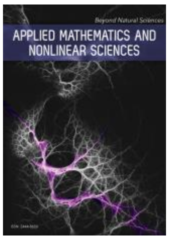
sciendo Applied Mathematics and Nonlinear Sciences, 9(1) (2024) 1-16



Applied Mathematics and Nonlinear Sciences

https://www.sciendo.com

Innovative Research on Deep Learning Algorithm in Intelligent Attractions Recommendation Technology

Yu Wang[1,†]

1. Party Committee Teacher Work Department (Personnel Office), Chengdu Polytechnic, Chengdu, Sichuan, 610041, China.

Submission Info

Communicated by Z. Sabir
Received January 10, 2023
Accepted April 14, 2023
Available online October 4, 2023

Abstract

In order to discover users' interest preferences from the massive travel information and meet users' needs for personalized tourist attraction recommendations. In this paper, we construct correspondingly according to the domain knowledge and complete the design and implementation of the mapping according to the corresponding process. The item representation enhancement module is designed to obtain a more comprehensive user preference by extracting information from the user's historical preferences and then using graph convolution to learn the correlation between the user's historical behavioral preferences and entities to enhance the KG semantic information of items. An item representation enhancement model based on RippleNet is proposed, from the item representation enhancement module to the interest module, to establish a connection between the user and the implicit information of the item so as to consider their mutual influence and represent the missing information more comprehensively, and finally, a personalized travel recommendation model based on knowledge graph and deep learning is constructed. From the results of the study, it is obtained that the model in this paper is 4.14% and 4.6% better than the best model SASRec in the comparison method in NDCG@10, Recall@10, MRR@10 indicators on both 1.93% higher. At the recommended list length K=20, the Recall rate Recall@20 has reached 80.78%. The superiority of the performance of the model constructed in this paper is demonstrated in the experiments, which brings technical innovation to the personalized attraction recommendation system.

Keywords: Attraction recommendation; Knowledge graph; Deep learning; RippleNet; Item representation enhancement model.
AMS 2010 codes: 68T05

†Corresponding author.
Email address: wangyu2015_cn@126.com

ISSN 2444-8656
https://doi.org/10.2478/amns.2023.2.00501

1 Introduction

With the rapid development of artificial intelligence, big data and other information technologies, empowering the most cutting-edge information technology to the development of the tourism industry has become a new trend in the current development of the tourism industry [1]. The rapid rise of various online travel services platforms, such as Ctrip, Where to Go, Catwalk and Ma Hive, has not only provided the latest and most comprehensive information for people's travel but also provided convenience for people's travel [2]. However, the Internet is full of diverse and complicated information, and the overall amount of information on the Internet is growing exponentially, so the problem of information overload is becoming increasingly serious. The presence of a large amount of information is beyond what people can afford, process and integrate valuable information, resulting in low utilization of information [3-4]. The emergence of recommendation systems, however, provides new ways for people to access effective information.

Recommender systems can autonomously mine the data that users need and push it to them. The research significance of the recommendation system is that it can help users to improve the efficiency of information acquisition, increase the activity of users in business platforms, and bring business value to enterprises. The accuracy of the recommendation list provided by the recommendation system affects the end-users perception and also determines whether the system can convert short-term users into long-term users [5-6]. Currently, users on travel websites can not only search for travel-related information but also share their photos of tourist attractions, express their opinions about tourist attractions, and interact with other tourists online [7]. All these can help the recommendation system to collect more diverse travel information from users, determine their preferences more precisely, and understand their changing interests at different times [8]. Applying recommendation technology to the field of tourism can better organize and process the collected user travel information for analysis, dig deeper into the connections between travel information, and provide stronger technical support for the development of the tourism industry, bringing more users and benefits to the online tourism service platform [9-10].

The literature [11] uses a user collaborative filtering technique to combine the friendship relationship between users and the geographical information between users and attractions to achieve personalized tourist attraction recommendations. However, due to the sparsity of the user-attraction matrix, the traditional collaborative filtering and matrix decomposition approaches face great challenges. The literature [12] proposed the personalized attraction similarity model (PSA) using images, texts and scores to obtain the similarity of attractions but did not consider the influence of the popularity of attractions on user preferences. The literature [13] et al. proposed activity and behavior-induced personalized recommendation systems as a hybrid approach to predict persuasive recommendations for tourist points of interest, mainly achieved by combining the user's activity behavior with location attributes.

The literature [14] mines users' check-in records and attraction category information combined with matrix decomposition methods for travel landmark recommendations. In their paper, literature [15] et al. focus on a comprehensive analysis of intelligent e-tourism systems using artificial intelligence techniques, comprehensively investigating the functions offered by these systems and their use of artificial intelligence techniques in the presence of different types of interfaces and diverse recommendation algorithms. The literature [16] proposed a travel recommendation method based on the dynamic topic model (DTM) and matrix factorization using the dynamic topic model to mine the explicit features of attractions and textual information in combination with MF for travel recommendation, ignoring the image information of the attractions. Literature [17] et al. proposed an algorithm called PersTour, where the algorithm provides personalized travel recommendations for tourists based on the user's interest attractions, the time spent at the interest point visit and the time of the most recently visited interest point. Literature [18] et al. proposed a personalized travel route recommendation system based on travel logs and shared photos and metadata information related to photos.

