Research on recommendation algorithm of Graph attention Network based on Knowledge graph

Yiping Zeng¹, Shumin Liu²
1 School of Software Engineering, Jiangxi University of Science and Technology, Nanchang, Jiangxi, 330013, China
2 School of Software Engineering, Jiangxi University of Science and Technology, Nanchang, Jiangxi, 330013, China
*Corresponding author’s e-mail: 6720190972@mail.jxust.edu.cn

Abstract. The introduction of knowledge graph as the auxiliary information of recommendation system provides a new research idea for personalized intelligent recommendation. However, most of the existing knowledge graph recommendation algorithms fail to effectively solve the problem of unrelated entities, leading to inaccurate prediction of potential preferences of users. To solve this problem, this paper proposes a KG-IGAT model combining knowledge graph and graph attention network, and adds an interest evolution module to graph attention network to capture user interest changes and generate top-N recommendations. Finally, experimental comparison between the proposed model and other algorithms using public data sets shows that KG-IGAT has better recommendation performance.

1. Introduction
As more and more data in the Internet can be perceived and acquired, multi-source heterogeneous data including images, texts and labels contain rich user behavior information and personalized demand information[1], and the hybrid recommendation method integrating multi-heterogeneous auxiliary information has attracted more attention from researchers. Knowledge Graph (KG) aims to describe various entities or concepts and their relationships in the real world, and constitutes a huge semantic network Graph, which facilitates the effective integration of various auxiliary information into the recommendation algorithm, thus alleviating the problems of cold start and data sparsity.

Wu X Y et al. [2] use KG feature learning embedding semantic data into low dimension space, by calculating the semantic similarity between projects itself into semantic information of the collaborative filtering recommendation. Wang et al. [3] proposed a RippleNet algorithm that applies KG to the recommendation system for predicting web page click rate. As the graph attention network GAT proposed by literature [4] et al. uses the attention mechanism to replace the static normalized convolution operation, which avoid the noise influence caused by excessive information of neighbor nodes. Wang et al. [5] proposed a new heterogeneous graph attention network based on attention mechanism, providing a new idea for the application of graph attention network to heterogeneous graph. Based on literature [6] KGAT algorithm is put forward, in the end-to-end way explicitly modeling the relationship of higher order connectedness in KG, but this method is computationally expensive. Subsequently, KG was combined with graph convolution network [7] and KGCN[8] was proposed. Literature [9] proposed to use user preference tree to solve the problem of irrelevant entities
when exploring user potential preferences along the relationship in KG. However, for large-scale KG, the construction of user preference tree is too complex.

The main contributions of this paper are as follows:
(1) KG merged into the field of collaborative filtering recommendation framework, designed an end-to-end collaborative knowledge map network.
(2) Based on the drawing attention network, points for attention to update the threshold cycle unit of user interest model. So as to obtain user preferences more accurately and achieve personalized recommendations.

2. KG-IGAT algorithm
The overall framework of KG-IGAT model is shown in Figure 1 below.

**Figure 1. General framework of KG-IGAT**

2.1 Pretreatment Layer
In this paper, TransR[11] method is used to preprocess collaborative knowledge graph $G$. For any given triplet $(h, r, t)$, whether or not it actually exists in $G$, the following distance formula is used to describe its authenticity:

$$g(h, r, t) = || M_r e_h + e_r - M_r e_t ||_2$$

(1)

Where $M_r \in \mathbb{R}^{kd}$ is the transformation matrix of the relation $r$, which is used to project the entity from the d-dimensional entity space to the K-dimensional relation space; $|| \cdot ||_2$ represents L2 regularization to prevent overfitting. Here, a lower score means that a given triple $g(h, r, t)$ is more likely to be real in $G$. 

```plaintext
MCTE 2021  
Journal of Physics: Conference Series  2113 (2021) 012085  
IOP Publishing  
doi:10.1088/1742-6596/2113/1/012085
```
2.2 Attention embedding communication layer

2.2.1 Information dissemination

After the information of G is embedded into the low-dimensional space, the corresponding embedding vector \( e \) is generated, and then the information dissemination stage.

