Volume: 15 Issue: 06 Pages: 6156 – 6162 (2024) ISSN: 0975-0290

Convolution Serialization Recommendation with Time Characteristics and User Preferences

Huiyuan Li [1], Zhengang Li [1] , Tao Jin [1] , Ruofei Wang [1], Haowei Ren [1], Jiahao Lu [1], Jiekun Xu [1], Tengda Hou [1], Ran Jin [1]

[1]College of Big Data and Software Engineering / Zhejiang Wanli University /China

Corresponding Author

ZhengangLi，Tao Jin.

Email: *447808600@qq.com*

Email: *1047830356@qq.com*

ABSTRACT

The recommendation system has been widely used in life, which greatly facilitates people's life. The traditional recommendation method is mainly used to analyze the interaction between users and items. analyze the history of users and items, and get only the users' preferences for items in the past. The serialization recommendation system analyzes the sequence of users interacting with objects in a recent period of time. To consider the relevance of the user's before and after behavior, can obtain the user's preference for items in the short term. However, the serialization method emphasizes the user's connection with the item in the short term. Ignoring the relationship between the properties of objects. In view of the above problems, the convolution serialization recommendation of fusion time characteristics and user preferences is proposed Convolutional Embedding Recommendation with Time and User Preference. CERTU model. The model is able to analyze the diversity relationships between items, thus capturing the user's dynamic preference for items over time. Otherwise, the model further considers the influence of a single item and multiple items present in the item sequence on the next item recommendation. The experimental results show that the No. The CERTU model outperforms the current baseline method.

Keywords - **Recommendation System, Convolutional Neural Network, Serialization Recommendation, User Interest, Time Characteristics.**

Date of Submission: March 07, 2024 Date of Acceptance: April 05, 2024

I. INTRODUCTION

With the development of the times, information technology has made people's lives more and more convenient. Human society has entered the era of big data, and at the same time, human beings have also fallen into the dilemma of data. It's too expensive for people to find valuable information in the data and hard to get the information you need from the massive amount of data. The recommendation system is exactly for: It was born to solve such problems and has developed rapidly. The traditional recommendation system is often analyzed from the user's preferences or the characteristics of the item. For example, based on the content recommendation system analyzes the record of items that the user has interacted with in the past Items are divided into a variety of attributes.

However, users' interests and preferences are not fixed, but will change over time. For a certain type of item, the user may have a great interest in it before owning, and after owning and using for a period of time, the interest may diminish, at this time the recommendation system should not recommend the same item to the user. Top-n serialization recommendation mainly deals with the sequence of items that users interact with in the past period of time, with the purpose of predicting the Top N items that users may like in the near future, but many top-N recommendation algorithms only consider the general preferences of users, and ignore the relationship of items in time. For example, in the field of e-commerce, the user's last or previous purchase behavior will greatly affect the subsequent purchase behavior. After purchasing a smart phone, the user may want to buy a headset. At this time, the recommendation system can recommend related items to the user. After the user has purchased the headset, the demand for the same type of headset is not high, and the recommendation system should not recommend the headset related items to the user. This is a typical scenario for considering how users' interests are affected by changes in time.

At the same time, the user's own preferences are also an important factor affecting the recommendation system Plain. For example, some users are particularly fond of skirts and like to collect various types of skirts. This is a long-term and fixed behavior, and the recommendation system should take these factors into account. It is often not enough to consider the user's own interests and hobbies, and there are also things between them. There are more abundant attributes. For items, items can be roughly generalized from each one. There are two kinds of relationships: complementarity and substitution [1]. For complementary relationships: between two items, roles can complement and support each other, such as mobile phones and headphones. For substitution generational relationship: one item can replace another item if the user is missing. One item can be replaced by another, and this relationship is often the case. Items of the same type, different brands, or different styles, such as different brands of headphones. In addition to the correlation between the attributes of items, items can also be found in the sequence of items relationship. For item sequences, different items may act independently on a predicted target item, or they may be combined with different items to have an effect on a predicted target item. For example, after buying milk, flour can be recommended, after buying butter, you can recommend flour, and consider the two buying behaviors together. The probability of powder is higher. Intuitively, individual behavior and joint behavior are also important factors that affect the performance of recommendation systems.