In this paper, we design a new item representation module with a knowledge graph-based recommendation method for research. The user behavior data that can be obtained on the platform is not comprehensive and cannot better express users' personal preferences. In this paper, we extract information from users' historical preferences to obtain a more comprehensive user preference and to distinguish each user preference more accurately. In order to be able to strengthen the connection between user and item implicit information, the association between the user's historical behavioral preferences and entities is learned by using graph convolution to enhance the KG semantic information of items. A new model of item representation enhancement based on RippleNet is proposed. The recommendation model based on the knowledge graph is studied, and the problems of the RippleNet model are analyzed. The importance of user-to-item relationships is reduced, and the lack of considering the role of interaction between users and implicit information of items leads to the performance of recommendations that cannot be more accurate. The item representation enhancement module, which enriches item information, enhances the KG semantic information of items, and then combined with the RippleNet model to discover the potential hierarchical interest information of users, establishes a connection between users and implicit information of items, and considers the interaction between them to get more accurate recommendation results.

2 Design and implementation of personalized travel recommendation system based on a knowledge graph and deep learning

2.1 System construction

2.1.1 General Overview

The personalized travel recommendation system based on a knowledge graph and deep learning combines the current popular knowledge graph and deep learning to discover the connection between each tourist attraction through the knowledge graph and then use deep learning to mine the user's interest preference for the attraction on the knowledge graph to achieve personalized recommendation for the user. This system targets the data of tourist attractions and analyzes the user's interest preferences of attractions through the user's behavioral data in this system to provide the users of tourism with assistance in decision making and better solve the problem of foreign tourists to selecting the attractions they are interested in from many tourist attractions in a limited time [19-20].

2.1.2 General framework

The system can be structured into four layers to describe the overall architecture of the system. The first layer is the data layer, which contains user behavior data and attraction data, mainly stored in MySQL, while the data for the model algorithm is stored in Neo4j, which is updated by MySQL. The algorithm layer includes the model algorithm and the popular recommendation algorithm. The model algorithm is the algorithm designed in this paper, which uses the user's behavioral data to calculate the user's preferred attractions through the model algorithm, and the popular recommendation algorithm is the algorithm that recommends the attractions based on the clicks and ratings of all users. The interface layer is used for the connection between the algorithm layer and the application layer, which is mainly responsible for passing the parameters obtained from the application into the algorithm layer application, and then outputting the results of the algorithm layer to the application layer. The application layer is the main function provided by this system, which is divided into two functions: personalized recommendation and popular recommendation. The specific architecture is shown in Figure 1.

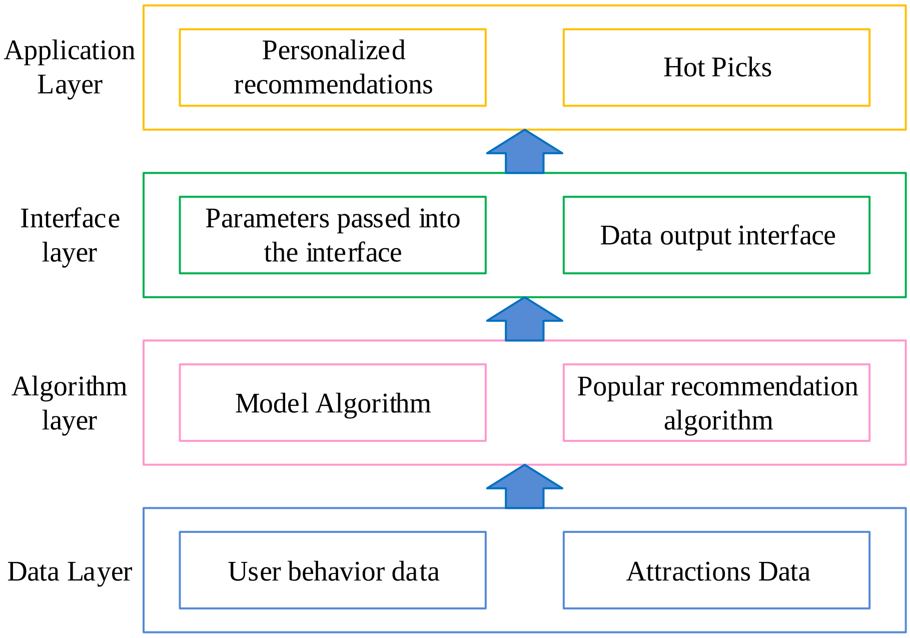


Figure 1. Overall system architecture

According to the above general architecture, the flow of this system is shown in Figure 2. The user applies the functions provided by the system according to the access to the front-end page. When the user is using some business that does not involve a model algorithm and knowledge graph, the front end will send the user's request to the business logic on the server side to complete the user's business requirements by calling the data returned from MySQL. When the administrator is updating the attraction data, the business logic will update the Neo4j database in addition to completing the MySQL database calls. When the user is using personalized recommendation, the business logic will input the user's behavior data from MySQL into the model algorithm, and then call the Neo4j database to get the attraction knowledge map to realize the prediction of personalized recommendation, and finally return the prediction result to the front-end page.

