MatchRec: A User-item Matching Method for Sequential Recommendation

Lv Luo\textsuperscript{1,a}, Tiankai Chen\textsuperscript{1,b} and Rong Qiu\textsuperscript{1,r,c}

\textsuperscript{1}School of Computer Science and Engineering, Central South University, Changsha 410083, China

\textsuperscript{a}Email:luolv77@qq.com; \textsuperscript{b}1046732647@qq.com; \textsuperscript{r,c}qiurongrong@126.com

Abstract. Sequential recommendation is based on the sequence of items that users interact with to predict the items that users will interact with next time. The existing methods usually consider the hidden sequential patterns in the user’s action sequence, but modeling candidate item features is ignored. To solve this problem, in this paper, we use the two-tower network structure to model the user’s action sequence and candidate item features respectively, then recommend items to users according to the matching score of user interest and item attribute. Specifically, we use self-attention mechanism to mine the information in the user’s action sequence, and use Factorization machines to automatically construct the cross features of item. We named this method MatchRec. The experiments on the public datasets demonstrated that MatchRec has a significant improvement in recommendation accuracy compared with that of various baseline methods.

1. Introduction
With the rapid development of the Internet, there is a huge amount of information, music, movies, and goods on the Internet. People surfing or shopping online are facing many choices. Therefore, good recommendation technology is very important for music sites, shopping sites, video sites, and so on. Recommendation technology usually uses the user's historical interaction records to infer items that the user may like. User’s historical interaction records often show a certain sequential relationship. For example, if a user purchases a mobile phone, then he is likely to purchase mobile phone accessories next. The user's interaction behavior on the website can be organized into a series of interaction records in chronological order. Sequential recommendation is to use various methods to mine the patterns and information in the user interaction sequence in an attempt to characterize the user's interest and predict the next possible interaction items.

Classic recommendation methods, such as Collaborative Filtering [1] and Matrix Factorization [2], have been widely used in the industry, but they all ignore the sequential patterns in the user-item interaction sequence. In [3], the author uses Markov chain to model state transition relationships in user-item interaction sequence. In the fields of natural language processing and computer vision, RNN, CNN, and self-attention technologies have been proven to be high effective. In recent years, many researchers have used these techniques in the field of sequence recommendation and achieved good results. RNN is a neural network that recursively processes sequence data. It recursively determines the new state based on the historical state and the current input information, so as to capture the relationship between the current data and the historical data. CNN uses a certain size sliding window to extract information between adjacent time steps in the sequence data. self-attention uses the
correlation between the data at each time step and the overall data to re-encode the sequence data. Combining with position encoding can effectively extract the global information in the sequence data. In [4], RNN was used to model the user's session data. In [5], CNN was used to extract the information in the user's interaction with the item. In [6], the self-attention mechanism was used to model user sequence data to predict items that the user will interact with next time. In [7], the author argue that only the item-level sequences cannot reveal the full sequential patterns, while explicit and implicit feature-level sequences can help extract the full sequential patterns.

In all of the research results we know, researchers use RNN, CNN, self-attention and other methods to model the user's action sequence, including the item sequence and item features sequence. It is capturing user interests, lacking in modeling the rich features of candidate items, and matching user interests and item attributes.

There are many different items on the website. An item often has many features. For example, a piece of clothing on an e-commerce website may have features such as brand, category, style, store, brand, and so on. Item features are usually numerous and complex, and the importance of different features to users is different. It is important to properly characterize the attributes of an item by making effective use of its features.

From the user information and the user 's historical interaction behavior, we can have a more accurate description of the user 's interests. For example, in shopping websites, a user often buy goods with lower prices, which can explain that users are more sensitive to prices and are more inclined to choose good goods at a rock-bottom price. If we can effectively use the features of the candidate items, we will know which items are cheap, then it is a good choice to recommend this cheap candidate items to this user.

