Travel Attractions Recommendation based on Max-negative the Gated Recurrent Unit trajectory mining Representation

Shunyao Zhang¹, Phatpicha Yochum¹, Chenzhong Bin¹* and Liang Chang¹

¹Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, Guangxi, 541000, China

Corresponding author’s e-mail: binchenzhong@guet.edu.cn

Abstract. Although the traditional recommendation algorithm has achieved good results in the field of travel recommendation, due to the lack of data, the cold start and data sparseness problems and the neglect of the semantic problems hidden in the travel track, the low recommendation accuracy remains unresolved. Recently, the RNN model performed very well in recommending system sequence learning. We use RNN to model the travel sequence in the travel recommendation. In the pairwise, we can achieve a more accurate recommendation by effectively processing the negative samples and then training to generate a smaller loss function. Our method (Max-GRU) is optimized by adding additional negative sample and finding Max-negative on the Gated Recurrent Unit trajectory mining Representation Model. On the Shanghai tourism data and Guilin tourism data, both MRR@10 and RECALL@10 have been significantly improved compared to the use of the RNN model and baselines.

1. Introduction

In the recommending system and travel recommendation, the traditional method has been widely applied such as content-based recommendation [1][2], collaborative filtering based recommendation [3][4], mixed recommendation [5], etc. The content-based recommendation establishes attribute information based on the visitor's visited attractions, and then creates a user portrait for the visitor, and then calculates the similarity between the user portrait and the attraction attribute data, and recommends the similarity attraction to the visitor. The establishment of the tourist portrait depends on the learning method used. There are commonly used decision trees, Bayesian classifiers [6][7] and vector-based representations. Content-based recommendations require long-term historical preference information of visitors to create user portraits. User portraits may change as visitors' preferences change. The disadvantage is that the cold start problem cannot be solved, using this method can't find and recommend new attractions and the information in the sequence of attractions is not taken into account. Recommendations based on collaborative filtering [8][9] are one of the most successful technologies in the recommendation system. The recommendation based on collaborative filtering defines the similarity between "tourist-tourist" and "attractions-attractions" according to the long-term historical preference information of the tourists, and the recommendation based on the similarity of the tourists and does not take into account the information in the attractions sequence. Both of these recommendations are faced with cold start problems caused by data sparsity and the inability to obtain long-term historical preference information from visitors. For example, 1) tourists usually choose places that they have never been to as destinations, resulting in the sparseness of the "tourist-attraction" correlation matrix. 2) When encountering a new tourists, the historical information of the tourists cannot be obtained to construct a user matrix to make a recommendation. Both of the
above recommendations use only information between individual attractions and tourists, ignoring the information in the tourist trajectory. The RNN neural network [10][11] stands out in solving the sequence problem. It solves the cold start problem of calculating the matrix similarity of "tourists-tourists" and "attractions-attractions", and takes into account the information of the entire tourist track, effectively digging into the scenic spots. The hidden semantics between them enable high quality recommendations. Recently, RNN has been widely used in the recommendation field, and RNN has combined with session-based recommendations to achieve high quality recommendations.

Inspired by session-based recommendations [12], the tourist trajectory visited by tourists is treated as a session. Our approach takes into account the entire tourist trajectory and relies on all the attractions of the tourist sequence to make recommendations. The advantage of the session-based recommendation method is that it does not need to acquire the attribute information of tourists and attractions to construct the information matrix for calculation, so the cold start problem caused by the sparsity of the "visitor-attraction" correlation matrix is solved. At the same time, it also realizes the sequence learning in the tourist trajectory, effectively mining the high-level semantics hidden between the scenic spots and the scenic spots in the tourist trajectory, and achieving high-efficiency recommendation. The loss function uses the pairwise method. We think that negative sample scores greater than positive sample scores are valid negative samples and vice versa is an invalid negative sample [13]. If the score of the positive sample in the recommendation list is already in front of all negative scores, the positive sample is already in the correct position and there is no need to update the recommended list. Therefore, we choose the negative sample with the highest score as the largest negative sample. The other negative samples score are compared with the largest negative sample score to get the weight. The higher the score of the negative sample, the greater the weight obtained. The weight of the invalid negative sample is small, and the impact on the loss function will be small. The loss function will become smaller. This idea makes our model's RECALL @ 10 and MRR @ 10 get better results on the Shanghai Tourism Dataset and the Guilin Tourism Dataset compared to the baseline of the recommendation system.