Based on the above analysis, this paper proposes a Convolutional Serialization Recommendation (CERTU) model that integrates temporal characteristics and user preferences. CERTU models primarily from two dimensions: the user and the object. On the user side, not only the user's long-term preferences are taken into account, but also the user's preferences for the same item over time. In terms of items, the analysis of a single item prediction and the joint prediction of multiple items is carried out, as well as the analysis of the relationship between items.

The main contributions of this paper are as follows:

(1) This paper comprehensively considers the two aspects of users and items for modeling. The model proposed in this paper not only analyzes the impact of time on users, but also divides items into more fine-grained ones, integrating the relationship between items, the prediction of a single item, and the joint prediction of multiple items.

(2) The model proposed in this paper not only considers the short-term preference of users, but also the user's long-term preference is analyzed, and the user's behavioral characteristics are processed in a more fine-grained way.

(3) In this paper, the proposed CERTU model is experimented on the Amazon dataset. The experimental results show that the CERTU model outperforms the current popular methods.

Section 2 of this paper describes the related work; Section 3 introduces the CERTU model proposed in this paper; Section 4 mainly describes the experimental part and analyzes the experimental results; finally, the paper is summarized..

II. RELATED WORK

2.1 Collaborative Filtering Algorithms

User-based Collaborative Filtering (UserCF) typically recommends items liked by other users with similar interests to the user. Item-based Collaborative Filtering (ItemCF) recommends items similar to those the user previously liked. While collaborative filtering algorithms can make it easier for machines to analyze complex information (such as music), they struggle to capture how a user's interest in items changes over time, lacking an analysis of changes in user interests. This paper differentiates the relationships between items as complementary and substitutive, analyzing how user interest in items changes over time under these two relationships.

2.2 Sequential Recommendation

In real life, people's current behavior is always influenced by historical behaviors. Sequential recommendation considers users' historical behaviors, recommending items by analyzing their past behaviors. The FPMC model combines matrix factorization with Markov chains, learning a transition matrix for each user and using a pairwise interaction model to observe matrix transitions, embedding the transition information between adjacent behaviors into the latent next behavior and analyzes users' general preferences, using representation learning to treat serialized information as latent factors [3]. Although these models consider sequential user behavior, they mainly focus on two adjacent behaviors and are applied to dense data, showing limitations in handling more complex multi-step behaviors or sparse data. The Fossil model primarily uses an improved Markov chain to solve the problem of data sparsity in sequential recommendation, capturing users' dynamic interests in sparse data [4].

2.3 Neural Network Methods

The RBM model was the first to successfully apply a two-layer neural network to recommendation problems [5]. LeCun introduced the world's first Convolutional Neural Network (CNN), primarily used for solving the problem of machine recognition of handwritten fonts [6]. Convolutional neural networks are generally used in image processing, and Zheng et al. proposed the Deep Cooperative Neural Network (DeepCoNN), innovatively using CNNs to process user review information, thus extracting user features [7]. Although these models have achieved good results in recommendation systems, none considered serialized information. The Convolutional Sequence Embedding Recommendation Model (Caser) drew inspiration from CNNs' image processing methods, treating an item sequence as an “image” and learning the serialization pattern of this “image” through convolutional filters [8]. The Caser model has requirements for datasets, performing well in datasets with high serialization density (such as MovieLens) but achieving average results in datasets with lower serialization density (such as Tmall). The LSTPM model, based on RNN, considers both short-term and long-term user interests, also analyzing the geographical relevance of recently browsed locations [9]. The ARNN model jointly models the sequential regularity and transition rules of similar positions, capturing contextual information controlling user mobility, thus obtaining more effective sequential rules [10]. The KRED model aggregates news entity information, using knowledge graphs and multi-task learning methods to achieve better recommendation results [11]. The ACPR algorithm improves recommendation accuracy by using a consumer-item-producer triplet interaction approach [12]. The UIG algorithm uses a graph Convolutional Neural Network-based method to learn user representation vectors from user knowledge graphs, then introducing these vectors into item knowledge graphs to effectively mine personalized user preferences [13]. The Chorus model mainly analyzes the relationship between user preferences for products and time changes, also considering the relationships between products. This paper comprehensively considers these two models, Caser and Chorus, with Caser focusing on the relationships between products and the changes in user preferences for products over time.