Because the interest of users is diverse, it is not reasonable to use the same vector to represent user interests in the process of matching user interests with a large number of item attributes. The diversity of user interests is reflected in different preferences for item categories, item prices, item brands, etc. That is, the user's interests include multiple topics. When dealing with user’s action sequence, it is important to have different representations of user interests based on different topics.

Based on the problems described above, we did the following:
A: We propose a novel framework for sequential recommendation task. Our method uses the two-tower structure to model user’s action sequence and candidate item features, uses Factorization machines to automatically construct cross features of items.
B: Because user interest is diverse, we propose a multi-topic interest method.
C: We compare the performance of our method and baseline methods on a public dataset.

2. Preliminaries

2.1. Attention mechanism
From daily experience, we can know that humans will try to quickly extract high-value information from a large amount of information by focusing on the most interesting parts in the field of vision. In [8], two attention mechanisms of hard attention and soft attention were proposed. In [9], the soft attention method was used to improve the effect of the Seq2Seq structure in the translation task, so that when translating different positions of the sentence, the model could focus on more relevant information in the original sentence. RNN combined with attention mechanism has achieved good results in sequence modeling, but RNN cannot be parallelized, which makes the model training time too long. In [10], a new network structure called transformer was proposed, and the attention method used in this structure is called self-attention mechanism. In this paper, the self-attention mechanism combined with the position embedding has achieved promising empirical results in machine translation task, and is convenient for parallelization.

2.2. Factorization machines(FM)
In the recommendation system and the click-through rate (CTR) prediction, it is very important to
construct cross features. In order to construct cross features automatically and efficiently, in [11], the author uses the inner product of two feature vectors to represent the weight of second-order cross features. The weights of all second-order cross features are calculated on the premise of linear time complexity.

3. The proposed model: MatchRec

3.1. Problem description
Let \( U = \{u_1, u_2, ..., u_{|U|}\} \) be a set of users and \( I = \{i_1, i_2, ..., i_{|I|}\} \) be a set of items. We use \( S = (s_1, s_2, ..., s_{|s|}) \) to denote a sequence of items in chronological order that a user has interacted with before, where \( s_i \in I \). Each item has some features, such as the price, category and brand. For example, let \( b_i \in B \) be the category of item \( i \), where \( B \) denote the set of brands. Let \( S' = (s'_1, s'_2, ..., s'_{|s'|}) \) be a sequence of item features in \( S \). Each element in \( S' \) represents a set containing all features of corresponding item. The task of sequential recommendation is to predict the item that the user will interact with next time based on the existing user-item interaction sequence. During the training process, the model predicts the item at time step \( t \), depending on the previous \( t - 1 \) items. In this process, the order relationship of the items in the user interaction sequence is used to capture the sequence pattern in the user interaction sequence. In the existing NN-based sequence recommendation methods, sequence modeling techniques such as RNN, CNN, and Self-attention are used to complete the modeling of user interaction sequences. A vector representing the current user's interest can be obtained, the vector is used as the input of DNN, and finally the softmax function is used to obtain the probability that all items will become the user's next interactive item. In the recommendation field, in addition to accurately capturing the user's interest, it is also very important to characterize the attributes of the item. Therefore, our method effectively models the user's action sequence and the original features of the candidate items, and uses the method of calculating the inner product of vectors, get the match score of user interest and item attributes, and make recommendations for users based on the match score. Our method is described in detail below. Matchrec's network structure is shown in Figure 1.

3.2. Embedding Layer
Because the length of the user's action sequence is not fixed, We need to transform sequences \( S = (s_1, s_2, ..., s_{|s|}) \) and \( S' = (s'_1, s'_2, ..., s'_{|s'|}) \) into sequences of fixed length \( n \), where \( n \) represents the maximum length that our model can handle. If the length of the user's action sequence is less than \( n \), then fill it with specific symbols at the beginning of the sequence. If the length of the sequence is greater than \( n \), then remove the corresponding length element before the sequence. OneHot encoding is usually used to process discrete value variables, but because of the large number of items, the dimension of the item encoding after OneHot encoding is very large, so we need to have an Embedding layer to reduce the dimension of OneHot encoding. By using the Embedding layer, the OneHot encoding of the items in the sequence is correspondingly converted into a dense vector of dimension \( d \). Then, we transformed the item sequence \( S \) into an item vector sequence \( E^s \in \mathbb{R}^{n \times d} \). For the item feature sequence \( S' \), we used the same processing method to obtain the item feature vector sequence \( E' \in \mathbb{R}^{n \times h \times d} \), where \( h \) denote the number of item features.

