

Scholar Recommendation Based on High-Order Propagation of Knowledge Graphs

Pu Li, Zhengzhou University of Light Industry, China*

Tianci Li, Zhengzhou University of Light Industry, China

Xin Wang, Zhengzhou University of Light Industry, China

Suzhi Zhang, Zhengzhou University of Light Industry, China

Yuncheng Jiang, South China Normal University, China

Yong Tang, South China Normal University, China

ABSTRACT

In a big data environment, traditional recommendation methods have limitations such as data sparseness and cold start, etc. In view of the rich semantics, excellent quality, and good structure of knowledge graphs, many researchers have introduced knowledge graphs into the research about recommendation systems and studied interpretable recommendations based on knowledge graphs. Along this line, this paper proposes a scholar recommendation method based on the high-order propagation of knowledge graph (HoPKG), which analyzes the high-order semantic information in the knowledge graph and generates richer entity representations to obtain users' potential interest by distinguishing the importance of different entities. On this basis, a dual aggregation method of high-order propagation is proposed to enable entity information to be propagated more effectively. Through experimental analysis, compared with some baselines, such as Ripplenet, RKGE, and CKE, the method has certain advantages in the evaluation indicators AUC and F_1 .

KEYWORDS

Dual Aggregation, High-Order Propagation, Knowledge Graph, Scholar Recommendation

1. INTRODUCTION

In recent years, people have entered an era of information explosion, due to the rapid development of information technology. A large number of data is being generated all the time in daily life. How to use these data more effectively to facilitate our lives is an urgent problem in the field of current information science. Along this line, the recommendation system came into being and was applied in many aspects of life. From e-commerce platforms, search engines, social platforms to short video platforms, portal websites, mobile applications, etc. All of them have certain recommendation functions (such as user behavior prediction, user interest perception, etc.). Traditional recommendation systems can be roughly divided into two types: content-based recommendation and collaborative filtering-based recommendation. They provide recommendations based on the similarity of the content, the users'

DOI: 10.4018/IJSWIS.297146

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

interactive behavior, and so on. These two classic recommendation systems are widely used because of their better recommendation effects. However, facing the current massive amount of information, traditional recommendation methods generally suffer from data sparseness and cold start (Guo, Zhuang, Qin, Zhu, Xie, Xiong, & He, 2020). To solve these problems, many studies have introduced knowledge graphs into recommendation research, hoping to realize interpretable recommendations from the perspective of semantics.

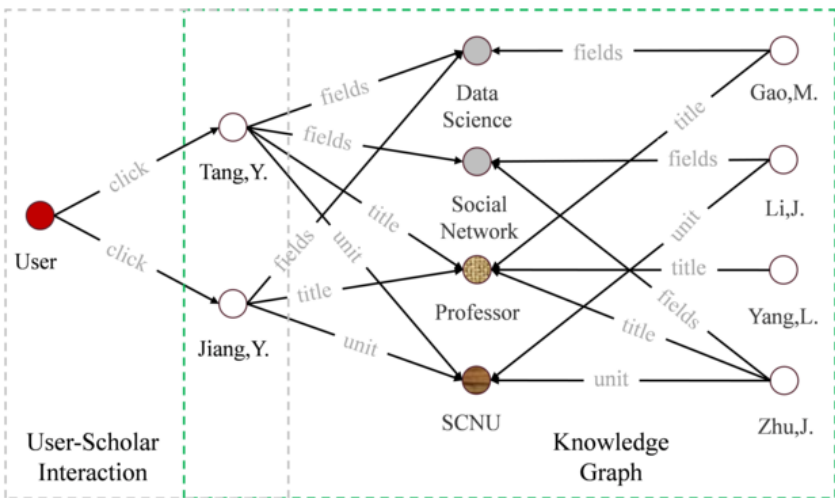
As a kind of semantic network, knowledge graph contains rich information. A typical knowledge graph consists of nodes and edges. Nodes and edges are used to represent entities and the relationships between entities. Specifically, the knowledge graph is composed of many “subject-predicate-object” triples, which can be denoted as “(S, P, O)”. Since the knowledge graph contains rich semantic relations, it is an effective attempt to use it as the information source of the recommendation system for the interpretable recommendation. Meanwhile, as a link in the use and creation of knowledge, the academic field are filled with a large amount of knowledge. This paper takes scholars as the subject, analyzes the research fields of scholars and the relationship between different scholars, uses scholars’ knowledge to construct a knowledge graph, and recommends related scholars based on the knowledge graph, which has certain theoretical significance for grasping academic trends, scientific and technological frontiers, as well as the development of research work and the introduction of talents.