III. CERTU MODEL

The convolutional embedding recommendation model with time and user preference (CERTU), proposed in this article, utilizes knowledge graph to obtain the relationship between goods. Then, by analyzing the user purchase records within a specified time kernel function, it obtains the representation vector of items. The item representation vector will be learned through Convolutional Neural Network to capture its sequential features. CERTU takes into account the user's general interests and hobbies, transforms the user information into a user representation vector, and connects it with the final convolutional item vector. The CERTU model mainly consists of 3 layers: data processing input layer, convolution layer, and full connection layer. As shown in Figure 1, the item relationship is constructed into a knowledge graph [14], where the yellow lines represent the sequential purchase relationship between two items, indicating that their functions are complementary, while the blue lines represent that after purchasing one item, the user only viewed another item but did not buy it, indicating that their functions are similar. For this type of complementary items, users' interest in this type of items often shows a normal distribution trend over time [1]. For this type of similar items, users' interest in this type of items often shows two normal distribution trends over time. Users' interests and hobbies not only have short-term features but also long-term preferences. The item vector obtained from the data input processing layer will enter the horizontal convolution layer and vertical convolution layer for convolution calculation. The results will enter the full connection layer for connection calculation with the user's long-term preference vector to obtain the final prediction result.

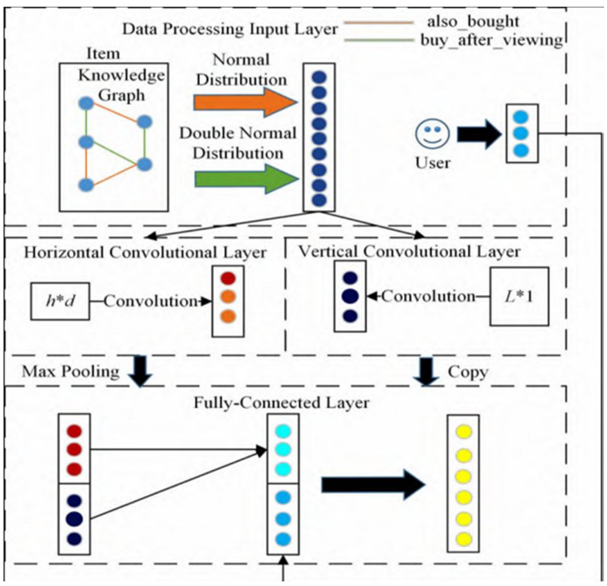

Figure 1: The flow of the methodology

3.1 Data Input Processing Layer

The data input processing layer mainly consists of two parts: items and users. There are complementary and alternative relationships among items. In order to fuse the information of items and the relationships between them, these information can be introduced into knowledge graph. The relationship between items can be constructed in the form of triplet [1], for example (headphones, complementary, mobile phone). Such triplet can be used as elements in the knowledge graph to construct the entire knowledge graph of item relationships. For items, this article groups items with complementary functions as one category. After purchasing a certain item, users will be interested in items that are complementary to that item. For example, after purchasing a mobile phone, users may have a strong interest in headphones. In a short period of time, users may be very interested in a certain item, but as time passes, they may purchase other complementary items, so their interest in this item will gradually decrease over time. The curve of users' interest and hobbies changing over time shows a normal distribution trend. The time function representing changes in interest over time is:

$$T(\Delta t) = N(\Delta t \mid 0, \sigma^c) \tag{1}$$

Among them, $N(\Delta t \mid 0,\sigma^c)$ represents a normal distribution with a mean value of 0, σ^c represents the variance of the normal distribution, and c is related to the category of items.