Given an entity \( h \), \( N_h = \{(h, r, t) \mid (h, r, t) \in G\} \) represents a collection of triples, where \( h \) is the header entity. Then, the representation vector \( e_{N_h} \) of \( N_h \) can be calculated by:

\[
e_{N_h} = \sum_{(h, r, t) \in N_h} \pi(h, r, t)e_t
\]

(2)

Where, \( \pi(h, r, t) \) is the attenuation factor controlling each transmission of the triplet, indicating how much information is transmitted from \( t \) to \( h \) through \( r \).

2.2.2 Based on attention mechanism

The second stage of embedding the propagation layer is based on the attentional mechanism. The calculation formula of attention mechanism is as follows:

\[
\pi(h, r, t) = (M_r e_t)^\top \tanh((M_h e_h + e_t))
\]

(3)

Where, \( \tanh \) is nonlinear activation function; \( M_r \) is the transformation matrix on the relation \( r \) generated in the embedding layer. The size of \( \pi(h, r, t) \) depends on the distance between node \( h \) and neighbor node \( t \) in relation \( r \) in the embedded layer. Then, softmax normalized the attention score calculated for all triples \( (h, r, t) \) as follows:

\[
\pi(h, r, t) = \frac{\exp(\pi(h, r, t))}{\sum_{(h', r', t') \in N_h} \exp(\pi(h', r', t'))}
\]

(4)

2.2.3 Information aggregation

The entity representation vector \( e_h \) of \( h \) and the neighborhood representation vector \( e_{N_h} \) obtained from its neighbors in the process of information collection are aggregated as the representation form of the first-order neighbor information aggregated by entity \( h \), denoted as \( e_h^{(1)} \), \( e_h^{(1)} = f(e_h, e_{N_h}) \), \( f(\cdot) \) is the aggregator function. A graph aggregator is used to connect two vectors for nonlinear transformation, \( f_{GraphSage} = \text{LeakyReLU}(e_h \parallel e_{N_h}) \).

2.2.4 High-order propagation

Next, update \( e_h \) iteratively to obtain higher-order information of \( h \). In this paper, the embedding of \( l \) -order information of entity \( h \) is denoted as \( e_h^{(l)} \), and the recursive updating formula is as follows:

\[
e_h^{(l)} = f(e_h^{(l-1)}, e_t^{(l-1)})
\]

(5)

\[
e_{N_h}^{(l)} = \sum_{(h, r, t) \in N_h} \pi(h, r, t)e_t^{(l-1)}
\]

(6)

\( e_t^{(l-1)} \) is the \((l-1)\) order representation of entity \( t \), storing the neighborhood information of \( t \).

2.2.5 Evolution of interest

In order to deeply explore how user Interest evolution module is added in this stage, this module adopts a new structure-- AUGRU[10] to model user interest, as shown in Figure 2.
AUGRU can be expressed by the formula:

\[ \tilde{u}_t = \pi_t \odot u'_t \]  \hspace{1cm} (7)

\[ \tilde{h}_t = (1 - \tilde{u}_t) \hat{h}_{t-1} + \tilde{u}_t \tilde{h}_t \]  \hspace{1cm} (8)

Where \( u'_t \) is the initial state of updating gate in AUGRU, \( \tilde{u}_t \) is the state of updating gate with attention score \( \pi_t \) added, and \( \tilde{h}_t \), \( \hat{h}_{t-1} \) and \( \tilde{h}_t \) are the states of hidden layer at AUGRU \( t \) moment.

In this interest evolution module, AUGRU uses attention score \( \pi_t \) to influence the update gate in GRU. After the addition of \( \pi_t \), users' interests are focused according to \( \pi_t \), and there are fewer attributes corresponding to users' interests in the items to be recommended.

2.3 Prediction layer of model

After the L-layer superposition, we can obtain the multi-scale representation of user node \( u \) and item node \( i \) as \( \{ e^{(1)}_u, ..., e^{(L)}_u \} \) and \( \{ e^{(1)}_i, ..., e^{(L)}_i \} \) respectively. Inspired by the multi-scale fusion in the image field[11], the hierarchical aggregation mechanism is adopted to connect the representation of each step into a vector representation, such as: \( e^*_u = e^0_u \| \cdots \| e^L_u \), \( e^*_i = e^0_i \| \cdots \| e^L_i \). The above operations can not only adjust the initial embedding parameters, but also control the scope of information aggregation and strengthen the propagation by adjusting L. Finally, this paper predicts the matching score of user and project through the inner product of their representation: \( \hat{y}(u,i) = \langle e^*_u \rangle^T e^*_i \), sort \( \hat{y}(u,i) \) in descending order to generate a top-N list.