2. Track mining model

We use the variant of RNN, the gated Recurrent unit neural network (GRU) to model the tourist trajectory make full use of the semantics between the attractions, use multiple trajectories to train at the same time, and train the same minibatch. The attractions is added to the set of negative sample, and the largest negative sample is selected. Each negative sample is weighted to update the loss function compared with the largest negative sample. Finally, the model provides tourists with recommendations for tourist attractions.

Our dataset uses the Shanghai Tourism Dataset and Guilin Tourism Dataset obtained on Ctrip.com. We used Recall Rate@10 and MRR@10 (Mean Reciprocal Rank) as the criteria for judging. Compared with the baseline of the recommendation system, the experimental results show that our method is recommended more accurate.

First, organize the data into a tourist trajectory <v, t, a> means visitor v, time t, visit attraction a. Visitor v1 visits attraction a1 at time t1 and can get data <v1, t1, a1>. After that, visitor v1 browses the attraction a2 at time t2 to get <v1, t2, a2>. The trajectory of the tour is a sequence of all the attractions of the v1 tour sorted by time.

For example, S: <v1, t1, a1>, ..., <v1, tk, ak>, ..., <v1, tL, aL>, where S represents the tourist trajectory of the visitor v1, and L represents the length of the tourist trajectory.

The training network of MAX-GRU is the Gated Recurrent Unit Neural Networks. The input of Gated Recurrent Unit Neural Networks is the first attraction of the tourist trajectory. The output of the model is the predicted score of all attractions, that is, the probability of access. In the feature learning layer, one-hot is embedded as a low-dimensional vector as an input to the GRU. Get the score for each attraction after Gated Recurrent Unit Neural Networks, which is the probability of predicting the next attraction.
Figure 1. The network model flow chart of MAX-GRU

Attractions feature learning layer: All attractions use the one-hot representation of the code, embedded in this layer is represented by a low-dimensional vector.

Gated Recurrent Unit Neural Networks: The input is one-hot representation of the attractions, and the score obtained by the output of the scenic spot score output layer also represents the predicted access probability.

Attractions score output layer: The output of the GRU network is expressed by the softmax activation function to indicate the score of each attractions, that is the predicted access probability.

3. The Gated Recurrent Unit Neural Networks

RNN is a neural network that has recently achieved good results in sequence modeling in the recommended area. Feature extraction of variable length tourist trajectories can be performed in tourist recommendations. The difference between RNN and feedforward neural network is that there are hidden units inside RNN, which can retain the previous information. RNN update hidden state application formula (1):

\[ h^t = \tanh(Wh^{t-1} + Ux^t) \] (1)

\( h^t \) is the hidden state output at time \( t \), \( h^{t-1} \) is the hidden state output of the previous moment, \( x^t \) is the hidden layer input at time \( t \), \( W \) and \( U \) are weight matrix.

GRU is a variant of RNN that alleviates the problem of gradient disappearance of RNN. The structure diagram is Figure 2. The principle of GRU is similar to that of LSTM, that is, two gates control the input information of the previous time step and the current time step, and make predictions at the current time step. The parameter update expression of the GRU model is as follows:

\[ z^t = \sigma(x^tU^z + h^{t-1}W^z) \] (2)

\[ r^t = \sigma(x^tU^r + h^{t-1}W^r) \] (3)

\[ \hat{h}^t = \tanh(x^tU^h + (h^{t-1} \odot r^t)) \] (4)
\[ h^t = (1 - z^t) \odot h^{t-1} + z^t \odot \tilde{h}^t \]  \hspace{1cm} (5) \]

\( z^t \) is the output of the reset gate at time \( t \), \( x^t \) is the input at time \( t \), \( h^{t-1} \) is the hidden state of the previous moment, \( U^z \) and \( W^z \) are the weight matrix of the update gate, \( \tilde{h}^t \) is a temporary hidden state at time \( t \), \( U^h \) is the weight matrix of the current time, \( h^t \) is the final hidden state at time \( t \), \( \sigma \) represents the sigmoid function, \( \odot \) represents hadamard multiplication.