This article considers items with similar functions as another category. After users purchase items, they often lose interest or no longer need similar items in the short term. At this time, users' interests are often negative. However, as time passes, users' purchased items may experience wear and tear, and users may need similar items as substitutes. At this time, users' demand for similar items increases or their interest in them rises, and the recommendation system should recommend similar items to users. However, after users purchase the item, the situation of no longer being interested in it mentioned above occurs again, that is, users' interest in similar items often rises to a peak and then falls. Since users' interest in similar items is often negative at the beginning, the vertical coordinate of the curve should start with a negative value, so the time-varying curve is two normal distributions, one of which is a negative normal distribution. The calculation formula of the distribution function is as follows:

$$T(\Delta t) = -N(\Delta t \mid 0,\sigma^c) + N(\Delta t \mid \mu^c,\sigma^c) \tag{2}$$

Among them, $N(\Delta t \mid \mu^c,\sigma^c)$ represents the normal distribution, where μ^c represents the peak value of user interest in this type of items, which is related to the category of items c.

The user icon in Figure 1 represents the user's long-term interests and hobbies, while the Chorus model considers more the characteristics of user interests and hobbies over time, as well as the changes in user demand for items. For some users, they may always prefer certain types of items. In this case, considering the user's demand for items alone is often too one-sided, so this article considers the user's attributes as their long-term preferences.

3.2 Convolutional calculation layer

Convolutional computation has achieved great success in image processing. Convolutional computation is performed in two matrices, where there is a convolution kernel that moves at a certain step length in the input matrix and performs dot product calculations to obtain the output matrix. The Caser model applies CNN to movie recommendation, analyzes the cases of recommending multiple movies separately and recommending multiple movies together, and proposes horizontal convolution kernels and vertical convolution kernels for movie viewing records. For items, the recommendation results of multiple item purchase records and single item purchase records may be different in some cases. Therefore, for items, both horizontal convolution kernels and vertical convolution kernels are applicable.

This article divides convolutional layers into two types. One is the horizontal convolutional layer, which is mainly implemented through horizontal filters. The size of the horizontal filter is $h*d$, where d represents the number of columns of the horizontal filter, and $h \in \{1,\cdots,L\}$ represents the height of the horizontal filter. This article uses M^k to represent the horizontal filter, where k represents the serial number of the horizontal filter. This article defines the item embedding vector obtained from the data input processing layer of the previous layer as I, Therefore, the formula for the horizontal convolution of item i is:

$$item_h = f_a(I \odot M^k) \tag{3}$$

Where, $f_a(\cdot)$ represents the activation function of the convolutional layer, and $\odot$ represents the inner product operator.

The second is the vertical convolutional layer, which is mainly achieved through vertical filters. The size of the vertical filter is L*1, where L represents the height of the vertical filter, and the width of the vertical filter is 1 by default. The vertical filter will be applied from left to right and from top to bottom. The calculation formula is as follows:

$$item_v^k = I \odot F^k \tag{4}$$

Among them, F^k represents the k vertical filter, and $item_v^k$ represents the kth calculated item vector. Finally, the calculated item vectors are combined to obtain the vertical convolutional item representation:

$$item_v = [item_v^1 item_v^2 item_v^3 \cdots item_v^k] \tag{5}$$

3.3 Full Connection Layer

In this paper, the output of the horizontal convolution layer is connected to the output of the vertical convolution layer, and the fully connected layer combines the extracted features and outputs the result. The connection formula for the two convolution layers is:

$$g = f_b(W([\begin{smallmatrix} e_1 \\ e_2 \end{smallmatrix}] + b)) \tag{6}$$

Among them, $f_b(\cdot)$ represents the activation function of the fully connected layer, W represents the weight matrix, b represents the bias term, and g represents the vector representation obtained after connecting the item convolution.