1.1 A Background Example

A real example of scholar recommendation based on knowledge graph is shown in Figure 1.

It can be seen from Figure 1 that when the user “User” clicks on the scholars “Tang, Y.” and “Jiang, Y.”, we can assume that “User” pays more attention to these two scholars. By observing the nodes connected with the two scholars, we can find that the two scholars have the same research fields “Data Science”, the title “Professor” and the unit “SCNU”. Therefore, based on the nodes closely connected to these three nodes, we can infer that “User” may be more interested in the scholar “Gao, M.”, because this scholar also has the same title “Professor” and the research fields “Data Science”. At the same time, it can be inferred that “User” is more interested in the scholar “Zhu, J.”. Because he has the same title “Professor”, the unit “SCNU”, and he also has the same research fields “Social Network” with one of the scholars “Tang, Y.”. In addition, the scholar “Li, J.” has the same unit “SCNU” with the two scholars, but only has the same research fields “Social Network” with one of the scholars “Tang, Y.”. Therefore, we can infer that “User” may be interested in “Li, J.”. Similarly,

Figure 1. A Real Example of Scholar Recommendation Based on Knowledge Graph



for the scholar “Yang, L.”, although he shares the title “Professor” with the two scholars, there is no other connection to indicate that it is related to the two scholars. Therefore, we infer that “User” is less likely to be interested in him. It can be seen that in the research of introducing the knowledge graph into the recommendation system, a more reasonable recommendation effect and certain semantic interpretability can be obtained.

1.2 Main Contributions

Taking this as a starting point, we take the knowledge graph as the object and conducts further research on the recommendation system. The main contributions of this paper are as follows: (1) Construct scholar knowledge graph and user-scholar interaction based on scholar data, and input them into the recommendation system as a data source. (2) A new high-order information propagation method based on the knowledge graph is proposed. According to the relationships of entities, it explores the high-order propagation between entities, thereby discovering the potential information in the knowledge graph. In addition, to better enrich the entity representation, a dual entity aggregation method based on high-order propagation is proposed. (3) Through experimental analysis, it proves the validity and interpretability of the HoPKG for recommendation, and it has certain advantages over the previous methods in evaluation indicators AUC and F_1 .

The remainder of this paper is organized as follows. Section 2 summarizes the relevant research in recent years. Section 3 defines the problem and delves into our proposed model. Section 4 conducts experiment on the dataset and compares it with the baselines to analyze the results. Section 5 summarizes the work of this paper and looks forward to future work.

2. RELATED WORKS

Since Google released Google Knowledge Graph in 2012, knowledge graph has developed rapidly and has been widely used in various fields, such as data analysis, intelligent retrieval, intelligent recommendation, human-computer interaction, etc.(Indra & Thangaraj, 2019; Jadhav & Patil T, 2017; Lee, He, Su, Chen, Xiao, & Yang, 2020; Wu, Shen, Deng, & Cheng, 2019) As a large-scale semantic network, the knowledge graph contains entities, concepts, and various rich semantic relationships(Heist, Hertling, Ringler, & Paulheim, 2020). Compared with the traditional semantic network, the knowledge graph has the characteristics of large scale, rich semantics, good quality and structure. Therefore, to improve the recommendation effect, many studies use knowledge graphs as the information source of the recommendation system. It has gradually attracted the attention of researchers and has been widely used in search, query, and recommendation fields(Li, Jiang, Wang, & Yin, 2017; Li, Li, Zhang, Li, Tang, & Jiang, 2020; Li, Xiao, Akram, Jiang, & Zhang, 2018; Li, Xiao, Ma, Jiang, & Zhang, 2017). Currently, there are three methods for implementing recommendations using knowledge graphs: embedding-based methods, path-based methods, and propagation-based methods.