Figure 2. Framework of the GRU

3.1. Model training method

RNN's method of using mini-batch is usually to use a sliding window on the words in the sentence, and put the words in the window together using the mini-batch method, but this method is not suitable for travel. So this article draws on the idea of mini-batch and uses multiple tourist trajectory to train at the same time. When multiple tourist trajectory is simultaneously calculated, the same location in each trajectory becomes the same mini-batch, and other attractions in the same mini-batch are added to the negative example set to improve the training efficiency and reduce the calculation amount. For example, \( i_1,1, i_2, 1, i_3, 1, i_4, 1 \) in Fig.3 are the same mini-batch, \( i_1, 2, i_2, 2, i_3, 2, i_4, 2 \), which are the same mini-batch.

Simultaneous training of multiple tourist trajectories first creates a window, for example, the window size is 4, that is, 4 tourist trajectories are simultaneously trained. Sort each travel trajectory by using the first attractions in the first tourist trajectory as the first input and the second attractions as the second input until the end of a tourist trajectory. After the end of a tourist trajectory, new tourist trajectories will continue to be entered. As shown in Figure 3, both the tourist trajectory 2 and the tourist trajectory 4 are three scenic spots. After the training is completed, there are tourism trajectories 5 and tourist trajectories 6 and then input for training. In order to ensure that the model trains a single trajectory, in a tourist trajectory the weight matrix is reset after the training is completed.
3.2. Select the method of the negative sample

When the dataset sample is large, only the target sample is a positive sample, and the remaining samples are negative samples. The calculation of the scores of all negative samples and the calculation time are large. What we need is a negative sample (effective negative sample) with a score greater than the positive sample. If the positive sample score is greater than the negative sample, then the positive sample is already in front of the recommended list. Such a negative sample is called an invalid negative sample. The negative sample of MAX-GRU uses a random sample of negative samples and other attractions in the same mini-batch as a negative sample. The negative sample with the highest score (max negative) is found in all the negative sample, and then the scores of all other negative and max negative sample are calculated to calculate the weight. The higher the weight of the negative sample, the greater the weight, and the greater the impact on the loss function. The method of random extraction is to extract the number of occurrences of all the attractions, and the more frequently the spots appear, the higher the probability of attractions. Such attractions are also popular attractions, and generally speaking, the negative sample of higher scores. Other attractions in the same mini-batch are generally popular, as the likelihood of an attractions appearing in other itineraries of the same batch is directly proportional to the number of visits to that attractions. We propose a method for constructing negative sample: The tourist attractions in the same mini-batch in other tourist tracks are set as the negative sample set A, and all the attractions are taken out in a random sampling manner as an additional negative sample set B. Because the negative sample set A is the same mini-batch attraction, in general, the attractions in the same mini-batch in front of the tourist trajectory are popular attractions, because tourists will choose to browse according to the popularity of the attractions. The negative sample set B accumulates the number of occurrences of all attractions in the tourist database. For example, Seven Star Park has appeared 100 times in the Guilin Tourism Database, and Xiangbi Mountain has appeared 200 times in the Guilin Tourism Database. Yinziyan is in the Guilin Tourism Database. There are 50 occurrences in total, and the number of occurrences of all the attractions is accumulated. The total number of occurrences is 100+200+50=350, which is randomly selected from 350 scenic spots. Such sampling methods are based on the number of occurrences of attractions, and the more likely the more popular attractions are selected. All the attractions of the final set A and the set B are the new negative sample. Although the collections of the collection A and the collection B are different, it can be considered that most of the selected attractions are popular attractions.

3.3. Loss function and pairwise

The recommendation system does not need to recommend the entire recommendation list to the user. We only need to recommend the first few items of the recommendation list to the user. In the travel recommendation, it is not important that the attractions that the tourists do not like are listed behind the list, because there is no need to recommend them to the tourists, and it is important that the
attractions in front of the list are correctly arranged such as top-5, top-10, top-20[14][15]. There are generally three types of methods for sorting, namely Pointwise, Pairwise, and Listwise. Pointwise calculates the scores and rankings of independent attractions, somehow defining the loss function. Pairwise compares the scores and rankings of positive and negative sample. The positive ranking should be in front of the negative ranking. Listwise uses the scores and rankings of all the attractions, and then compares them to get a new sort, often Listwise effect is relatively good, but its calculation is very large, so it is not used often. This method applies pairwise methods. The final loss function (6) is as follows:

$$L = \frac{1}{S_n} \sum_{q=1}^{S_n} S_m \sigma(r_q - r_p) + \sigma(r_q^2)$$

(6)

