

Research on Personalized Recommendation Algorithm of Internet Platform Goods Based on Knowledge Graph

Na Nie*

Party School of the Shanxi Provincial Committee of CPC (Shanxi Institute of Public Administration),
Taiyuan 030000, China

* Corresponding Author Email: 397221521@qq.com

Abstract. Personalized recommendation method is an effective means to filter out the information users need from a large amount of information, which is rich in practical value. Personalized recommendation methods are maturing, and many e-commerce platforms have been using different forms of recommendation methods with great success. In the recommendation systems of large-scale e-commerce platforms, traditional recommendation algorithms represented by collaborative filtering are modeled only based on users' rating data, and sparse user-project interaction data and cold start are two inevitable problems. The introduction of knowledge graphs in recommendation systems can effectively solve these problems because of their rich knowledge content and powerful relationship processing capability. In this paper, we study the personalized recommendation algorithm based on knowledge graph as auxiliary information, and use the temporal information of user-item interaction in the graph to model users' interests over time at a finer granularity, taking into account the problem of high training time cost of the model due to frequent updates of the knowledge graph when recommending to users dynamically. The article proposes the Interactive Knowledge-Aware Attention Network Algorithmic Model for Recommendations (IKANAM) and conducts comparison experiments on public datasets. The results show that the IKANAM recommendation algorithm can effectively improve the effectiveness of personalized recommendation of products on Internet platforms.

Keywords: Collaborative filtering, knowledge graph, deep learning, recommendation algorithm, graph neural network.

1. Introduction

With the explosive growth of data volume on various online platforms, recommendation systems have started to be applied in scenarios such as e-commerce platforms due to information overload. Recommendation algorithms analyze the user's historical behavior to uncover the user's potential needs, achieve group feature matching of the user, and further recommend content that the user may like in order to alleviate the information overload problem. Recommendation methods based on deep learning methods are widely used in the prior art [1]. Neural network models use their own feature extraction and learning functions to determine users' preference information as the basis for recommendation. Early recommendation algorithms only calculated similarity based on users' historical interaction data, and the limited number of items led to sparse information about users' behaviors, resulting in inaccurate recommendation results and cold start problems. Knowledge Graph (KG) contains a large amount of auxiliary information and can provide an effective way to organize and manage a large amount of information. Introducing Knowledge Graph in recommendation algorithms can help to explore deeper potential connections between users and items and enhance the recommendation results [2].

The introduction of knowledge graphs in recommendation algorithms can help to explore deeper potential connections between users and items and enhance the diversity of recommendation results [3]. Combining network representation learning with knowledge graphs and then introducing the extracted semantic information into recommendation algorithms is the basic idea of existing research.

2. Recommendation System Related Technologies

2.1 Recommended System Framework

The recommendation system has a user set U and an item set I . The data in the system includes the interaction records of users and items, user information and item information called scene information C (Context) [4]. We define the problem to be solved by the recommendation algorithm as follows: build a recommendation model $f(U, I, C)$ in the recommendation system, predict the user's interest ratings for all items, then sort the items according to the interest ratings, and select the top K recommended items to form the recommendation list [5]. The logical framework of the recommendation system is shown in Figure 1.