2.1 Embedding-Based Methods

The embedding-based methods implement entity embedding (Gesese, Biswas, & Sack, 2019) by using the translation model and integrate it into the recommendation system to achieve recommendation. (Zhang, Yuan, Lian, Xie, & Ma, 2016) proposed a collaborative knowledge based embedding for recommender systems (CKE), which extracts structured content, text content, and visual content from the knowledge base, using the TransR(Lin, Liu, Sun, Liu, & Zhu, 2015), denoising autoencoders and convolutional autoencoders embed these three contents into the representation. In addition, the integrator integrates the three representations into the implicit vector of the items, and finally obtains the score through the implicit vector of the items and the users. (Wang, Zhang, Xie, & Guo, 2018) proposed the deep knowledge-aware network (DKN), which uses CNN(Krizhevsky, Sutskever, & Hinton, 2017) to learn text embedding in sentences and TransD(Ji, He, Xu, Liu, & Zhao, 2015) to

model the embedding of entities in news content, and according to this calculates the users' preferences for news. (Wang, Zhang, Zhao, Li, Xie, & Guo, 2019) proposed a multi-task feature learning recommendation model (MKR), which is composed of a knowledge graph embedding module and a recommendation module. These two modules share knowledge through cross and compression unit links. The recommendation module is trained to estimate the users' preferences for the product, and the knowledge graph embedding module is trained to estimate the tail entity representation based on the relationship and the head entity. The two modules are trained alternately to obtain the users' real preferences. The embedding-based methods have good scalability, but they ignore the information between higher-order entities, and the interpretability is poor.

2.2 Path-Based Methods

The path-based methods explore the connection between two entities according to the connection path existing in the knowledge graph and realize the recommendation accordingly. (Zhao, Yao, Li, Song, & Lee, 2017) proposed FMG, which captures the complex relationships between entities in heterogeneous graphs by replacing meta-paths with meta-graphs. On this basis, a similarity matrix is established, and a factorization machine is used to fuse users and items features into different meta-graphs to obtain users' preferences. (Hu, Shi, Zhao, & Yu, 2018) proposed MCRec, which uses CNN to learn the embedding of each path instance and calculates the meta-path embedding through a pooling operation. In addition, the weighted average value of meta-path embedding is used to obtain the interactive embedding between the users and the items, thereby obtaining users' preferences. The path-based methods have certain interpretability, but due to the need of constructing meta-graphs and meta-paths, they have poor scalability and are prone to information loss.

2.3 Propagation-Based Methods

The propagation-based methods use embedding to improve the entity representation (Ji, Pan, Cambria, Marttinen, & Yu, 2021) by integrating the multi-hop neighbor representation to obtain richer information. (Wang, Zhang, Wang, Zhao, Li, Xie, & Guo, 2018) proposed RippleNet, which overcomes the limitations of the embedding-based and path-based methods by introducing a preference propagation mechanism. It automatically propagates the users' potential preferences and uses the KGE (Wang, Mao, Wang, & Guo, 2017) regularization method in the Bayesian framework to unify the preference dissemination, and then predict the click-through rate. (Tang, Wang, Yang, & Song, 2019) proposed AKUPM, which uses the TransR to model entities. In addition, it uses the self-attention mechanism to assign weights to entities and obtain entities representations to predict users' preferences. The propagation-based methods can make full use of the information in the knowledge graph and have a good recommendation effect and interpretability. However, due to their high computational complexity and noise, the recommended results will be influenced by varying degrees.

Due to the poor performance, data sparseness and high computational complexity of the above methods, and inspired by the ideas of GCN (Kipf & Welling, 2016) and GraphSage (Hamilton, Ying, & Leskovec, 2017), we propose a scholar recommendation method based on high-order propagation of knowledge graph.