IV. EXPERIMENTAL ANALYSIS

This article compares the CERTU model with the current state-of-the-art methods.

4.1 Experimental setup

4.1.1data set

The data used in the experiment is a publicly available Amazon dataset. The Amazon dataset includes user review IDs 、 basic information about items 、 buy_after_viewing and also_buy. In this paper, the buy_after_viewing in the Amazon dataset is used as the item substitution relationship, and the also_buy is used as the item complementation relationship. This article mainly uses the mobile phone and accessories dataset in the Amazon dataset, as listed in Table 1.

Table 1 Statistics of dataset

Dataset	Amazon Cellphones
Users	227900
Items	10300
Entry	193200

4.1.2 Evaluation indicators

In this paper, the experimental results are calculated using the left-one method, leaving only one sample as the test set and the other samples as the training set, and using the average of all the results

Measure the performance of your model. In this paper, HR (Hit Ratio) and NDCG (Nor-malized Discounted Cumulative Gain) were used as evaluation indicators. HR@K calculates whether the result of the correct prediction appears in the Top-K list, if it does, it is counted as a hit, and finally the hit rate is calculated. NDCG@K are more focused on where the prediction results are in the rankings.

4.1.3Compare models

(1) BPR[15]. This method proposes a double-sort loss optimization matrix factorization model, which is an excellent method for recommending non-serialized items based on implicit feedback data.

(2) Tensor0[16]. In this method, the historical context information is taken into account, and the traditional two-dimensional user-item matrix is replaced by n-dimensional tensors through tensor factor factorization.

(3) GRU4Rec[17]. The method uses the GRU to get the sort score of the recommended item.

(4) NARM[18]. This method improves the performance of session-based serialized recommendation

through GRU and attention mechanism.

(5) CFKG[19]. This method uses the TransE model to obtain the relationship between users and items, and between items and items.

(6) SLRC[20]. This method considers the user's habit of repeated consumption, and analyzes the user's long-term and short-term behavior.

(7) Caser[8]. In this method, horizontal filters and vertical filters are used to extract different item sequence pattern information, emphasizing the influence of short-term information.

4.2 Experimental results

A comparison of the experimental results of CERTU and the 7 baseline methods is shown in Table 2, with the horizontal axis representing the names of each method and the vertical axis representing the performance metrics. Compared to other methods, CERTU achieves better performance in both HR and NDCG. The BPR algorithm is a traditional collaborative filtering algorithm, which uses triplet information to represent the interaction between users and items. Tensor takes into account the information of the context by modeling the data as a tensor of a user, item, and context instead of the traditional user-item matrix, and Tensor is better than traditional collaborative filtering algorithms. GRU4Rec is a serialized recommendation, which uses the GRU model to derive the ranking score and then make recommendations, and its performance is better than that of the collaborative filtering algorithm. This also shows that the characteristics of serialization play a positive role in improving the recommendation effect. NARM introduces an attention mechanism on the basis of GRU and applies it to serialized recommendations, which also performs better than collaborative filtering algorithms. CFKG considers the relationship between the user and the item, converts it into a knowledge graph, and uses the TransE model to learn the graph embedding, which performs better than most baseline methods, indicating that the relationship between the user and the item can be introduced into the recommendation system to achieve better results. SLRC considers the effects of short-term and long-term effects, and introduces the Hawkes process into collaborative filtering, and the performance effect is the best among all baseline methods, indicating that the long-term and short-term interests of users in items are of great significance to improve the performance of the recommendation system. Caser uses Convolutional Neural Network (CNN) to extract the information of short-term sequences, and divides the sequence patterns into point level and parallel level, which has better performance than the collaborative filtering algorithm, indicating that the Convolutional Neural Network can obtain better results by introducing Convolutional Neural Network into the recommendation system.