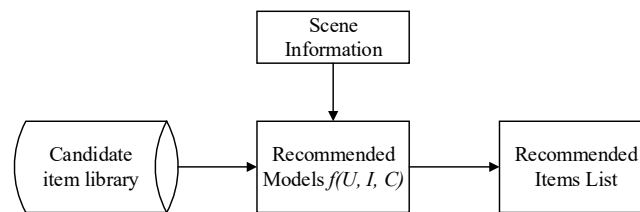


Fig. 1 Logical framework of the recommendation system

2.2 Collaborative Filtering Recommendation Algorithm

The core idea of collaborative filtering recommendation algorithm is to discover the common interests among users through their history records. First, the similarity of the target item (or user) is calculated by using the existing user rating information; then, the set of nearest neighbors of the target item (or user) is found based on the similarity; finally, the rating information of the nearest neighbors is used to predict the rating of the target user-item pair and generate the recommendation result for the target user. Collaborative filtering algorithms are mainly divided into two categories: Neighborhood-based Collaborative Filtering and model-based approaches [6]. Among them, Neighborhood-based algorithms include Item-based CF and User-based CF. Collaborative filtering algorithms have the advantages of user-specific recommendations, do not rely on industry domain information, can personalize recommendations and have a high degree of automation, and improve performance over time. Collaborative filtering algorithms only focus on shallow user-item interactions and tend to favor popular items, resulting in a lack of novelty in recommendations. Collaborative filtering algorithms do not consider item content information and user attribute characteristics, and suffer from data sparsity and cold start problems [7].

2.3 Sequence-based Recommendation Algorithm

Sequential recommendation focuses on understanding the user's decision process and getting more accurate recommendation results by modeling fine-grained user dynamic preferences. The factorizer-based sequential recommendation algorithm uses matrix decomposition to decompose the transfer probability between items into a vector of user and item features, and uses this vector for subsequent recommendation tasks. Markov chain-based sequential recommendation calculates transfer probabilities in two ways: the first way maps Markov chains into Euclidean space and calculates their corresponding transfer probabilities by measuring the distance between items. The second way converts the user history interaction data into a directed graph, where the nodes of the graph represent items, observes the co-occurrence frequency of the target node and the current node in the graph, and considers the frequency as the weight of the edge, and then calculates the transfer probability between nodes according to the weight size [8]. The neural network-based sequence recommendation algorithm treats the historical items contained in the interaction sequence as words, and obtains the feature representation of each item after training by Recurrent Neural Network (RNN). The algorithm captures fine-grained user long- and short-term preferences by modeling the sequence information in

the same session to obtain the association relationship between items and the order dependency between different sessions for the purpose of assisted recommendation.

2.4 Hybrid Recommendation Algorithm

Hybrid recommendation algorithms combine the results of multiple algorithms to achieve the goal of exploiting the advantages of multiple recommendation algorithms and overcoming their disadvantages as much as possible. There are many types of hybrid recommendation algorithms. The weighted hybrid recommendation algorithm gives several different recommendation algorithms different levels of importance, and the final result is obtained by weighting the algorithms [9]. For example, when collaborative filtering does not recommend new items to the system, it can switch to the content-based recommendation algorithm. Combined recommendation algorithm combines the results of multiple recommendation algorithms into a larger recommendation list, which is directly displayed to the user and given to the user to choose.

2.5 Principles of Attentional Mechanisms

The attention mechanism often contains three inputs, Query, Key and Value, and the final weighted sum of multiple Values will be the output of the attention mechanism. The attention mechanism can be summarized as the following two processes: calculating the weighting factor based on Query and Key, and getting the output based on the weighting factor and Value. The attention mechanism can be formalized as Equation (1). where Q, K, and V represent Query, Key, and Value. Respectively, L represents the number of Key and Value, and Similarity (-) represents the weight function.

$$Attention(Q, K, V) = \sum_{i=1}^L Similarity(Q, K_i) * V_i \tag{1}$$

Different weight functions are introduced to calculate the similarity between Query and Key, using inner product, cosine similarity, and multilayer perceptron to calculate the weight coefficients in the attention mechanism [10]. Query and Key, Value should come from different data sources, and the attention mechanism can obtain the dependency between the two data sources to know the weight corresponding to each Value. The self-attentive mechanism differs from the general attention mechanism in that the Query, Key and Value come from the same data source, and the purpose is to obtain the connection between the data of the data sources. The self-attention mechanism can be formalized as Equation (2).