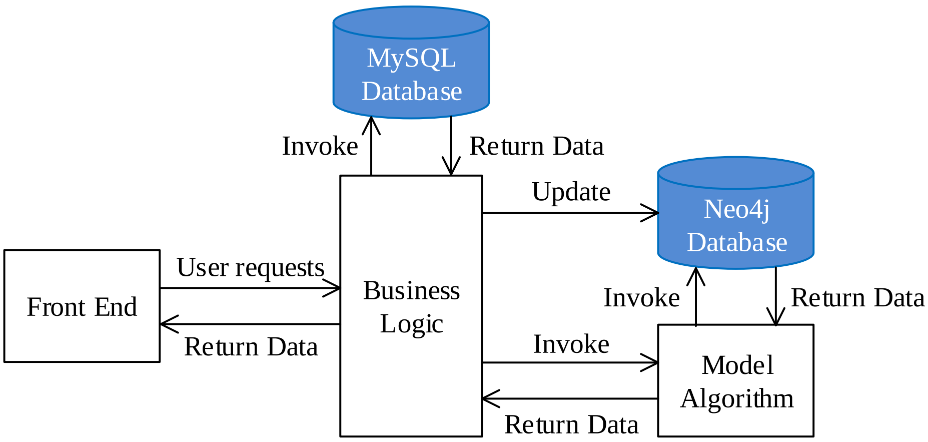


Figure 2. Overall system flow chart

2.2 A framework of personalized travel recommendation model based on knowledge graph and deep learning

In the recommendation scenario, suppose we have a set of user settings $U=\{u_1,u_2,\ldots,u_m\}$ and a set of item sets $V=\{v_1,v_2,\ldots,v_n\}$. Based on implicit user feedback, define the user set U and item set V to form a user-item interaction matrix $Y \in \mathbb{R}^{m\times n}$:

$$y_{uv} = \begin{cases} 1, \text{if interaction } (u,v) \text{ is observed;} \\ 0, \text{otherwise.} \end{cases} \tag{1}$$

where $y_{uv}=1$ denotes an implicit interaction between user u and item v, such as the act of viewing, clicking or purchasing, otherwise $y_{uv}=0$. In addition to the interaction matrix Y, we have the knowledge graph G of additional information, which is composed of $\{(h,r,t) \mid h\in E, r\in R, t\in E\}$ triples, R and E denote the set of relations and the set of entities in the knowledge graph, respectively.

For a given user-item interaction matrix Y and knowledge graph G, the task of our model is to predict whether a user u is potentially interested in an item v with which he/she has not interacted before. In general, our model focuses on learning the prediction function $\hat{y}_{uv}=F(u,v\mid\theta,Y,G)$. Where $\hat{y}_{uv}$ denotes the probability of a user u with a predicted clicked item v, F denotes the function, and θ denotes the parameters of the model [21].

Eir-Ripp takes this user's historical interest S_u^0 and an item v as inputs and outputs the predicted probability of a user u clicking on an item v, where $u\in U, v\in V, U$ denotes all users and V denotes all items. Note that the user's historical interest here is the set of items that the user liked in the past. The historical interest S_u^0 of the user u (light green matrix) is obtained by a series of APReLU, Transformer Layer and pooling to obtain the information-extracted user historical interest $h\_history$ (blue block), and the $h\_history$ and item v are put into the graph convolutional neural network to obtain the item vector $item\_embedding$ (orange block) that represents the information of user-item interaction. $item\_embedding$ Iterative interaction is performed by ripple set to obtain the user's u interest vector for the item v (light green block), which is then summed up to represent the user's u personal interest vector $user\_embedding$ (gray block). The $user\_embedding$ and $item\_embedding$ are concatenated to obtain the splicing vector (orange-red blocks), and finally, the predicted probability is calculated by two fully connected layers FC $\hat{y}$. The framework of the Eir-Ripp model is shown in Figure 3.

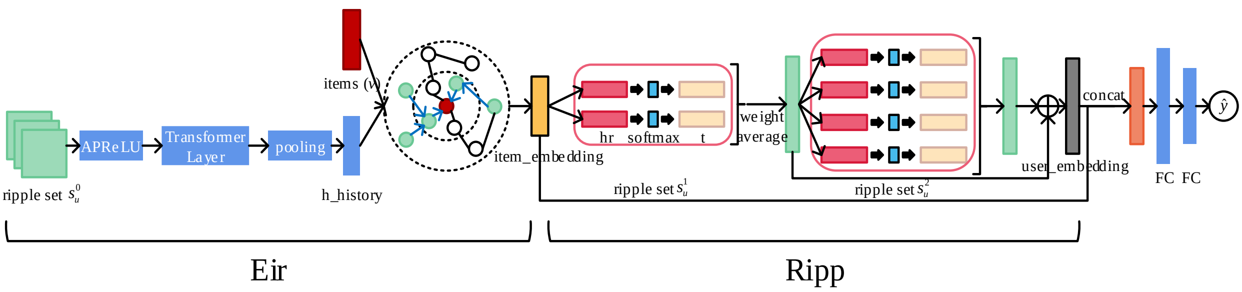


Figure 3. Eir RIPP model framework

2.3 Building a personalized travel recommendation model based on knowledge graph and deep learning

Eir-Ripp consists of three parts in total, one part is Eir, an item representation enhancement module that enriches item information; another part is Ripp, an interesting module that discovers potential layers of user interest; and the last part is a fully connected layer that merges user interest and item representation and performs deep mining. Each of these three parts is described in detail in three subsections below.