3. METHODOLOGY

This section will describe the specific content of the scholar recommendation method based on the high-order propagation of the knowledge graph. First, construct the input part of the recommendation system based on the source data, then design the recommendation method based on the input part, and finally use the predictive function to predict the score.

3.1 Preliminaries

Firstly, we will define some basic concepts:

Definition 1 (Knowledge Graph): The knowledge graph consists of many triples, which can be regarded as a collection of relations and entities (also called edges and nodes). According to the relationships between entities, we define entities and relationships as triples, and mark them as (h, r, t) , where h represents the head entity, r represents the relationship, and t represents the tail entity. Each triple is connected by the head entity h and the tail entity t through a relationship r . Therefore, we extract triples from the source data to construct a knowledge graph, and define it as $G = \{(h, r, t) | h, t \in E, r \in R\}$, where E represents the entity set, R represents the relationship set.

Definition 2 (User-Scholar Interaction): User-Scholar Interaction is a directed graph, which is consisted of user and scholar. It is constructed by the history of user-to-scholar interaction. Let $U = \{u_1, u_2, \dots, u_{d_1}\}$ denote the user set, let $S = \{s_1, s_2, \dots, s_{d_2}\}$ denote the scholar set. Where u_i denotes user i , d_1 denotes the number of users, $i \in [1, d_1]$, s_j denotes scholar, d_2 denotes the number of scholars, $j \in [1, d_2]$. Meanwhile, we can also reflect User-Scholar Interaction as an interaction matrix $I = \{\eta_{us} | u \in U, s \in S\}$ use η_{us} denote the interaction between user and scholar:

$$\eta_{us} = \begin{cases} 1, & \text{if an interaction between user } u \text{ and scholar } s \text{ is true} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $\eta_{us} = 1$ denotes that they generate an interaction or a positive comment between user u and scholar s , $\eta_{us} = 0$ denotes the opposite situation.

3.2 HoPKG

In this section, we will introduce our proposed HoPKG in detail. The main framework is shown as Figure 2. It is composed of 3 parts: Input Layer (User-Scholar Interaction and Knowledge Graph), Middle Layer (High-order Propagation, Single-Network and BiPart Aggregator) and Output Layer (Predicted Function). Then, we will introduce the Middle Layer from 3 aspects: High-order Propagation, Single-Network and BiPart Aggregator.

3.2.1 High-Order Propagation

Knowledge graph contains rich entities and semantic relations. Provided that we analyze the results by the nodes which are connected to the target nodes directly, it is hard to achieve precise recommendation results. Hence, it is significant to explore the high-order propagation of knowledge graph. We denote the nodes which are connected to the target nodes of the previous layer as neighbors, according to the structural characteristics of the knowledge graph. We denote the neighbor sampling size as $N(n)$, according to the number of neighbors. Where n represents the number of neighbors sampling, $N(n)$ represents the set of neighbors sampling. Using high-order propagation to diverge and expand from the target nodes makes it possible to use more useful information in the knowledge graph. With the increasing of propagation layer l , more and more information is obtained from the target node, as shown in Figure 3. Moreover, we are inspired by GraphSage and set a fixed neighbor sampling size according to the sparsity of the knowledge graph. For the target node, if its number of neighbor nodes is less than n , we will adopt a repeated sampling strategy, until all of the nodes are adopted. If its number of neighbor nodes is more than n , we will sample neighbor nodes randomly and unrepeatably. In this way, the information of neighbor nodes can be fully utilized, and unnecessary calculation overhead can be reduced.