3. Experimental research

3.1 Experimental Settings

In this paper, movielens-1M data set is selected to conduct comparative experiments between the model KG-IGAT in this paper and existing recommendation algorithms, including embedding based recommendation algorithm CKE[12], path-based recommendation algorithm RippleNet [3] and KGCN [8]. MovieLens is a public, standardized collection of data about how people rate movies. In this paper, DBpedia ontology knowledge base [13] was used to build a knowledge map for movielens-1M dataset, and users, movies and their interactive information were mapped and linked to corresponding entities, and the attributes in the dataset were all regarded as relations. In addition, a user triplet (user, gender, age) is constructed, corresponding to the user's ID, gender, age and other
attribute information. The statistics of the dataset are shown in Table 1 below. In this paper, 80% of the interactive information in the data set was randomly selected as the training set and the remaining 20% as the test set.

| Table 1. Data set statistics |
|-----------------------------|
| Data set                    | MovieLens-1M     |
| Users                       | 6040             |
| Movies                      | 3883             |
| Interaction                 | 100209           |
| Entity                      | 13593            |
| Item KG                     |                  |
| Relation                    | 3                |
| Triple                      | 25462            |

For the generated top-N list, that is, the most consistent recommendations of the first N items are given. In this paper, NDCG and recall are used to evaluate the model recommendation performance.

NDCG evaluates the sorting performance through the position of the item in the top-N list, so as to obtain the gap between the top-N list and the user's real interaction list,

\[
NDCG @ N = \frac{1}{IDCG} \sum_{i=1}^{N} \frac{{2}^{rel_i} - 1}{\log_2(i + 1)}.
\]

Where, \(rel_i\) represents the probability of the project at the \(i\) position, and IDCG is the maximum DCG value under the ideal state,

\[
IDCG @ N = \sum_{i=1}^{|REL|} \frac{{2}^{rel_i} - 1}{\log_2(i + 1)}.
\]

Where, \(|REL|\) is the set of the first N terms of the recommended results arranged according to the highest to lowest probability. It can be seen that the higher the item is in the list, the larger the NDCG is.

Recall is an indicator to measure the proportion of items in the top-N list in the test set,

\[
recall @ N = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|},
\]

Where \(R(u)\) is the list of recommendations generated for user \(u\) and \(T(u)\) is the list of items that the user actually interacts with in the test set. The more accurate the recommendation, the larger the proportion of \(T(u)\) in \(R(u)\), and the larger the recall.

The initial learning rate of KG-IGAT was set to 0.001, the L2 regularization coefficient was set to 1E-5, and the embedding dimension was set to 64. Adam optimizer was used for optimization, and all parameters in the model parameter set \(\Theta\) were updated and adjusted.

### 3.2 Experimental results and analysis

In this section, \(L\) is selected as 1, 2, 3 and 4 respectively to determine the most appropriate model depth, and \(N=20\). The experimental results are shown in Table 2:

As can be seen from the table, with the increase of \(L\), recall and NDCG also increase. When \(L=4\), it is found that the values of recall and NDCG decrease somewhat, indicating that the larger \(L\) is, the better is not. In addition, when \(L=3\), recall and NDCG values are the maximum, indicating the best model performance. When the threshold is exceeded, the performance degradation is due to the effect of noise on the model, so \(L=3\) is selected in this paper for subsequent experiments.

Taking \(L=3\) as the optimal model depth, the performance of each model at \(N=20\) is compared respectively in the following Table 3:
Table 3. Performance comparison of each model index

| model      | recall@20 | NDCG@20 |
|------------|-----------|---------|
| CKE        | 0.2323    | 0.6478  |
| RippleNet  | 0.1441    | 0.5235  |
| KGCN       | 0.1572    | 0.5532  |
| KG-IGAT    | 0.2441    | 0.6697  |

Table 3 shows that when N=20, recall@20: KG-IGAT > CKE > KGCN > RippleNet and NDCG@20: KG-IGAT > CKE > KGCN > RippleNet. RippleNet and KGCN may be superior to CKE in other aspects, but from the two evaluation indexes in this paper, CKE is superior to RippleNet and KGCN. In general, the model KG-IGAT in this paper has the best performance.