Table 2 Performance comparison of different methods

Model	HR@5	HR@10	NDCG@5	NDCG@10
BPG	0.3357	0.4430	0.2309	0.2658
Tensor	0.3469	0.4315	0.2739	0.3012
Caser	0.3626	0.4723	0.2884	0.2524
NARM	0.4028	0.5014	0.2985	0.3303
CFKG	0.4023	0.5350	0.2833	0.3263
SLRC	0.4133	0.5140	0.3040	0.3368
GRU4Rec	0.4019	0.4997	0.3009	0.3325
CERTU	0.4359	0.5497	0.3153	0.3524

The CERTU model proposed in this paper works better than all baseline methods, suggesting that it is beneficial to consider the characteristics of user interests over time and the relationships between items. Based on the Convolutional Neural Network, the CERTU model divides the sequence patterns of long-term item information and short-term item information into point level and parallel level, which can enable the recommendation system to obtain richer information sources. Compared to the CFKG model, CERTU not only considers the relationship between items, but also incorporates the law of the relationship between items over time. Compared with Ca-ser, CERTU not only considers the sequence of short-term items, but also considers the long-term item information, and also introduces the characteristics of the relationship between items and the change of user interests over time, which can make the recommendation system more flexible and obtain better results.

4.3 Effectiveness of modeling the relationship between user interests and items

In order to further prove that the relationship between the characteristics of user interest over time and the item has a positive effect on the recommendation effect, this paper proposes the variants of the model, CERTU-I and CERTU-T, based on the CERTU model. The CERTU-I model does not consider the relationship between items, and ignores the complementary and substitution relationships between items. The CERTU-T model does not analyze the characteristics of user interest over time, and user interest will remain constant. By comparing these three models, we can verify the effectiveness of user interest and item relationship modeling on recommendations. Figure 2 shows the experimental results of the three models of CERTU, CERTU-I and CERTU-T, and it can be seen that the experimental results of CERTU are better than those of CERTU-I and CERTU-T, which indicates that the relationship between the characteristics of

Volume: 15 Issue: 06 Pages: 6156 - 6162 (2024) ISSN: 0975-0290

user interests over time and items can improve the recommendation effect of the recommendation system.

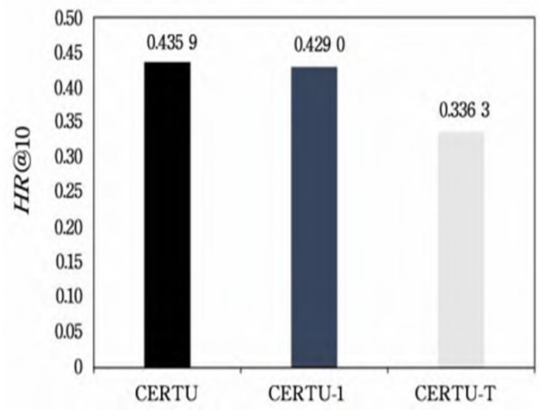

Figure 2 Effectiveness of user preference and the relationship of items

4.4 Parametric analysis

Figure 3 shows the effect of Learning rate on HR@10, where the abscissa is the x exponent in 10-x, and the learning rate decreases by a factor of 10 each time. It can be seen that the learning rate starts at 0.1 and as the learning rate gets smaller and smaller, the value of the HR@10 increases until the learning rate is 0.001 and HR@10 reaches its maximum with a value of 0.5497. When the learning rate is greater than 0.001, the HR@10 gradually decreases. From the above analysis, it can be seen that when the learning rate is 0.001, the best results can be obtained HR@10.