We use the self-attention mechanism to mine the information in these two sequences separately. Below we introduce the processing method of the item features sequence.
3.3. CrossFeatures Layer
In the recommendation system or click-through rate (CTR) prediction, it is very important to construct cross features, so we add cross features based on the original item features. We use FM method to calculate the vector point multiplication of pairwise features as the second-order cross feature, and add all the second-order feature vectors and all the original feature vectors to obtain the item feature vector containing the cross feature. The CrossFeature Layer is defined as:

$$CrossFeature(V) = \left( \sum_{i} v_{i} \right)^{2} - \sum_{i} v_{i}^{2} + \sum_{i} v_{i},$$

(1)

Where $V$ represents the feature vector matrix of an item.

Using the feature vectors of each time step in $E^{t}$ as the input of the Cross-Features Layer, we get cross feature sequence:

$$E^{cf^{t}} = \{CrossFeature(e_{1}^{t}), CrossFeature(e_{2}^{t}), ..., CrossFeature(e_{n}^{t})\}.$$  

(2)

3.4. Position Embedding
Because in sequential recommendation, the order relationship between items in the sequence is very important, but if self-attention is used to model the user's action sequence, the order relationship in the sequence cannot be directly modeled, so we need to use position embedding. We use matrix $P \in R^{n \times d}$ as the position embedding matrix, let the item cross feature vector sequence $E^{cf}$ add the corresponding position embedding,
3.5. Self-attention Layer

In natural language processing (NLP), the attention mechanism is widely used because it can make the model pay more attention to important information, so the attention mechanism helps researchers make good progress in machine translation, text classification and other tasks. Attention methods usually assign different weights to value based on the relevance of the query and key. Usually, the key and value are usually the same, but the query is different. In the self-attention method proposed by [10], query, key, and value are all obtained by linear transformation of the original matrix. The self-attention layer can effectively capture the global information in the sequence.

We perform different linear transformations on $E^f$ to obtain the query, key, and value matrices.

$$Q = E^f W^Q,$$

$$K = E^f W^K,$$

$$V = E^f W^V,$$

where, $W^Q, W^K, W^V \in \mathbb{R}^{d \times d}$. Then, the similarity matrix is calculated according to $Q$ and $K$, and the similarity matrix is used to multiply the $V$ matrix to obtain the re-encoded sequence,

$$A^f = \text{Softmax} \left( \frac{QK^T}{\sqrt{d}} \right) V .$$

To alleviate the problem of gradient disappearance and gradient explosion, we use residual connection and LayerNorm[12] here,

$$A^f = \text{LayerNorm}(A^f + E^f) .$$

We use the vector of each time step of $A^f$ as the input of the forward neural network to mine non-linear information. Similarly, residual connection and LayerNorm are also used here. For $a^f_t$, there is

$$z^f_t = \text{LayerNorm}(W a^f_t + b + a^f_t),$$

where $W \in \mathbb{R}^{d \times d}$, So we get $Z^f$ from $A^f$.

Above, we described how to mine the information in the item feature sequence of the user behavior sequence. The mining of the item sequence information is similar to the processing of the feature sequence, but without the feature cross-feature layer, it will not be described. After processing the item sequence, we can finally get $Z^i$.

We splice $Z^f$ and $Z^i$ to get the user interest matrix,

$$Z = \begin{bmatrix}
    z^f_1 ; z^i_1 \\
    z^f_2 ; z^i_2 \\
    \vdots \\
    z^f_n ; z^i_n
\end{bmatrix} .$$

We use the vector of the last time step of $Z$ as the user's short-term interest vector representation.