In addition, Figure 3 and Figure 4 respectively show the comparison of recall index and NDCG index performance of each model.

The below four models are all based on KG. With the increase of N, the performance of each model shows an overall upward trend. In terms of recall and NDCG, the performance of the proposed model is superior to the other three models. When N=20, recall value: KG-IGAT is 1 percentage point, 10 percentage point and 9 percentage point higher than CKE, RippleNet and KGCN, respectively. NDCG value: KG-IGAT is 2, 14 and 11 percentage points higher than CKE, RippleNet and KGCN.

4. Conclusion

This paper proposes a model, KG-IGAT, which makes full use of the information of the central node and the neighbor node in the process of embedded propagation to model, and then disseminates the information aggregation to a higher order. Meanwhile, the evolution of user interest is integrated into the attention mechanism to model, so as to capture the changes of user interest more accurately. A large number of experimental data show that when N=20, the recall rate and NDCG of this model are 24.4% and 66.9%, respectively. Therefore, this model is superior to the other three models and has good robustness.

For the research of intelligent recommendation algorithm, it can combine various algorithms to mix recommendation in the future in order to achieve better results.
Acknowledgement
This work was supported in part by National Natural Science Foundation of China (No. 61761021), Natural Science Foundation of Jiangxi Province (Grant No. 20181bab202018, 20202BAB212003), Special Fund for Postgraduate Innovation of Jiangxi Province(No. YC2020-S481).

References
[1] HUANG L W, JIANG B T, LYU Y B, et al. Survey on deep learning based recommender systems[J]. Chinese Journal of Computers: 2018, 41(7): 1619-1647.
[2] Wu X, Chen Q, Liu H, et al. Collaborative filtering recommendation algorithm based on KG representation learning[J]. Computer engineering, 2018, 44(2): 226-232, 263.
[3] Wang H W, Zhang F Z, Wang J L, et al. Ripple Net: propagating user preferences on the knowledge graph for recommender systems[C] // Proc of the 27th ACM International Conference on Information and Knowledge Management. New York: ACM Press, 2018: 417-426.
[4] CUCURULL G, CASANOVA A, et al. Graph attention networks [J]. ar Xiv: 1710.10903, 2017.
[5] Wang X, Ji H Y, Shi C, et al. Heterogeneous graph attention network[C] // Proc of World Wide Web Conference.New York: ACM Press, 2019: 2022-2032.
[6] Wang X, He X N, Cao Y X, et al. KGAT: knowledge graph attention network for recommendation[C] // Proc of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2019: 950-958.
[7] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[J]. ar Xiv: 1609. 2907, 2016.
[8] Wang H W, Zhao M, Xie X, et al. Knowledge graph convolutional networks for recommender systems[C] // Proc of World Wide Web Conference. New York: ACM Press, 2019: 3307-3313.
[9] Li Q Y, Tang X L, Wang T Y, et al. Unifying task oriented knowledge graph learning and recommendation[J]. IEEEAccess, 2019, 7: 115816-115828.
[10] Zhou G, Mou N, Fan Y, et al. Deep Interest Evolution Network for Click-Through Rate Prediction[J]. In Proceedings of the AAAI Conference on Artificial Intelligence, 2019: 5941-5948.
[11] Zhao J, Qian Y, Nan F, et al. CNN multi-layer feature fusion and ELM for breast disease diagnosis[J]. Computer engineering and applications, 2020, 56(4): 122-127.
[12] Zhang F Z, Yuan N J, Lian D F, et al. Collaborative knowledge base embedding for recommender systems[C] // Proc of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2016: 353-362.
[13] Bizer C, Lehmann J, Kobilarov G, et al. DBpedia: a crystallization point for the Web of data[J]. Journal of Web Semantics, 2009, 7(3): 154-165.