2.3.1 Item representation enhancement module

In this paper, the information about attractions obtained from the Internet is limited, and there is still a large amount of information that is still used and accessible; therefore, item representation cannot be performed well simply with the current information. In this regard, this paper designs an item representation enhancement module, which extracts information from users' historical preferences to get a more comprehensive user preference, and then learns the correlation between users' historical behavioral preferences and entities by using graph convolution, thus strengthening the interaction of implicit information between users and items and enhancing the KG semantic information of items.

The item representation enhancement module first extracts the global information of positive features and the useful information in the global information of negative features from the user's historical interest S_u^0 through APReLU to get a more comprehensive user preference, then mines the implicit information between items in the user's historical interest through Transformer Layer, and then uses pooling to reduce the number of parameters to get the user u extracted After the historical interest $h\_history$. Finally, the predicted item v is convolved within a local graph, and the weights of different neighbors in the local domain depend on the relationship between the historical interest of a particular user u and the items so that the implicit information between the user and the items interacts with each other as follows:

$$\pi_r^u = g(u, r) \tag{2}$$

π_r^u denotes the degree of preference of the user u for relation r, where $u, r \in \mathbb{R}^d, \pi_r^u \in \mathbb{R}$. The g function here is used to calculate the degree of preference of the user with different relations:

$$v_{N-(v)}^{h\_history} = \sum_{S_u^0 \in N(v)} \tilde{\pi}_{r_{v,S_u^0}}^{h\_histroy} S_u^0 \tag{3}$$

$$\tilde{\pi}_{r_{v,s_u^0}}^{h\_histroy} = \frac{\exp\left(\pi_{r_{v,s_u^0}}^{h\_histroy}\right)}{\sum_{S_u^0 \in N(v)} \exp\left(\pi_{r_{v,s_u^0}}^{h\_histroy}\right)} \tag{4}$$

$v_{N(v)}^{h\_hstory}$ denotes the topological domain structure of user u to item entity v, $\tilde{\pi}_r^{h\_histroy}$ denotes the normalization of user relationship score, $N(v)$ denotes the set of entities directly connected with item v in KG, and the size of the set is large in order to control the computational pressure brought by too many neighbors, here we adopt the method of randomly selecting k a fixed number of neighbor nodes and set the parameter hop_n to control the depth of the neighborhood:

$$S(v) = \{e \mid e \in N(v), \mid S(v) \mid = k\} \tag{5}$$

$v_{N(v)}^{h\_history}$ obtains $v_{S(v)}^{h\_history}$ the set of nodes representing item v the neighborhood k of the entity based on the set of neighborhood nodes $S(v)$. The entity v and its neighborhood $v_{N(v)}^{h\_history}$ are summed and aggregated into a vector and then nonlinearly transformed as follows:

$$item\_embedding = \sigma\left(W^{Eir}\left(v + v_{S(v)}^{h\_history}\right) + b^{Eiri}\right) \tag{6}$$

where $W^{Eir} \in \mathbb{R}^d$ is the weight of the fully connected layer and $b^{Eir} \in \mathbb{R}^d$ is the bias term.

2.3.2 RippleNet Module

For the input user u, the items he has clicked and liked in the past are used as his historical interest S_u^0 and S_u^0 is used as the seed of KG in the user u's historical interest set, which then spreads along the links to the periphery, forming multiple ripple sets S_u^{hop}:

$$\varepsilon_u^{hop} = \left\{t \mid (h,r,t) \in G, h \in \varepsilon_u^{hop-1}\right\}, hop = 1, 2, \ldots, H \tag{7}$$

$$S_u^{hop} = \left\{(h,r,t) \mid (h,r,t) \in G, h \in \varepsilon_u^{hop-1}\right\}, hop = 1, 2, \ldots, H \tag{8}$$

where G denotes the KG of the whole item, ε_u^{hop} denotes the set of entities of hop hops of the user u, and S_u^{hop} denotes the set of neighbors of hop hops of the user.

Using $item\_embedding$ as the item v input to the prediction, calculate the similarity p_i with the (h,r,t) normalization in the first diffusion of the ripple set:

$$p_i = soft\max\left(item\_embedding^T h_i r_i\right) = \frac{\exp\left(item\_embedding^T h_i r\right)}{\sum_{(h,r,t)\in S_u^1} \exp\left(item\_embedding^T h_i r\right)} \tag{9}$$

The interest representation of user ripple set S_u^1 hop is obtained by weighting the sum of t in the first layer 1 according to the similarity:

$$o_u^1 = \sum_{(h_i,r_i,t_i)\in S_u^1} p_i t_i \tag{10}$$

Repeating the operation of equation (6) above, the interest vector $user\_embedding$ of the user u is obtained:

$$user\_embedding = o_u^1 + o_u^2 + \ldots + o_u^H \tag{11}$$

2.3.3 Forecasting module

Through the calculation of the previous two modules, the item vector $item\_embedding$ containing the information on user-item interaction and the interest vector $user\_embedding$ of the user u is obtained. These two vectors are spliced horizontally by concat, input to FC1 of dimension (2*dim, dim) and activated by ReLU, and then the output values are input to FC2 of dimension (dim, 1) and calculated using the sigmoid function σ:

$$FC\_input = Connect(item\_embedding, user\_embedding) \tag{12}$$

$$\hat{y}_{uv} = \sigma\left(W_2 \cdot \mathrm{ReLU}\left(W_1 \cdot FC\_input + b_1\right)\right) + b_2 \tag{13}$$