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

Where $Q = W_q X$, $K = W_k X$, $V = W_v X$, $X = \{x_1, x_2, \dots, x_n\}$, X denotes a sentence, n denotes the number of words in the sentence, and x_i denotes the vector of the i th word in the sentence. The attention mechanism is shown in Figure 2.

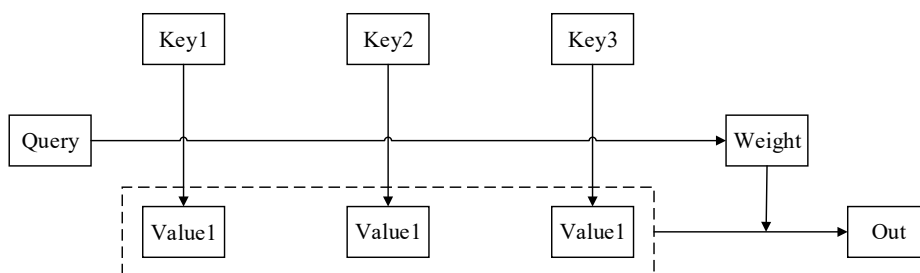


Fig. 2 Descriptive framework of attentional mechanisms

3. Recommendation algorithms based on interactive knowledge-aware attention networks

3.1 Graph Neural Networks

Graph Neural Network (GNN) is a kind of neural network based on graph structure. GNN can compute on the whole graph and generate Embedding feature vector of nodes. GNN is very suitable for handling recommendation task on knowledge graph. GNN is divided into four major networks, Recurrent Graph Neural Networks(Rec GNN), Graph Convolution Neural Networks (GCN), Graph Autoencoders (GAE), Graph Spatial-temporal Networks (STGNN) are classified into four categories [11]. The structure of Graph Attention Network (GAT) is similar to Graph Sage in that it represents the node embedding by aggregating the information of neighboring nodes. GAT distinguishes the importance of information passed by nodes by calculating the attention factor between nodes.

3.2 Recommendation Algorithm Based on Graph Neural Network

The representative algorithms of graph neural network based recommendation algorithms are Pin Sage, Knowledge Graph Convolutional Network (KGCN), Knowledge Graph Attention Network (KGAT). KGCN uses the knowledge graph of attributes of items to embed representations of items to obtain knowledge graph-enhanced item representations, and then combines the user's feature vector to make recommendations. KGCN takes the inner product of the user vector and the item vector enhanced by the knowledge graph as the fractional value of the user's likelihood to interact with the item. KGAT proposes to merge the two graphs into a complete knowledge graph, and to complete the recommendation task on this graph. KGAT constructs a recommendation model based on graph attention network GAT, which significantly improves the recommendation effect.

3.3 Interactive Knowledge-Aware Attention Network Algorithmic Model for Recommendations (IKANAM)

IKANAM models the items in each history record separately when propagating the user-side history records in the knowledge graph, and then uses a knowledge-aware attention network to model the information at the item granularity to obtain a knowledge graph-enhanced item representation, and also uses an interaction-based attention network to distinguish the importance of different items in the user's history records, thus improving the recommendation The accuracy of the results is improved. The overall structure of IKANAM is shown in Figure 3.

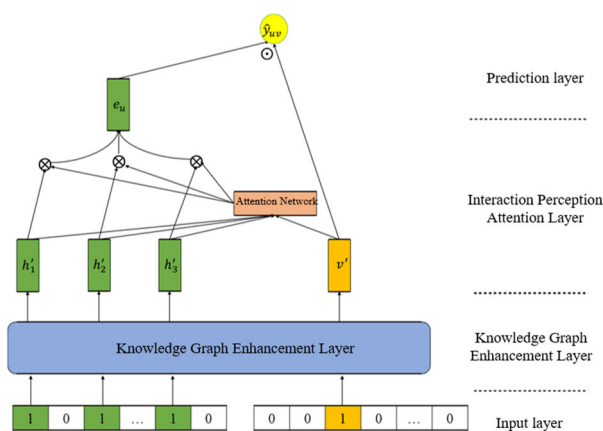


Fig. 3 Overall structure of IKANAM

After getting the set of entities corresponding to items and entities corresponding to users, the knowledge graph augmentation layer uses the knowledge graph to augment the representation of each entity. For each entity, the neighbor entities of the entity in each layer of the knowledge graph are

