time frames. During the processing, the products and users were discarded that fewer than five related behavioral. We divide the data set into training set, validation set and test set. Among them, in the behavior sequence, the most recent action is used as the test set, the second most recent action is used as the verification set, and the rest is used as the training set. The processed data set is shown in Table 1.

Table 1. Statistics of the datasets.

Dataset	#users	#items	avg. #act. per user	avg. #act. per item	#actions
ML-1M	6.0K	3.4K	165.50	292.06	0.993M
Gowalla	13.1K	14.0K	40.74	38.12	0.533M

4.2 Comparison Methods

We have selected the following mainstream sequence recommendation algorithms as our comparison algorithm.

Factorized Markov Chains (FMC) [9]: Based on the Markov chain algorithm, we use two product embeddings to decompose the product conversion matrix, and generate a recommendation sequence based on the last viewed product,

Factorized Personalized Markov Chains (FPMC) [9]: It is an extension based on the Markov model. FPMC sets up a Markov chain for each shopping sequence. The established Markov chain not only learn the long-term preferences of users but also capture the demands of user that provide personalized recommendations for each user. Therefore, this model has a strong advantage in modeling sequences.

GRU4REC [5]: The method used recurrent neural networks for conversation recommendation tasks, using recurrent neural networks to model conversation sequences. The method treats each user's feedback sequence as a session.

Convolutional Sequence Embeddings (Caser) [10]: The first application of convolutional neural network in sequence model, convolution from the vertical and horizontal dimensions respectively to capture high-order Markov chains, this method has achieved good results.

Convolutional Neural Networks (CosRec) [12]: The sequence between items is encoded into a three-dimensional tensor in a pair-wise manner, and a 2D convolutional neural network is used to learn local features.

4.3 Implementation Details

The data preprocessing and training in this article run on the Ubuntu operating system, the graphics card is NVIDIA GTX 1080, the memory is 32 GB, and the integrated development environment is PyCharm. The data preprocessing mainly is Python 3.7, and the related extension libraries are Numpy, Pickle scipy, etc. to support large-scale file data to reading and simple matrix operations. At the same time, in order to further improve training efficiency, the open-source deep learning framework is pytorch1.1 during training, and CUDA10.0 was introduced to the GPU acceleration.

We used 10 convolutional blocks. For the ml1m dataset, the dimension of d is set to 20, the length of l is 10, and T is set to 2. For the gowalla dataset, the dimension of d is set to 100, and the length of l is 10, T is set to 3. We use two self-attention blocks. The optimizer is the adam optimizer, the learning rate is set to 0.000005, and the batch size is 512. The dropout rate of turning off neurons is 0.2 for MovieLens-1m and 0.5 for the other datasets due to their sparsity.

4.4 Evaluation Metrics

The precision rate indicates how many positive samples are predicted correctly among the samples whose predictions are positive, and the recall rate indicates

how many positive samples are correctly predicted among all the positive samples. We evaluate our model with precision@n, recall@n and mean average precision, as shown in the Formula 14 and Formula 15.

$$Prec@N = \frac{\left|R \cap \hat{R}_{1:N}\right|}{N} \tag{14}$$

$$Recall@N = \frac{\left|R \cap \hat{R}_{1:N}\right|}{|R|} \tag{15}$$

4.5 Recommendation Performance

Table 2 shows the experimental results of five comparison methods and our algorithm on two data sets. It can be concluded from the table that our algorithm obtains the best performance on all data sets, which shows the effectiveness of the proposed algorithm. In addition, from observing the comparison method, it is found that FMC and FPMC perform poorly on some data sets. Both FMC and FPMC deal with fully parameterized transition graphs, and their premise is that users and commodities have independent parameters, and the calculation of each parameter does not consider the influence of other parameters. However, in the personalized recommendation process, we need to decompose the transferred three-dimensional matrix to break the independence between parameters and estimation, so that the mutual influence between similar users, products, and transfer situations can be considered. At the same time, it is observed that

Table 2. Performance comparison on the four data sets.