Figure 2. The Framework of HoPKG

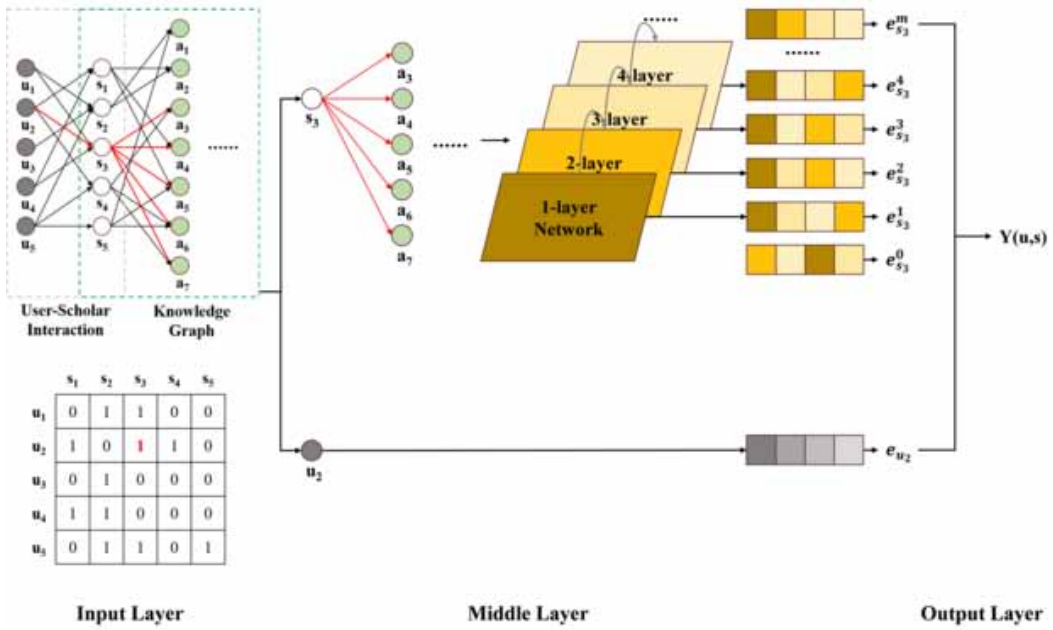
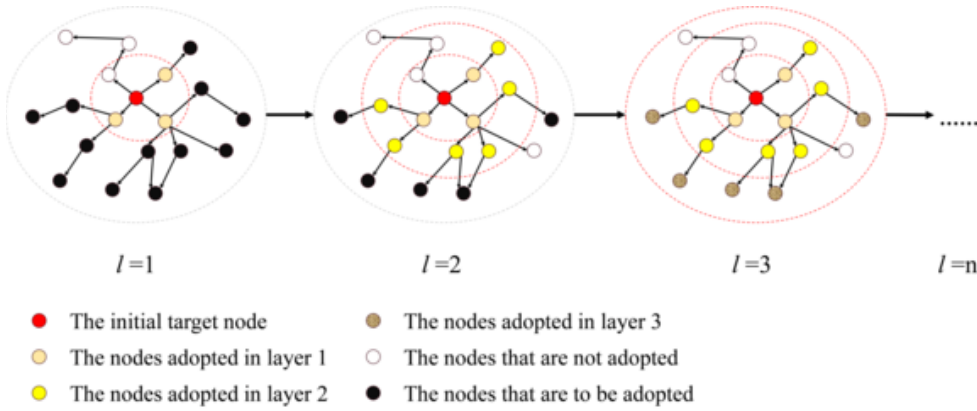


Figure 3. The Process of Knowledge Graph High-order Propagation



3.2.2 Single-Network

To achieve the high-order propagation of knowledge, we need to define the structure of every layer of the network. We obtain higher-order propagation in the knowledge graph according to the structure of each layer, to extract more semantic information in the knowledge graph. We define the structure of each layer as a single-network. It is similar to ego-network(Qiu, Tang, Ma, Dong, Wang, & Tang, 2018) and calculated by the target node's entity representation of the current layer and the entity representation of its neighbor nodes. The structure is shown as Equation 2:

$$e_{N(n)}^u = \sum_{t \in N(n)} \alpha_{u,t} e_t \quad (2)$$

where $\alpha_{u,t}$ represents the importance score of neighbor node (tail node) t to user u , e_t represents entity representation of current neighbor node t . It is also the importance level of the user's attention to the attributes of the current scholar (research field, title, work unit, etc.).