first obtained through the knowledge graph propagation at the item granularity, and then the weights of these neighbor entities are obtained through the knowledge-aware attention network to obtain the weights of these neighbor entities are obtained through the knowledge-aware attention network to obtain the neighbor representation of each layer. Finally, the final representation of the entity is obtained by multi-hop information aggregation. The structure of the knowledge graph enhancement layer is shown in Figure 4.

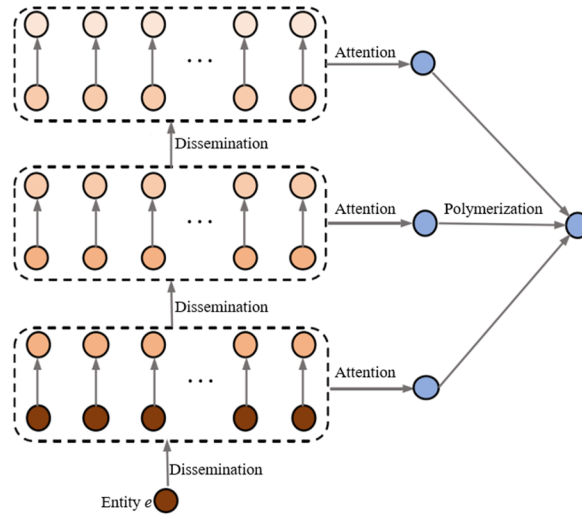


Fig. 4 Structure of the knowledge graph enhancement layer

We consider the neighboring entities of a single entity in the knowledge graph as the auxiliary information of that entity. The initial set of entities of entity e is denoted as $\varepsilon_e^0 = \{e\}$, and after propagation over the set of triples G , the neighboring entities ε_e^l and the triplet information T_e^l of the entity in the knowledge graph are obtained, respectively, expressed by the following equations.

$$\varepsilon_e^l = \{t | (h, r, t) \in G \text{ and } h \in \varepsilon_e^{l-1}, l = 1, 2, \dots, L \} \quad (3)$$

$$T_e^l = \{(h, r, t) | (h, r, t) \in G \text{ and } h \in \varepsilon_e^{l-1}, l = 1, 2, \dots, L \} \quad (4)$$

Where l is the number of propagation layers of the entity in the knowledge graph and L is the number of propagation layers of the knowledge graph set by the algorithm. For different entities, the size of their associated neighbors in the knowledge graph is different, and the size of the set of triples grows too fast as the number of knowledge graph propagation layers grows. For the set of entities in each layer as well as the set of triples, a randomly selected subset of fixed size can be used instead of directly. With this strategy, IKANAM can be implemented in a small batch manner and maintain computational efficiency.

We use a knowledge-aware attention network to assign different weights to the entities in the set of neighboring entities for each hop. The weights will be calculated considering the relationship information as well as the information of the entities themselves, and then the weighted sum of the set of entities is used to represent the set of entities. In this paper, the weighted sum of the tail entities in the set is used to represent the neighbor entities in that layer, and for each entity, a knowledge-aware attention network is used to calculate its weight. The l th-hop entity neighbor e_l of entity e is represented as follows.

$$e_l = \sum_i^{|\tau_e^l|} \pi_k(t_i, r_i) \cdot t_i, \quad l = 1, 2, \dots, L \quad (5)$$