Dataset	Metric	FMC	FPMC	GRU4Rec	Caser	CosRec	Ours
$ML-1M$	MAP	0.0687	0.1053	0.1440	0.1507	0.1883	**0.1895**
	Prec@1	0.1280	0.2022	0.2515	0.2502	0.3308	**0.3308**
	Prec@5	0.1113	0.1659	0.2146	0.2175	0.2831	**0.2818**
	Prec@10	0.1011	0.1460	0.1916	0.1991	0.2493	**0.2506**
	Recall@1	0.0050	0.0118	0.0153	0.0148	0.0202	**0.0204**
	Recall@5	0.0213	0.0468	0.0629	0.0632	0.0843	**0.0837**
	Recall@10	0.0375	0.0777	0.1093	0.1121	0.1438	**0.1438**
Gowalla	MAP	0.0229	0.0764	0.0580	0.0928	0.0980	**0.1894**
	Prec@1	0.0517	0.1555	0.1050	0.1961	0.2135	**0.3374**
	Prec@5	0.0362	0.0936	0.0721	0.1129	0.1190	**0.2790**
	Prec@10	0.0281	0.0698	0.0782	0.0571	0.0884	**0.2473**
	Recall@1	0.0064	0.0256	0.0155	0.0310	0.0337	**0.0206**
	Recall@5	0.0257	0.0722	0.0529	0.0845	0.0890	**0.0834**
	Recall@10	0.0402	0.1059	0.0826	0.1223	0.1305	**0.1436**

the performance of recommendation algorithms based on convolutional neural networks (Caser and CosRec) is better than traditional recommendation algorithms (FMC and FPMC), which shows that CNN-based can effectively model the interaction between users and items. Compared with Coser and our method, our algorithm has better performance, which shows that the attention mechanism is effective for mining the short-term and long-term intentions of historical users for modeling user and project interaction. Compared with all recommendation methods, the performance of our algorithm is better in solving the dynamic attention mechanism. Our model will assign different weights to different candidate items according to the user's interaction sequence and the user's historical score. Items are often highly related to candidate items. As can be seen from the data in the table, in the user interaction sequence, the attention mechanism can assign different weights to different products, which is higher than other types of product activation weights.

Figure 2 and Fig. 3 show the performance of our method on Gowalla and ml1m datasets, respectively. In each figure, the horizontal axis represents the number of training sessions, and each table shows the performance indicators of six tests. We can see from Fig. 2 that as N increases, precision gradually decreases. In addition, the performance reaches its best around twenty-eight rounds, and the performance will not increase as the number of rounds increases. Recall and precision are just the opposite. With the increase of N, the recall gradually becomes smaller. When the number of rounds reaches fourteen, the recall will not increase and the performance tends to be stable. Figure 3 shows the performance of our method on the ml1m data set. In general, the performance increases with the increase in the number of rounds. Although there are

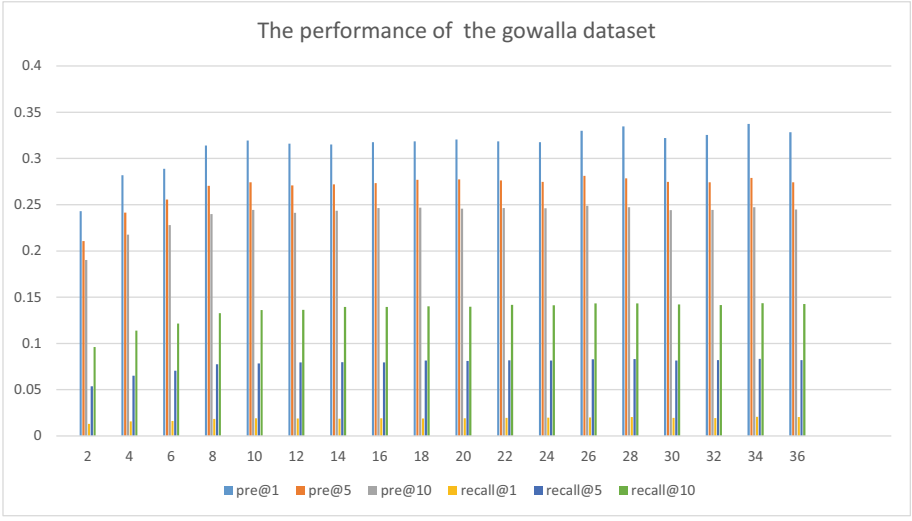

Fig. 2. The performance of gowalla data with various epoch.