3.6. Match Layer

For a candidate item $i$, after its features pass through the embedding layer, its feature embedding is
represented by \( e_i^f = (e_{i1}, e_{i2}, ..., e_{ik}) \). In order to calculate cross features, we use the original feature vector of candidate item \( i \) as the input of the CrossFeature layer to get its cross feature vector \( e_i^c = \text{CrossFeature}(e_{i}) \). Then the item's Embedding vector is concated with the cross features vector to get the attribute vector of item \( i \).

If only considering short-term interests of users, we use the inner product of the user's short-term interest vector and the candidate item attribute vector as the matching score,

\[
y_{u,i} = z_n \odot p_i. \tag{11}
\]

### 3.7. Match Layer with Multi-topic Interest

In addition to short-term interests of users, long-term interests of users are also important. However, a user's long-term interest involves all aspects. For example, a user will be interested in electronic products, novel books, and clothes at the same time, so it is difficult for a user's long-term interest to be represented by a single vector. We propose a multi-topic interest method to model the long-term interests of users. We use a matrix \( M^p \in \mathbb{R}^{c \times d} \) (\( c \) is a hyperparameter, representing the number of topics) as the query, \( Z \) as the key and value, and use the attention method to calculate user's multi-topic interest matrix based on the correlation between the vectors in \( Z \) and \( M^p \). The specific formula is:

\[
O = \text{Softmax}(M^pZ^T)Z, \tag{12}
\]

where, the \( M^p \) matrix is obtained by random initialization and participates in model training.

By adding the user's multi-topic interest matrix and the user's short-term interests, we can obtain a comprehensive representation of user interests,

\[
O = \begin{bmatrix}
o_1 + z_n \\
o_2 + z_n \\
... \\
o_c + z_n
\end{bmatrix}. \tag{13}
\]

By performing matrix multiplication on the user interest matrix \( O \) and the item attribute vector \( p_i \), we get \( c \) matching scores. The weight vector of \( c \) matching scores is obtained according to the correlation between the candidate item attribute vector and the \( M^p \) matrix.

\[
\alpha = \text{Softmax}(\text{relu}(WM^p + b)p_i), \tag{14}
\]

\[
y_{u,i} = \alpha(Op_i). \tag{15}
\]

### 3.8. Optimization

During training, we sample one hundred negative samples for each positive sample to form one hundred negative samples. We use the Softmax function to obtain the probability of each item for the output of the positive samples and negative samples, and then minimize the log loss.

\[
\text{Loss} = -\log \left( \frac{\exp(y_{u,\text{pos}})}{\exp(y_{u,\text{pos}}) + \sum_{i\in \text{negs}} \exp(y_{u,i})} \right) \tag{16}
\]

Where \( y_{u,\text{pos}} \) denotes the output value of the positive sample and \( \text{negs} \) denotes the negative item set. Finally, we use the gradient descent method to adjust the weights in the network and minimize the loss function.
4. Experiment

We evaluate the proposed method against popular baselines on a public dataset Tmall. Tmall, the largest B2C platform in China, is a user-purchase data obtained from IJCAI 2015 competition. The dataset contains users’ action records of clicking, purchasing and collecting items on Tmall. We only use the purchasing records and remove users with less than 15 interactive items and items with less than 5 interactive users.

In order to evaluate the performance of each method on the dataset, we select the widely used evaluation metric Recall@k that is the proportion of cases having the desired item among the top-k items in all test cases. In our experiments, we choose 10 and 20 as the value of k.

We use the following baseline methods to compare with our method:

- **PopRec**: Sorts items according to their popularity, that is, recommends the items that appear most often to users.
- **Item-KNN**: An item-based collaborative filtering method that recommends similar items to items that the user has clicked on to the user. The degree of direct similarity between items and items is measured by the number of co-occurrences in the same user interaction item set. This method is a classic recommendation algorithm.
- **BPR**: This method is also one of the classic algorithms using implicit feedback for recommendation. It establishes the LOSS function between the user and the item, optimizes it through the SGD algorithm, and then predicts the user's preference for the item.
- **GRU4REC**: Use GRU to model sequences for session recommendation.
- **SASREC**: For the sequence recommendation, the self-attention mechanism is used to model the sequence.
- **FDSA**: Use various features of items in user's action sequence for sequential recommendation.