Where \(S_n\) represents the sample size, \(S_m\) represents the score ratio of the negative sample \(q\) to the negative sample with the highest score, \(p\) denotes a positive sample, \(q\) denotes a negative sample, and \(r\) denotes a score of an attraction, \(r_q\) is the score indicating the attraction \(q\), \(r_p\) is the score of the attraction \(p\). The loss function is divided into two parts. The first part compares the scores of the positive and negative sample, and the second part is the regularization part. The purpose is to prevent overfitting and to make the negative sample’s score close to zero.

4. Experimental results and comparison

The following is an assessment of the performance of MAX-GRU on the Guilin Tourism Dataset and the Shanghai Tourism Dataset, and compares it with the baselines recently used in the Recurrent Neural Network Recommendation Paper. Visitor ids, visit times and play attractions in Guilin and Shanghai travel data collections are obtained from user reviews in the Ctrip travel web tag. Get all the visitor ids and comment time under the comments of all the attractions, organize them into tourist data, and then organize the travel data of each visitor into the tourist trajectory according to the comment time. The comment time is related to the visit time of the attraction, so the comment time is approximated as the visit time of the attraction. In the experiment, it is considered that the tourist trajectory is too short or the time interval between the attraction and the attraction is too long, so the information is less than 2, and the tourist trajectory with the length of less than 2 is filtered. The set is divided into training sets and test sets at a certain time. In the end, the Guilin tourism data set consisted of 19,724 tourist data, including 290 tourist attractions and 3,940 tourists. The Shanghai Tourism Data Set consists of 113,103 tourist data, including 3,097 tourist attractions and 31,308 visitors.

The recommendation system only needs to recommend the attractions listed in front of the list to the visitors. Therefore, the evaluation index uses the standard recall rate (Recall@10) and Mean Reciprocal Rank (MRR@10) used in the circulation network recommendation system in recent years. The main function of the recall rate is to verify the accuracy of the recommended method. The main function of the MRR is to evaluate the quality of the recommended method.

In this paper, the first indicator in the experiment, Recall@10, is the proportion of the top 10 points in the forecast score list. Recall@10 is represented by the formula (7):

$$\text{Recall@10} = \frac{\sum_{m=1}^{M} z_{m@10}}{M}$$

(7)

\(z_{m@10}\) indicates that the correct attraction \(m\) is in the top 10 of the list of predicted scores arranged from largest to smallest, \(z_{m@10}=1\), otherwise \(z_{m@10}=0\). \(M\) represents the total number of predictions. Recall does not consider the actual ranking of the attraction, it only needs to be in the top 10 of the predicted score list. The second indicator in the experiment, MRR@10, is the reciprocal of the sorting result of the correct attraction in the predicted score list, and then summed and averaged. MRR@10 is represented by the formula (8):

$$\text{MRR@10} = \frac{\sum_{m=1}^{M} \frac{1}{\text{rank}_m@10}}{M}$$

(8)

\(\text{rank}_m@10\) indicates that the correct attraction \(m\) is in the top 10 of the list of predicted scores
arranged from large to small, $\text{rank}_{m}@10 = a$, if the correct attraction ranks in the predicted score list after 10, $\frac{1}{\text{rank}_{m}@10} = 0$. $a$ indicates the specific number of rankings of the correct attraction $m$ in the predicted score list. $M$ indicates the total number of predictions.

4.1. Baseline model

Compare the MAX-GRU and baseline models on the Guilin Tourism Dataset and the Shanghai Tourism Dataset. The five baseline models commonly used in the loop network recommendation system in recent years are used in the experiment, which are described below.

RandomPred: Random prediction, random recommendation of attractions to tourists. This is the easiest way to recommend it.

POP: This baseline is always recommended for visitors to the most popular attractions in the training set. Although it is simple, it is a good baseline model in some recommended areas.