2.4 Algorithm optimization

The loss function of the Eir-Ripp model:

$$L = L_{RS} + L_{Eir} + L_{Ripp} \tag{14}$$

The loss of Eir-ripp consists of three components, the loss of the recommendation module, the loss of the Eir module and the loss of RippleNet:

$$L_{RS} = \sum_{u\in U, v\in V} f\left(\hat{y}_{uv}, y_{uv}\right) \tag{15}$$

For the recommendation module. u denotes the set of users, v denotes the set of items, and f denotes the cross-entropy function:

$$L_{Eir} = \frac{\lambda_{12}}{2}\left\|user\_embedding\right\|_2^2 + \left\|v_{N(v)}^{h\_history}\right\|_2^2 \tag{16}$$

For the Eir module, λ_{12} denotes the balance weight of the L2 regular term, $user\_embedding \in ¡^{d}$ is the user interest vector extracted by Eir, and $v_{S(v)}^{h\_history}$ denotes the set of entity v neighborhood nodes:

$$L_{Ripp} = \lambda_{kge} \left\| MEAN(\sigma(hrt) \right\|_2^2 + \lambda_{12} \left(\left\| h \right\|_2^2 + \left\| r \right\|_2^2 + \left\| t \right\|_2^2 \right) \tag{17}$$

For the Ripp module, $\lambda_{\mathrm{kg}e}$ is the equilibrium weight of KGE, $MEAN$ function represents the average value, and hrt is the product of entities and relations in triplet (h,r,t) to represent KGE.

3 Experimental analysis

3.1 Baseline Methodology

S-POP: Popularity prediction is based on the most frequent items in the training set, and the method recommends the most popular items in the current session.

BPRMF: Based on the co-occurrence of the items to be predicted, we find users who have also visited the items to be predicted and generate a recommendation list of length K by calculating the similarity between the items to be predicted and other items in the sequence of the user’s visits, and ranking the attractions with the highest similarity to the target attractions from highest to lowest. This method is one of the most commonly used recommendation methods in traditional recommendation systems.

MLP: This method uses a multi-layer perceptron to embed users and items to learn scoring functions from the data.

FMPC: This method combines matrix decomposition with Markov chains for the next recommendation, learning the user’s preferences and sequential behavior.

GRU4Rec: A pioneering approach, this method applies the recurrent gated unit GRU to the recommendation domain for the first time, modeling user sequence behavior, using parallel small-batch training and a loss function based on a ranking method to learn the model.

GRU4Rec+: The authors of GRU4Rec subsequently proposed the GRU4Rec+ model, which uses small batches of processed data and a new loss function to maximize positive sample scores as a way to improve training efficiency and accuracy.

NARM: Using the attention mechanism to capture the user’s main interest from the hidden layer state, combined with the sequence behavior as the final feature vector representation to generate the recommendation list.

SASRec: The first translation model in the natural language domain is applied to the recommendation domain with good results. This method only uses the self-control mechanism to model user sequences and considers that users pay different attention to different items, so different weights are assigned.

3.2 Evaluation Methodology

Considering that the items that users may choose in practical applications are located in the first few items of the recommendation list. Therefore, the following evaluation metrics commonly used in recommendation systems are used in this paper.

HR@10 indicates whether the correctly recommended items are in the predicted recommendation list among the items with a predicted recommendation list length of 10. The larger the HR value is, the higher the hit rate is and the better the recommendation effect is:

$$HR@10=\frac{\sum_{N-1}^{N} hit}{N} \tag{18}$$

N denotes the total number of tests, and for each attraction to be tested, hit takes the value 1 or 0, hit=1 means the item to be predicted is in the top K positions of the ranking list L, and hit=0 means it is not in the top K positions of the ranking list L.

MRR@10 indicates that the correctly recommended item is located in the top ranking, and the value is averaged after taking the reciprocal. When the recommended item is not in the top 10, the value is set to 0. A larger MRR value indicates that the correct item is at the top of the ranking list:

$$MRR@10=\frac{\sum_{t\in L}\frac{1}{Rank(t)}}{N} \tag{19}$$

The evaluation metric HR=Recall appears in this paper. NDCG@10 is a measure of the position of a positive example in the ranking list. Using the leave-one-out method to evaluate the attractions to be predicted, there is only 1 attraction to be predicted, so IDCG@10 ideally a particular attraction is ranked in the first position, for which NDCG@10 is simplified as follows:

$$NDGG@10=\frac{1}{N}\frac{\log_2(2)}{\log_2(Rank(t)+1)} \tag{20}$$

$Rank(t)$ indicates the position ranking of the item to be predicted in L, and N is the number of users to be tested.

3.3 Parameter setting

The Eir-Ripp model uses the TensorFlow framework with default parameters self-attention mechanism block b=2, vector dimension d=100, learning_rate=0.001, batch bath_size=64, dropout=0.2, multi-head=2, and visitor input length maxlen=10. The loss function is optimized using the Adam gradient descent method, initialized weights, and learnable parameters such as bias are generated using a truncated standard Gaussian distribution.