Where i denotes the ordinal number of the triad in the set, t_i denotes the tail entity of the i th triad, r_i denotes the relationship of the i th triad in the set, and π_k denotes the knowledge-aware

attention network. $\pi_k(t_i, r_i)$ controls the weight of the tail entity in the whole set of triads by considering the tail entity itself and the relationship type. In the knowledge-aware attention network, ReLU is used as the nonlinear activation function, and Sigmoid is used as the activation function in the last layer. To normalize the weights of the attention network output, Softmax is used for the weights computed in each triad set, which is calculated as follows.

$$\pi_k(t_i, r_i) = \frac{\exp(\pi_k(t_i, r_i))}{\sum_{(h,r,t) \in T_e^l} \exp(\pi_k(t,r))} \quad (6)$$

3.4 Experiments and Results Analysis

1) Realize data acquisition and processing: In this paper, we crawl the information of fruit products and their attributes from the fresh fruit section of Jingdong Mall, an e-commerce platform, as a dataset to verify the validity of IKANAM. For the fresh fruit column on Jingdong website, we use "Houyi Collector" to crawl the name, origin, category, shelf life and other data of fruit products. We extracted the information about fruit products and their basic attributes, and removed the unwanted redundant information. The different attributes are marked to distinguish them. The knowledge extraction includes entity extraction and relationship extraction, identifying entities in the acquired data, such as product name, origin, brand, etc.; identifying the relationship between entities, for example, the correspondence between entity "Shanxi Red Fuji" and entity "Shaanxi" is "Origin". Entity alignment is divided into entity disambiguation and co-reference disambiguation. Entity disambiguation is when an entity name expresses multiple meanings, then it is necessary to split its label to form multiple entities. Co-referential disambiguation is when multiple entity names correspond to the same item, e.g., "kiwi" and "kiwi" refer to the same fruit, and the two entities need to be fused. The fruit item data is stored in the graph database according to the defined entity type relationship type. Almost all fruit products have "primary" and "secondary" categories of "fresh" and "fruit". Therefore, we directly use "Level 3" as the "Category" attribute of fruit products. After the above processing, a total of 3368 fruit products data were obtained as experimental data.

2) Comparison of algorithms and evaluation metrics: To verify the effectiveness of the model proposed in this paper, the comparison model used is Bayesian Personalized Ranking (BPRMF), Collaborative Knowledge base Embedding (CKE), Personalized Entity Recommendation (PER), Recommendation algorithms based on knowledge graph propagation (Ripple Net), Knowledge Graph Convolutional Networks (KGCN), Knowledge Graph Attention Network (KGAT), Collaborative Knowledge-aware Attentive Network (CKAN).

AUC can represent the probability that a positive sample is ranked ahead of a negative sample in the ranking result, and therefore can be used to measure the effectiveness of the ranking result. In order to balance the precision rate and the completion rate, the summed average F1 of both can be used as a metric to evaluate the effectiveness of the algorithm. Therefore, this paper uses Area Under Curve (AUC), which is the summed average of precision rate and completeness rate, as the evaluation index.

3) Experimental results and analysis: In the experimental process for each data set, it is divided according to the ratio of 6:2:2 as the training set, validation set, and test set respectively in the experimental process. The models were trained on the training set, and the test results of the models that performed well on the validation set were taken as the final index of the algorithm on the test set. The experimental results are shown in Figure 5.

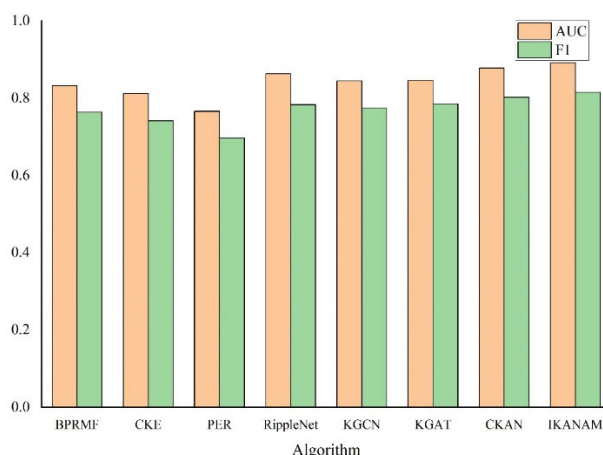


Fig. 5 Comparison of experimental results

From the experimental results, it can be observed that the AUC and F1 evaluation metrics of IKANAM proposed in this paper outperform other comparative algorithms on all three datasets.