Item-KNN: This baseline recommends attractions that are similar to the correct attraction to visitors. This baseline is one of the most common solutions for “attractions-attractions” in the recommendation system. It makes recommendations in the context of “what attractions have been visited by other people in the area.” [16][17]

BPR-MF: This method is one of the most common matrix decomposition methods. [18] It optimizes pairwise rankings by the Stochastic Gradient Descent Algorithm (SGD). Matrix decomposition cannot be directly applied to session-based recommendations because new sessions have no pre-computed feature vectors. This problem can be overcome by averaging the similarity of the feature vectors between the recommended attraction and the conversational attraction.

GRU: This method uses the GRU neural network to train the tourist trajectory, and the negative sample is selected from other attractions in the mini-batch.

4.2. experiment

The size of the window and the number of hidden units that are simultaneously trained in the experiment will affect the experimental results and training time. The more hidden units, the longer the training time. The more training windows and the more trajectories of training, the less training time the model has. We hidden units the experimental parameters and hope to set it to 50-900, and the training window size is set to 20-60. The test results are given in table 1, table 2, table 3, table 4.

| Window layers | 20   | 30   | 40   | 50   | 60   |
|---------------|------|------|------|------|------|
| 50            | 0.3341744306 | 0.328998447 | 0.331068840 | 0.329257246 | 0.325116459 |
| 100           | 0.362383540  | 0.359213250 | 0.359860248 | 0.360636645 | 0.36154244 |
| 200           | 0.386451863  | 0.392339544 | 0.381728778 | 0.382375776 | 0.382375776 |
| 300           | 0.396739130  | 0.397256728 | 0.395509834 | 0.392598343 | 0.392339544 |
| 400           | 0.4072204968 | 0.406767598 | 0.410714285 | 0.408514492 | 0.401074016 |
| 500           | 0.4193840579 | 0.415631469 | 0.416149068 | 0.410973084 | 0.408967391 |
| 600           | 0.4170548654 | 0.418672360 | 0.415760869 | 0.414790372 | 0.410649585 |
| 700           | 0.423654244  | 0.424689440 | 0.421777950 | 0.421389751 | 0.419254658 |
| 800           | 0.4219720496 | 0.429218426 | 0.423783643 | 0.421519151 | 0.421195652 |
| 900           | 0.4289596273 | 0.428053830 | 0.427924430 | 0.42656734 | 0.423718944 |
This experiment uses a single-layer GRU network. On the two data sets, our model can see that the effect of the model is getting better and better with the increase of layers, but it is not reflected in the method of using only the GRU network without adding the max negative example. We propose that the improvement based on the max negative example will normalize the model, and at the same time prove the excellence of our proposed method. On the two data sets, it can be seen that when the hidden layer unit is set to 900 and the travel window size is set to 20, the values of RECALL@10 and
MRR@10 are almost the best, so when compared with other models. Select this set of data. The effect of the model being improved with the increase of layers in the Shanghai dataset is more obvious because the Shanghai tourism dataset is more comprehensive than the Guilin tourism dataset, so the recommended effect on the model is better.

4.3. Model comparison experiment

In this experiment, the hidden unit is selected to be 900, the window size is selected as 20, tanh is used as the hidden layer activation function, and softmax is used as the activation function of the output layer.

| Baseline model | Recall@10 | MRR@10 |
|----------------|-----------|--------|
| RandomPred     | 0.03614457| 0.01009753 |
| POP            | 0.22891566| 0.05998278 |
| ItemKNN        | 0.24698795| 0.10447265 |
| BPR            | 0.30722891| 0.11882052 |
| GRU            | 0.39759036| 0.11539730 |
| MAX-GRU        | 0.42895962| 0.18716926 |

| Baseline model | Recall@10 | MRR@10 |
|----------------|-----------|--------|
| RandomPred     | 0.00347680| 0.00127563 |
| POP            | 0.04423929| 0.02400715 |
| ItemKNN        | 0.06689841| 0.03075094 |
| BPR            | 0.06941619| 0.02536318 |
| GRU            | 0.13535547| 0.06218345 |
| MAX-GRU        | 0.30084456| 0.13533729 |

It can be seen from Table 5 that our improvement on the original model (GRU) is effective. Recall@10 is 7.8% higher than the original model on the Guilin tourism dataset, and 62% higher on the MRR@10 than the original model and the experimental effect is better than the BPR model. It can be seen from Table 5 that Recall@10 and MRR@10 have greatly improved on the original model in the Shanghai tourism dataset. Recall@10 has increased by 122% compared with the original model. MRR@10 and the original Compared with the model, the model has increased by 117%. Both experiments are sufficient to prove the excellence of our model. The improvement of the Shanghai tourism dataset is because the data diversity and rich performance of the Shanghai tourism dataset reflect our model. The rationality and effectiveness of selecting effective negative examples.