3.4 Comparison of different algorithms

The traditional method S-POP, BPRMF, has relatively poor performance, and Figure 4 shows the algorithm comparison results. Due to the different personal preferences, tourists will consider the attributes of the attractions comprehensively when visiting, so such traditional methods have been discarded by the market for recommending the list of attractions only based on the user's feedback. The MLP model learns the high-level interaction semantic information of tourists and attractions through the multi-layer neural network, and its recommendation effect is improved than the traditional methods, but it does not consider the sequential location information; however, in the sequential behavior recommendation; the tourists' sequence sequential order is crucial. In the FPMC model, although the Markov chain-based modeling of tourists' sequential behavior is effective, the Markov chain only considers that the state of tourists at any moment is only related to the previous state, while some tourists' behaviors may be continuous and not only related to the previous moment. The recommendation results of GRU4Rec, GRU4Rec+, and NARM are all relatively good, proving that such methods as recurrent neural networks have an advantage in handling both temporal information and sequential user behavior, GRU4Rec, GRU4Rec+ model only sequential visitor behavior, while NARM considers both the degree of visitor preference for different attractions. SASRec using the most popular self-attention mechanism in recent years on the attraction dataset to take effect with this paper Eir-Ripp has been similar, also shows the effectiveness of the self-attention mechanism approach in sequence modeling and modeling different preferences of users, but the above methods do not consider the relationship between attractions and attractions, the impact of multiple attributes of attractions on visitor behavior, from the results can be seen Eir-Ripp model than the best model SASRec in the comparison methods in NDCG@10, Recall@10, MRR@10 indicators on both 1.93% higher, 4.14%, and 4.6%.

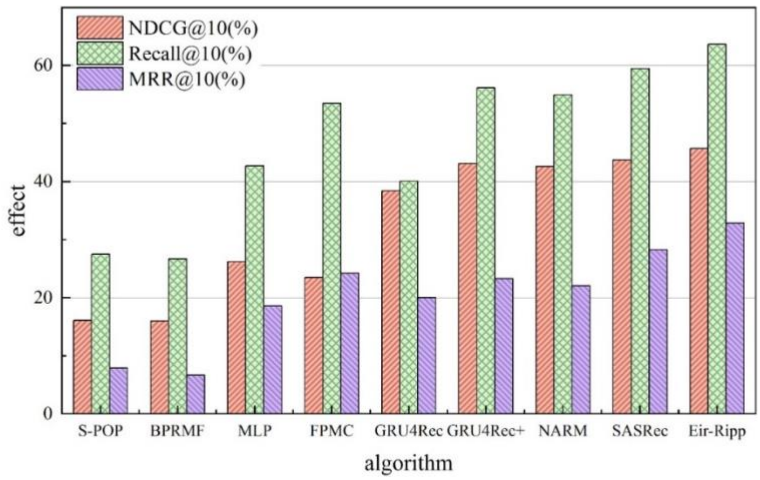


Figure 4. Comparison of algorithms

3.5 Effect of different dimensions on experimental results

In order to explore the influence of the dimensionality of the attractions on the experimental results by conducting experiments on the attractions in different dimensions, the results are shown in Figure 5. On the tourism dataset, the three recommendation effects of NDCG@10, Recall@10, and MRR@10 gradually increased as the dimensionality increased, reaching the best at dimension d=150, after which there were slight fluctuations in NDCG@10 and Recall@10 and a decrease in MRR@10 as the dimensionality continued to increase.

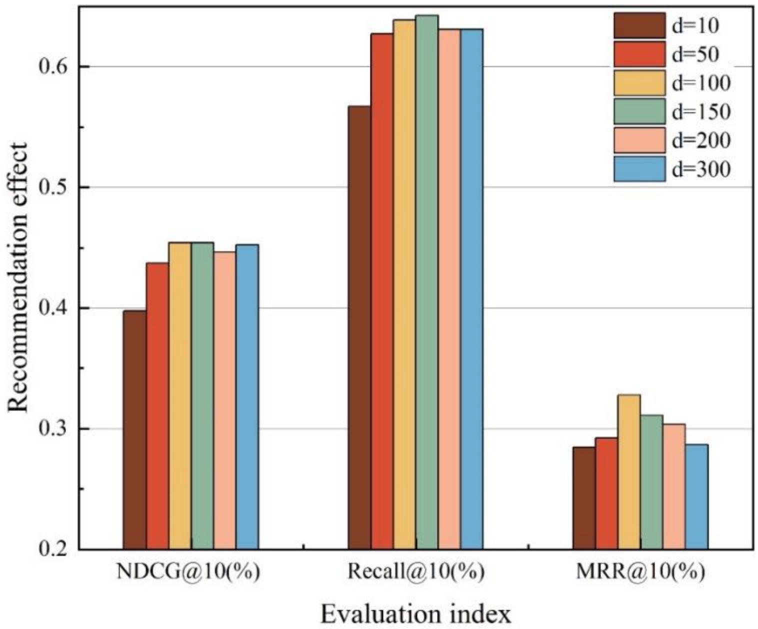


Figure 5. NDCG@10, Recall@10 and MRR@10 in different dimensions

3.6 Effect of different recommended list lengths on experimental results

In this paper, the experiments were selected in the case of recommendation list length from 1 to 20, and the NDCG, Recall, and MRR under different recommendation list lengths are shown in Fig. 6. With the increasing recommendation list length of attractions, the recommendation effects under different indicators are improving, and the sensitivity of different indicators to the parameters is not the same due to their different calculation methods. At the recommendation list length K=20, the recall Recall@20 has reached 80.78%, which indicates the effectiveness of the Eir-Ripp model multi-source information on the experimental results.