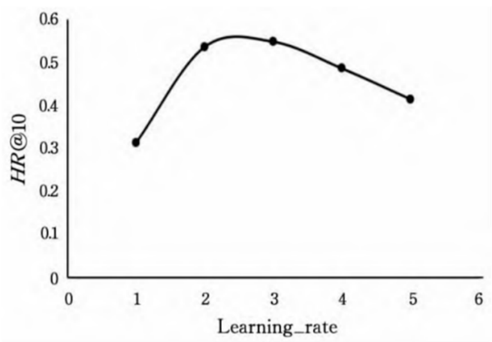

Figure 3 Influence of learning rate on HR@10

Figure 4 shows the effect of batch size on NDCG@10. In Figure 4, the abscissa batch size is 64 as the initial value, and each change increases by a factor of 2 for model training. As can be seen from the figure, the NDCG@10 decreases when the batch size changes from 64 to 128, because the value of the batch size is too small, the network convergence is unstable, and the NDCG@10 value is large in the oscillation. The batch size starts at 128 and the NDCG@10 begins to gradually increase. When the batch size is 512, the NDCG@10 reaches its maximum value of 0.3524. When the batch size is greater than 512, the NDCG@10 starts to decline. Within a certain range, the increase of batch size can improve the performance of the model to a certain extent, but with the continuous increase of batch size, the generalization ability of the model decreases, and finally the NDCG@10 value becomes smaller.

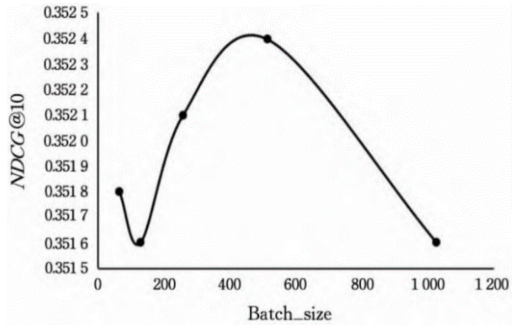

Figure 4 Influence of batch size on NDCG@10

Figure 5 shows the effect of the embedding vector dimension on HR@5. The abscissa embedding vector dimension in Figure 5 represents the dimension of the user and item vectors. As can be seen from the figure, the embedding vector dimension is the lowest when it is equal to 16, because the value of the embedding vector dimension is too small, which will lead to less user and item feature information obtained by the model, and the performance indicators obtained are lower. As the embedding vector dimension increases, so does the HR@5. When the embedding vector dimension increases from 32 to 48, although there is a brief decrease in the HR@5, the overall performance shows an upward trend. When the embedding vector dimension reaches 64, the HR@5 reaches the maximum value of 0.4359. As the value of the embedding vector dimension continues to increase, the HR@5 begins to decrease due to the overfitting of too many feature dimensions of the user and the item.

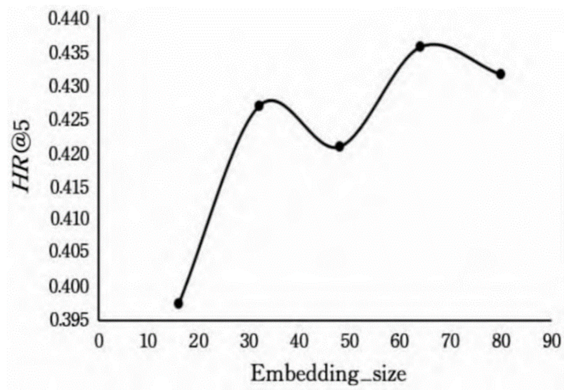

Figure 5 Influence of embedding size on HR@5

V. CONCLUSION

This paper comprehensively considers the complementary and substitution relationships between items, analyzes the user's preference for the item under these two relationships, and constructs it into a corresponding knowledge graph, which is used as the representation vector of the item. In this paper, we also consider the influence of a single item and multiple items in

Volume: 15 Issue: 06 Pages: 6156 - 6162 (2024) ISSN: 0975-0290

the item sequence on the user's preference for the next item, and use the horizontal filter and vertical filter to calculate the convolution of the item representation vector. Through the analysis and treatment of the above factors, this paper proposes the CERTU model, which is better than other baseline methods after experimental comparison, which reflects the effectiveness of the model.