To distinguish the importance of different nodes to users, we obtain the normalized importance score through the *softmax* function, as following Equation 3:

$$\alpha_{u,t} = \text{softmax}\left(\theta(e_u, e_t)\right) = \frac{\exp\left(\theta(e_u, e_t)\right)}{\sum_{u, t \in N(n)} \exp\left(\theta(e_u, e_t)\right)} \quad (3)$$

where $\theta(e_u, e_t)$ represents the function of calculating the importance score between user nodes and neighbor nodes. Its calculating way is $\theta: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, e_u and e_t represent users' entity representation and neighbors' entity representation respectively:

$$\alpha_{u,t} = \frac{\exp\left(\text{LeakyRelu}\left(\vec{e}_u \parallel \vec{e}_t\right)\right)}{\sum_{u, t \in N(n)} \exp\left(\text{LeakyRelu}\left(\vec{e}_u \parallel \vec{e}_t\right)\right)} \quad (4)$$

From Equation 3 we can infer Equation 4. Where, LeakyRelu denotes the Activation function.

3.2.3 BiPart Aggregator

Entity aggregator is the last part of high-order propagation of knowledge graph. The design strategy of the aggregation method determines the final entity representation. In this section, a dual aggregator BiPart is proposed. We will compare it with GraphSage Aggregator, to prove its effectiveness.

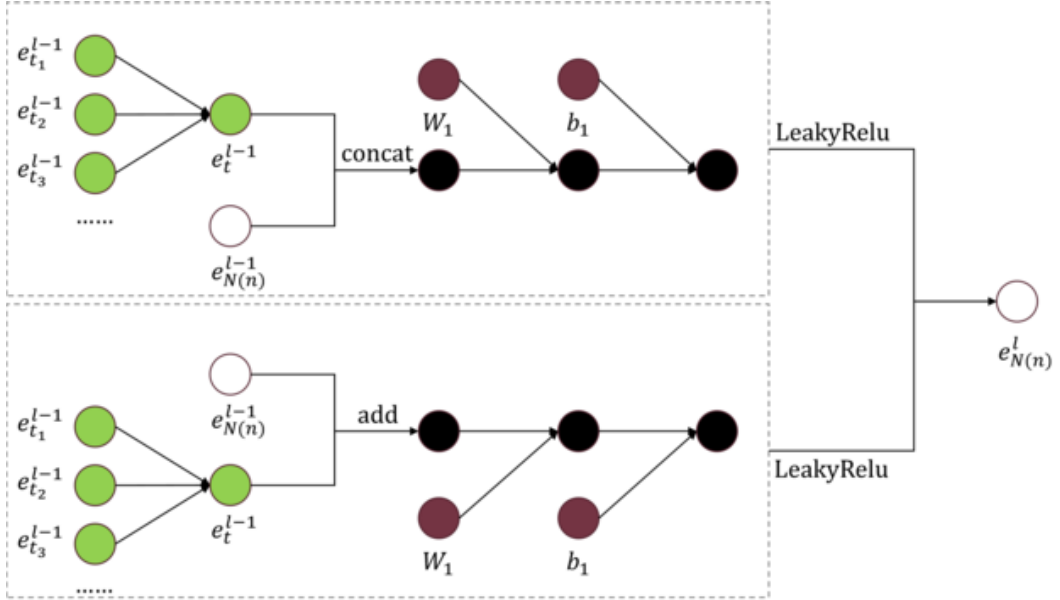
Aggregator-GraphSage (Hamilton, Ying, & Leskovec, 2017): It concatenates 2 types of entity representation (Neighbor entity representation and the entity representation calculate by single-network). Then, it obtains the current entity representation by nonlinear transformation. The specific process is as following Equation 5:

$$\text{Aggregator} - \text{GraphSage} : g_1 = \text{LeakyRelu}\left(W \cdot \text{Concat}\left(e_t, e_{N(n)}\right)\right) \quad (5)$$

where W denotes the learned weight matrix, *Concat* denotes the matrix's concatenation.