4. Conclusion

With the rapid development of information technology, there has been a great change in the way people buy goods. Currently, especially under the influence of the epidemic, buying goods on online platforms has become a major way. The current recommendation systems of e-commerce platforms ignore the attribute information of goods, and the recommendation algorithms based on knowledge graph are the hot research spots in recent years. These algorithms use graph neural networks to perform graph embedding on the knowledge graph, so as to obtain the embedding vector of nodes, and use this embedding vector as the feature representation of nodes. Based on the obtained feature representations of users and items, users' ratings of items are predicted to complete the recommendation task. In the recommendation system, the number of user-item interaction records is small compared with the size of users and items, so the existing recommendation algorithms often face the problem of sparse data. Based on this, this paper proposes a knowledge graph-based product recommendation model, which introduces auxiliary information to help improve the accuracy of recommendation results.

References

- [1] M. Q. Xu, X. Xiong, B.Y. Li, et al. Knowledge graph-based human job recommendation system construction. *Computer Application Research*,2022,39(01):194-198.
- [2] H. Tang, S. Fan, F. Tang, et al. A recommendation algorithm integrating knowledge graph and attention mechanism. *Computer Engineering and Applications*,2022,58(05):94-103.
- [3] H. L. Yu, L. Sun, Accurate recommendation algorithm of AMI resources based on knowledge graph. *Computer Simulation*, 2021,38(12):485-489.
- [4] X. W. Guo, H. B. Xia, Y. Yuan, A hybrid recommendation model fusing knowledge graph and graph convolutional network. *Computer Science and Exploration*,2022,16(06):1343-1353.
- [5] Q. Q. Qin, Y. Q. Zhang, R. Z. Wang, et al. A hierarchical granular recommendation method based on knowledge graphs. *Computer Science*,2022,49(08):64-69.
- [6] Z. C. Yang, Y. Tang, Research on the application of knowledge graph in recommendation system. *Journal of Shanghai Electric Power University*,2022,38(02):163-170.
- [7] Y. G. Ren, F. Z. Lv, Z. P. Zhang, Personalized sequence recommendation by fusing knowledge graph and attention mechanism. *Small Microcomputer Systems*, 2022,43(07):1362-1369.DOI:10.20009/j.cnki.21-1106/TP.2022-0162.

- [8] J. Wang, Z. B. Xie, J. T. Zhao, et al. A financial product recommendation system based on knowledge graph and user portraits. *Computer Applications*,2022,42(S1):43-47.
- [9] G. Wu, X. N. Wang, Y. L. Liu, Advances in knowledge graph-enhanced graph neural network recommendation research [J/OL]. *Computer Engineering and Applications*:1-16[2023-02-13]. <http://kns.cnki.net/kcms/detail/11.2127.TP.20221103.1125.003.html>.
- [10] M. X. Zhang, S. D. Zhang, S. H. Liu, et al. A review of research on recommender systems using knowledge graphs [J/OL]. *Computer Engineering and Applications*:1-16 [2023-02-13]. <http://kns.cnki.net/kcms/detail/11.2127.TP.20221108.1435.002.html>.
- [11] B. Zhang, W. Wang, Y. Du, et al. Fusion attention mechanism and knowledge graph neural collaborative recommendation algorithm[J/OL]. *Computer Engineering and Applications*:1-12 [2023-02-13]. <http://kns.cnki.net/kcms/detail/11.2127.tp.20221127.1059.004.html>.