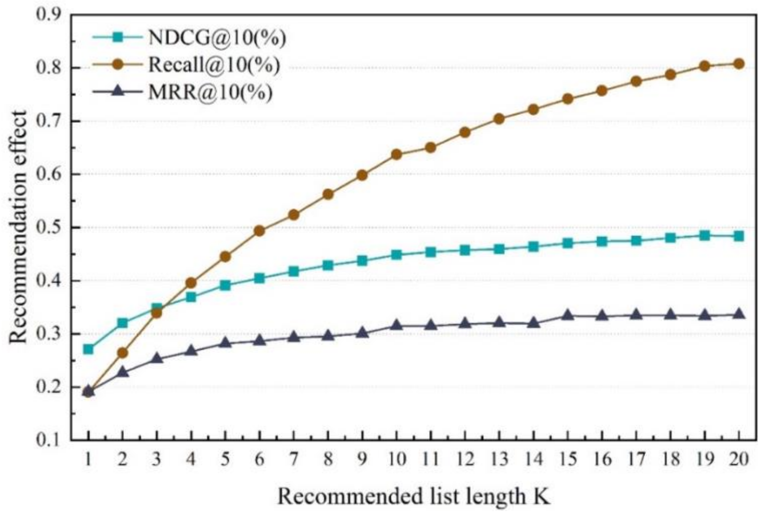


Figure 6. NDCG, Recall and MRR under different recommendation list lengths

3.7 Effect of different training times on experimental results

The NDCG@10, Recall@10, and MRR@10 under different training rounds are shown in Figure 7. Initially, the model is not effective, but as the model training deepens, the values of each index increase, which means that the recommendation effect is improving, and the best effect is reached when the training reaches 24 rounds, and then, NDCG@10 and Recall@10 tend to be stable, and MRR@10 slightly decreases, which is due to the fact that the model has reached overfitting.

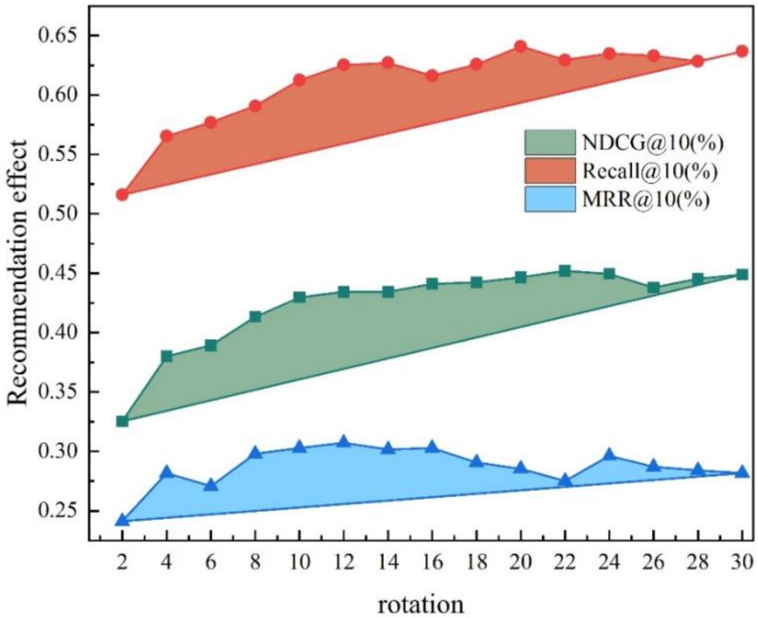


Figure 7. NDCG@10, Recall@10 and MRR@10 under different training rounds

3.8 Experimental disambiguation

1) A location embedding

The results of the self-attention mechanism are the same if the location embedding of attractions is not used, i.e., the order of tourists' attraction tours is not considered, and the attention weight of each attraction depends only on the attraction embedding. The experimental results show that the influence of location is not significant on the tourism dataset; the reason is that the attractions as special items; for some of the adjacent popular attractions, the order of tourists' successive tours is not very important, unlike commodity shopping which has a strong connection, after users buy a new cell phone, they will subsequently buy a protective case, and attraction recommendations such as some items in the amusement park may require queuing, tourists may play those items that do not require queuing items.

2) Attribute embedding of B attractions

The experimental results show that when the attribute embedding of attractions is not used, the results drop, indicating that the attribute information of attractions provides a reference value for tourists as an additional information supplement, such as parent-child tour tourists may prefer frolic and play items.

3) C Dropout

Dropout is set to prevent overfitting. The algorithm comparison is shown in Table 1, and the results are better when Dropout is used than when it is not.

4) D-F self-attentive block

The results showed that the best results were achieved with b=2 on the tour date. This implies that a hierarchical self-attentive structure is more helpful in learning the semantics of complex attraction interactions.

5) G-I multi-head

It can be seen from the table that the G-F experiments show that the results are not improved but somewhat decreased on the tourism dataset, presumably due to the small size of the dataset and the low dimension d. Because the dataset size used by Google in Transformer is quite large, and the default dimension is d=512.