On the dataset, we tested the Recall@k metric of our method and baseline methods, as shown in Table 1.

### Table 1. Experiment result of the proposed method and baseline methods. MatchRec denotes the MatchRec model using the multi-topic interest method.

| METHODS   | Recall@10 | Recall@20 |
|-----------|-----------|-----------|
| PopRec    | 0.0012    | 0.0025    |
| Item-KNN  | 0.2358    | 0.2584    |
| BPR       | 0.2360    | 0.2571    |
| GRU4REC   | 0.2397    | 0.2601    |
| SASREC    | 0.2402    | 0.2614    |
| FDSA      | 0.2636    | 0.2862    |
| MatchRec  | **0.2741** | **0.2952** |
| MatchRec\textsuperscript{ext} | **0.2825** | **0.3049** |

Item-KNN and BPR methods are better than PopRec, which shows that the classic recommendation method is still very good. GRU4REC is better than Item-KNN and BPR methods, which indicates that the modeling of sequence patterns in user interaction sequences is conducive to improving the accuracy of recommendation. The SASREC method is better than the GRU4REC method, indicating that the self-attention mechanism is better than the GRU neural unit in modeling user sequences. Our method has achieved a good improvement on the basis of several baseline methods, which indicates that the matching of users and items is very important in sequence recommendation. It also shows that our method effectively extracts user interests and item attributes. MatchRec\textsuperscript{ext} has been further improved on the MatchRec, which proves that the multi-topic interest method is effective.
5. Conclusion
In this paper, we proposed a novel method named MatchRec for sequential recommendation. MatchRec use the two-tower network structure to model the user’s action sequence and candidate item features respectively, then recommend items to users according to the matching score of user interest and item attribute. Considering the diversity of user interest, we also propose the multi-topic interest method to calculate the matching scores of user interest and item attribute. The experimental results have shown that our model outperforms various baselines.

Acknowledgments
This work is support by the National Natural Science Foundation of Hunan Province (No.2019JJ40363 to RQ).

References
[1] Wei S, Ye N, Zhang S, et al. Item-Based Collaborative Filtering Recommendation Algorithm Combining Item Category with Interestingness Measure[C]// International Conference on Computer Science and Service System. IEEE Computer Society, 2012:2038-2041.
[2] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. Computer, 2009, 42(8):30-37.
[3] Rendle S, Freudenthaler C, Schmidt-Thieme L. [ACM Press the 19th international conference - Raleigh, North Carolina, USA (2010.04.26-2010.04.30)] Proceedings of the 19th international conference on World wide web - WWW ‘10 - Factorizing personalized Markov chains for next-basket recommendation[C]// International Conference on World Wide Web. ACM, 2010:811.
[4] Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-based recommendations with recurrent neural networks[J]. arXiv preprint arXiv:1511.06939, 2015.
[5] Tang J, Wang K. Personalized top-n sequential recommendation via convolutional sequence embedding[C]//Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. ACM, 2018: 565-573.
[6] Kang W C, McAuley J. Self-attentive sequential recommendation[C]//2018 IEEE International Conference on Data Mining (ICDM). IEEE, 2018: 197-206.
[7] Zhang T, Zhao P, Liu Y, et al. Feature-level deeper self-attention network for sequential recommendation[C]//Proceedings of the 28th International Joint Conference on Artificial Intelligence. AAAI Press, 2019: 4320-4326.
[8] Show A. Tell: Neural image caption generation with visual attention[J]. Kelvin Xu et. al.. arXiv Pre-Print, 2015, 23.
[9] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.
[10] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.
[11] Rendle S. Factorization machines[C]//2010 IEEE International Conference on Data Mining. IEEE, 2010: 995-1000.
[12] Ba J L, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016.