In the future, we will improve the CERTU model from the following three aspects: (1) there may be the attribute of popularity in the item, and some items have a high degree of attention due to the publicity of the brand or their own good reputation, and the model should consider the attribute characteristics of the popularity of the item; (2) the curve of the user's interest in the item over time may not only have the characteristics of normal distribution, and the model will be beneficial; (3) The relationship between items may have a richer representation, and the mining of the relationship between items will be more in-depth in the future to improve the predictive power of the model.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant No. 61472348 and 61672455, by the Open Project Program of the State Key Laboratory of CAD&CG under Grant No. A2329, Zhejiang University, by the General Scientific Research Project of Zhejiang Provincial Department of Education under Grant No.Y202352364.

REFERENCES

[1] Wang, Chen yang, et al. "Make it a chorus: knowledge-and time-aware item modeling for sequential recommendation." Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 2020.
[2] Rendle, Steffen, Christoph Freudenthaler, and Lars Schmidt-Thieme. "Factorizing personalized markov chains for next-basket recommendation." Proceedings of the 19th international conference on World wide web. 2010.
[3] Wang, Pengfei, et al. "Learning hierarchical representation model for nextbasket recommendation." Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval. 2015.
[4] He, Ruining, and Julian McAuley. "Fusing similarity models with markov chains for sparse sequential recommendation." 2016 IEEE 16th international conference on data mining (ICDM). IEEE, 2016.
[5] Salakhutdinov, Ruslan, Andriy Mnih, and Geoffrey Hinton. "Restricted Boltzmann machines for collaborative filtering." Proceedings of the 24th international conference on Machine learning. 2007.
[6] LeCun, Y., et al. "Gradient-based learning applied to document recognition. P IEEE 86 (11): 2278–2324." (1998).
[7] Zheng, Lei, Vahid Noroozi, and Philip S. Yu. "Joint deep modeling of users and items using reviews for recommendation." Proceedings of the tenth ACM international conference on web search and data mining. 2017.
[8] Tang, Jiaxi, and Ke Wang. "Personalized top-n sequential recommendation via convolutional sequence embedding." Proceedings of the eleventh ACM international conference on web search and data mining. 2018.
[9] Sun, Ke, et al. "Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 01. 2020.
[10] Guo, Qing, et al. "An attentional recurrent neural network for personalized next location recommendation." Proceedings of the AAAI Conference on artificial intelligence. Vol. 34. No. 01. 2020.
[11] Liu, Danyang, et al. "KRED: Knowledge-aware document representation for news recommendations." Proceedings of the 14th ACM Conference on Recommender Systems. 2020.
[12] ZHANWJ.HONGZL.FANGL.P.et al. Collaborative Filtering Recommendation Algorithm Based on Adversarial Learning [J]. Computer Science,2021.48(7):172-177.
[13] Gu, Tianlong, et al. "Combining user-end and item-end knowledge graph learning for personalized recommendation." Journal of Intelligent & Fuzzy Systems 40.5 (2021): 9213-9225.
[14] BORDESA.USUNIERN.GARCIA-DURANA.etal.Tr ansl:-tingembeddingsformodelingmulti-relationaldata [J].Advances in Neural Information Processing Systetns.2013.26:2787-2795.
[15] Rendle, Steffen, et al. "BPR: Bayesian personalized ranking from implicit feedback. The 25th Conference on Uncertainty in Artificial Intelligence 2009: 452-161.
[16] Karatzoglou, Alexandros, et al. "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering." Proceedings of the fourth ACM conference on Recommender systems. 2010.
[17] ALAZS H. ALEXANDROS K. LINAS B. Session-based recommendations with recurrent neural networks International Conference on Learning Representations,2016:1-10.
[18] Loyola, Pablo, Chen Liu, and Yu Hirate. "Modeling user session and intent with an attention-based encoder-decoder architecture." Proceedings of the Eleventh ACM Conference on Recommender Systems. 2017.
[19] Zhang, Y., et al. "Learning over knowledge-base embeddings for recommendation. Special Interest Group on Information Retrieval,2018:8-1.
[20] Wang, Chen yang, et al. "Modeling item-specific temporal dynamics of repeat consumption for recommender systems." The world wide web conference. 2019.