Table 1. Algorithm comparison

	Evaluation index	NDCG@10(%)	Recall@10(%)	MRR@10(%)
	default	0.4420	0.6306	0.3309
A	Remove P	0.4454	0.6120	0.2861
B	Remove A	0.4509	0.6121	0.2825
C	Removal Dropout	0.4371	0.6143	0.2865
D	b=0	0.4311	0.6226	0.2849
E	b=1	0.4465	0.6200	0.2926
F	b=3	0.4492	0.6310	0.3109
G	multi-head=1	0.4520	0.6559	0.3189
H	multi-head=4	0.4270	0.6515	0.3050
I	multi-head=10	0.4293	0.6374	0.3077

4 Conclusion

In this paper, the way of attraction recommendation is a Top-N recommendation list, using the trained personalized travel recommendation model based on knowledge graph and deep learning to predict the click probability of all attractions for each user and rank the attractions according to their click probability to get the preliminary attraction recommendation list. The attractions that have been visited by the predicted users are removed, the attractions that have not been visited are kept as candidate attractions, and finally, the top N attractions with the highest click probability are selected as the list of attraction recommendations from the retained candidate attractions.

In the experiments comparing different algorithms, the model in this paper is 1.93%, 4.14%, and 4.6% higher than the best model SASRec in the comparison method in terms of NDCG@10, Recall@10, and MRR@10 metrics.

In the analysis of the effect of different recommendation list lengths on the experimental results, the recall Recall@20 has reached 80.78% at the recommendation list length K=20, which indicates the effectiveness of the multi-source information of the model in this paper on the experimental results.

In the analysis of the effect of different training times on the experimental results, the model's initial effect is not good, but with the deepening of the model training, the value of each index increases; that is, the recommendation effect is improving, and the best effect is reached when the training is up to 24 rounds.

References

[1] Chiang, H. S., Huang, T. C. (2015). User-adapted travel planning system for personalized schedule recommendation. Information Fusion, 21:3-17.

[2] Chen, Gencai, Zhenxing, & Ling. (2015). Topic based context-aware travel recommendation method exploiting geotagged photos. Neurocomputing.

[3] Chen, L., Cao, J., Wang ,Y., et al. (2022). Multi-view Graph Attention Network for Travel Recommendation. Expert Systems with Applications, 191:116234-.

[4] Kesorn, K., Juraphanthong, W., Salaiwarakul, A. (2017).Personalized attraction recommendation system for tourists through check-in data. IEEE access, 5, 26703-26721.

[5] Xueting, L., & Jan, N. (2022). Personalized recommendation algorithm of tourist attractions based on transfer learning. Mathematical Problems in Engineering.

[6] Park, S., & Kim, D. Y. (2017). Assessing language discrepancies between travelers and online travel recommendation systems: application of the jaccard distance score to web data mining. Technological Forecasting & Social Change, 123(oct.), 381-388.

[7] Chen, C. C., & Tsai, J. L. (2017). Determinants of behavioral intention to use the personalized location-based mobile tourism application: an empirical study by integrating tam with issm. Future Generation Computer Systems, 96(JUL.), 628-638.

[8] Chen, L., Cao, J., Zhu, G., et al. (2021). A multi-task learning approach for improving travel recommendation with keywords generation. Knowledge-Based Systems, 233, 107521-.

[9] Zhang, L., Xu, J., Gong, Y., et al. (2021). Unsupervised Image and Text Fusion for Travel Information Enhancement. IEEE Transactions on Multimedia, PP(99), 1-1.

[10] Wen, Y. T., Yeo, J., Peng, W. C., & Hwang, S. W. (2017). Efficient keyword-aware representative travel route recommendation. IEEE Transactions on Knowledge and Data Engineering.

[11] Zhang, Z., Pan, H., Xu, G., et al. (2016). A Context-Awareness Personalized Tourist Attraction Recommendation Algorithm. Cybernetics and Information Technologies, 16(6), 146-159.

[12] Shen, J., Deng, C., & Gao, X. (2016). Attraction recommendation: towards personalized tourism via collective intelligence. Neurocomputing, 173, 789-798.

[13] Logesh, R., Subramaniyaswamy, V., Vijayakumar, V., & Li, X. (2018). Efficient user profiling based intelligent travel recommender system for individual and group of users. Mobile Networks and Applications, 24(3), 1018-1033.

[14] Shi, Y., Serdyukov, P., Hanjalic, A., et al. (2013). Nontrivial landmark recommendation using geotagged photos. ACM Transactions on Intelligent Systems and Technology (TIST), 4(3), 47.

[15] Borràs, J., Moreno, A., Valls, A. (2014). Intelligent tourism recommender systems: A survey.Expert systems with applications, 41(16), 7370-7389.

[16] Xu, Z., Chen, L., Dai, Y., et al. (2017).A dynamic topic model and matrix factorization -based travel recommendation method exploiting ubiquitous data. IEEE Transactions on Multimedia, 19(8), 1933-1945.

[17] Lim, K. H., Chan, J., Leckie, C., et al. (2018). Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. Knowledge and Information Systems, 54(2), 375-406.

[18] Jiang, S., Qian, X., Mei, T., et al. (2016). Personalized travel sequence recommendation on multi-source big social media. IEEE Transactions on Big Data, 2(1), 43-56.

[19] Yang, M. (2022). An Intelligent Recommendation Method for Tourist Attractions Based on Deep Learning. Computational Intelligence and Neuroscience.

[20] Leask, A. (2016). Visitor attraction management: A critical review of research 2009–2014. Tourism Management, 57, 334-361.

[21] Zhang, H., Wu, Y., Buhalis, D. (2018). A model of perceived image, memorable tourism experiences and revisit intention. Journal of destination marketing & management, 8, 326-336.