Compared with the simple use of the GRU network, we added a negative sample set and selected a comparison with the max negative sample to significantly improve the recommendation accuracy. It also illustrates the importance of selecting effective negative sample.

5. Conclusion

In this paper, we propose a tourist trajectory that uses the gru network to model, by finding the most negative example of the score, the negative sample with the highest score (max negative) is found in all the negative sample, and then the scores of all other negative and max negative sample are calculated to calculate the weight. Improve recommendation accuracy by adding valid negative sample. The experimental results show that compared with other baseline methods, the GRU neural network is used to model the sequence information mined in the tourist trajectory, which can achieve better results and can make more accurate recommendations for tourists. The next step is to consider the location of the attraction and extract information from the picture to consider the accuracy of the recommendation for the tourist sequence.
References

[1] Kolstad A, Özgöbek Ö, Gulla J A, et al. Content-Based Recommendations for Sustainable Wardrobes Using Linked Open Data[J]. Mobile Networks and Applications, 2018, 23(6): 1727-1734.

[2] Wang C, Agrawal A, Li X, et al. Content-based top-N recommendations with perceived similarity[C]//2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2017: 1052-1057.

[3] Cui G, Luo J, Wang X. Personalized travel route recommendation using collaborative filtering based on GPS trajectories[J]. International journal of digital earth, 2018, 11(3): 284-307.

[4] Fenza G, Fischetti E, Furno D, et al. A hybrid context aware system for tourist guidance based on collaborative filtering[C]//Fuzzy Systems (FUZZ), 2011 IEEE International Conference on. Taiwan: IEEE,2011:131-138.

[5] Moreno A, Valls A, Isern D, et al. Sigtur/e-destination: ontology-based personalized recommendation of tourism and leisure activities[J]. Engineering Applications of Artificial Intelligence,2013,26(1):633-651.

[6] Lu E H C, Fang S H, Tseng V S.Integrating tourist packages and tourist attractions for personalized trip planning based on travel constraints[J].GeoInformatica,2016,20(4):741-763.

[7] FU Y P, QIU Y H. Method of Personalized Collaboration Filter Recommendation Based on Bayesian Network. Computer Science,2016,43(9):266-268. (in Chinese)

[8] Jiang S, Qian X, Shen J, et al. Author topic model-based collaborative filtering for personalized POI recommendations[J]. IEEE transactions on multimedia,2015,17(6):907-918.

[9] Cui G, Luo J, Wang X. Personalized travel route recommendation using collaborative filtering based on GPS trajectories[J]. International journal of digital earth, 2018, 11(3): 284-307.

[10] Hidasi B, Quadrana M, Karatzoglou A, et al. Parallel recurrent neural network architectures for feature-rich session-based recommendations[C]//Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016: 241-248.

[11] Bogina V, Kuflik T. Incorporating Dwell Time in Session-Based Recommendations with Recurrent Neural Networks[C]//RecTemp@RecSys. 2017: 57-59.

[12] Ludewig M, Jannach D. Evaluation of session-based recommendation algorithms[J]. User Modeling and User-Adapted Interaction, 2018, 28(4-5): 331-390.

[13] Hidasi B, Karatzoglou A. Recurrent neural networks with top-k gains for session-based recommendations[C]//Proceedings of the 27th ACM International Conference on Information and Knowledge Management. ACM, 2018: 843-852.

[14] Greenstein-Messica A, Rokach L, Friedman M. Session-based recommendations using item embedding[C]//Proceedings of the 22nd International Conference on Intelligent User Interfaces. ACM, 2017: 629-633.

[15] Devooght R, Bersini H. Long and short-term recommendations with recurrent neural networks[C]//Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization. ACM, 2017: 13-21.

[16] Davidson J, Liebald B, Liu J, et al. The YouTube video recommendation system[C]//Proceedings of the 4th ACM conference on Recommender systems. Barcelona:ACM,2010:293-296

[17] Linden G, Smith B, York J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering[J]. IEEE Internet Computing, 2003, 7(1):76-80.

[18] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[C]//Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, 2009: 452-461.