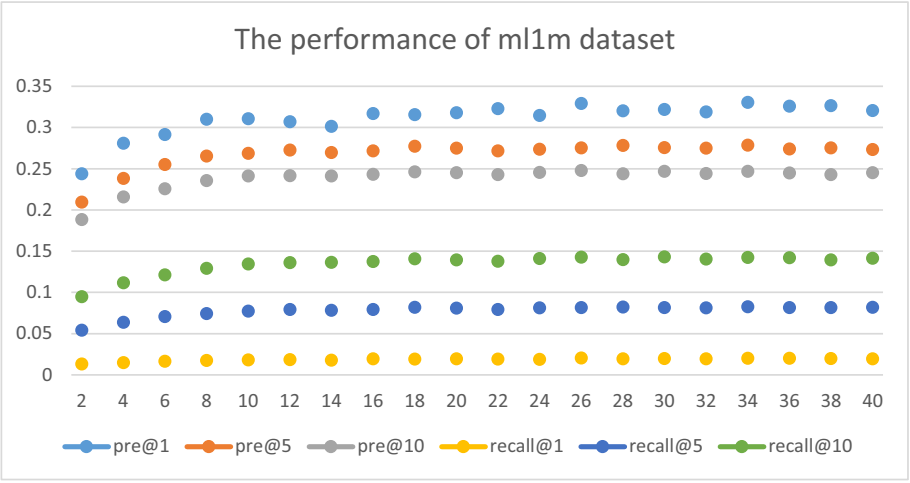

Fig. 3. The performance of gowalla data with various epoch.

fluctuations in a small range, the overall performance tends to be stable. The performance leveled off after the 26-th round. From the performance of the total extraction volume on the two data sets of Fig. 2 and Fig. 3, the precision increases with the increase of N, and the recall decreases with the increase of N.

5 Conclusion

This paper analyzes the personalized recommendation model. In order to make recommendations for users' hobbies more accurately, the current more accurate CNN-based network is selected as the basis for optimization and improvement, and finally, a feature-based weight extraction is proposed. In terms of data analysis and processing, this article adopts a public data set. The data is first analyzed for the defect value, and the original data is sampled and balanced to ensure the accuracy of later training. This paper designs a new feature weight extraction model, which assigns different weights to different important items in the user interaction sequence, and more accurately simulates the user's real purchase situation. In this paper, a corresponding feature weight extraction module is designed to further enhance the accuracy of model recommendation. Based on the original deep interest network, the corresponding advantages of the feature weight extraction model are introduced, and a more accurate recommendation effect is achieved through experimental verification. In the future research process, we will study multi-modal recommendation, take into account the multi-semantic user's intention understanding of the combination of text and pictures, and combine the current research situation to continue to research.

References

1. Chen, Q., Zhao, H., Li, W., Huang, P., Ou, W.: Behavior sequence transformer for e-commerce recommendation in Alibaba. CoRR abs/1905.06874 (2019)
2. He, R., McAuley, J.J.: Fusing similarity models with Markov chains for sparse sequential recommendation. In: ICDM, pp. 191–200 (2016)
3. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural collaborative filtering (2017)
4. Hidasi, B., Karatzoglou, A.: Recurrent neural networks with top-k gains for session-based recommendations. In: CIKM, pp. 843–852 (2018)
5. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. In: ICLR (2016)
6. Liu, F., Guo, W., Guo, H., Tang, R., Ye, Y., He, X.: Dual-attentional factorization-machines based neural network for user response prediction. In: WWW, pp. 26–27 (2020)
7. Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: STAMP: short-term attention/memory priority model for session-based recommendation. In: KDD, pp. 1831–1839 (2018)
8. Mao, X., Mitra, S., Swaminathan, V.: Feature selection for FM-based context-aware recommendation systems. In: ISM, pp. 252–255. IEEE Computer Society (2017)
9. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: WWW, pp. 811–820 (2010)
10. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: WSDM, pp. 565–573 (2018)
11. Xu, J., Shi, J., Yao, Y., Zheng, S., Xu, B., Xu, B.: Hierarchical memory networks for answer selection on unknown words. In: Calzolari, N., Matsumoto, Y., Prasad, R. (eds.) COLING, pp. 2290–2299. ACL (2016)
12. Yan, A., Cheng, S., Kang, W., Wan, M., McAuley, J.J.: CosRec: 2D convolutional neural networks for sequential recommendation. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, 3–7 November 2019, pp. 2173–2176 (2019)
13. Ying, H., et al.: Sequential recommender system based on hierarchical attention networks. In: IJCAI, pp. 3926–3932 (2018)
14. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A dynamic recurrent model for next basket recommendation. In: SIGIR, pp. 729–732 (2016)
15. Yuan, F., et al.: Future data helps training: modeling future contexts for session-based recommendation. In: WWW, pp. 303–313 (2020)
16. Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J.M., He, X.: A simple convolutional generative network for next item recommendation. In: Culpepper, J.S., Moffat, A., Bennett, P.N., Lerman, K. (eds.) WSDM, pp. 582–590. ACM (2019)
17. Zhou, G., et al.: Deep interest evolution network for click-through rate prediction. In: AAAI, pp. 5941–5948 (2019)
18. Zhou, G., et al.: Deep interest network for click-through rate prediction. CoRR abs/1706.06978 (2017)