Aggregator-BiPart: It obtains the entity representation in 2 different ways (concatenation and summation), and combines 2 types of results to obtain rich entity representation (see Figure 4). The specific process is as following Equation 6:

Figure 4. The Structure of Aggregator-BiPart



$$\begin{aligned} \text{Aggregator} - \text{BiPart} : g_2 = & \text{LeakyRelu} \left(W_1 \cdot \text{Concat} \left(e_t, e_{N(n)} \right) + b_1 \right) \\ & + \text{LeakyRelu} \left(W_2 \cdot \left(e_t + e_{N(n)} \right) + b_2 \right) \end{aligned} \quad (6)$$

where W_1 and W_2 denote the learned weight matrix, b_1 and b_2 denote the offset.

During the high-order propagation of knowledge graph, the above 2 aggregators are shown as Equation 7 and Equation 8:

$$\text{Aggregator} - \text{GraphSage} : e_{N(n)}^l = g_1 \left(e_t^{l-1}, e_{N(n)}^{l-1} \right) \quad (7)$$

$$\text{Aggregator} - \text{BiPart} : e_{N(n)}^l = g_2 \left(e_t^{l-1}, e_{N(n)}^{l-1} \right) \quad (8)$$

From Equation 7 and Equation 8, we can infer Equation 9 and Equation 10:

$$\text{Aggregator} - \text{GraphSage} : e_{N(n)}^l = \text{LeakyRelu} \left(W \cdot \text{Concat} \left(e_t^{l-1}, e_{N(n)}^{l-1} \right) \right) \quad (9)$$

$$\begin{aligned} \text{Aggregator} - \text{BiPart} : e_{N(n)}^l = & \text{LeakyRelu} \left(W_1 \cdot \text{Concat} \left(e_t^{l-1}, e_{N(n)}^{l-1} \right) + b_1 \right) \\ & + \text{LeakyRelu} \left(W_2 \cdot \left(e_t^{l-1} + e_{N(n)}^{l-1} \right) + b_2 \right) \end{aligned} \quad (10)$$

where l denotes the layer of high-order propagation, $e_{N(n)}^l$ denotes the entity representation of the current layer. $e_{N(n)}^l$ is calculated by the entity representation $e_{N(n)}^{l-1}$ and neighbor entity representation e_t^{l-1} of last layer.

3.3 Prediction Function

On the basis of obtaining entity representation through high-order propagation and BiPart aggregator, we will discuss how to calculate the click-through rate (CTR) of users to scholars through the prediction function. Then, we train HoPKG into a CTR prediction model, to achieve precise and interpretable recommendation. We calculate the CTR of users to scholars by the following Equation 11:

$$Y(u, s) = \sigma(e_u, e_s^m) \quad (11)$$

where e_u denotes the entity representation of user u , e_s^m denotes the final entity representation of scholar s , σ denotes the sigmoid activation function. $Y(u, s) \in [0, 1]$ and the closer its score is to 1, the more important the scholar s is to the user u , and the more likely the user u is to be interested in the scholar s .

3.4 The Algorithm for HoPKG

To explore the high-order semantic information in the knowledge graph, the algorithm we designed is shown in Algorithm 1. First, we build a mapping matrix based on the set of neighbor entities obtained by the entity and the neighbor sampling size. Then, a new set of neighbor entities is generated according to the mapping matrix. Based on this, the attention score between the entity and its neighbor entities is calculated, and a high-order propagation mechanism is introduced to calculate the entity representation. Finally, calculate the score based on the prediction function and use the evaluation method to evaluate the experimental results.

Algorithm 1. HoPKG algorithm

Input: Knowledge Graph G , User-Scholar Interaction I , parameter: learning rate δ , L_2 normalization coefficient ε entity embedding dimension d , propagation layer l , aggregator g , neighbor sampling $N(n)$;

Output: Evaluating function γ ;

```
(1) while HoPKG not converge do
(2)   for  $(u, s)$  in  $I$  do
(3)      $L(l) \leftarrow s$ ;
(4)     for  $i = l-1, \dots, 0$  do
(5)        $L(i) \leftarrow L(i+1)$ ;
(6)       for  $e \in L(i+1)$  do
(7)          $L(i) \leftarrow L(i) \cup N(n)$ 
(8)     for  $i = 1, \dots, l$  do
(9)       for  $e \in L(i)$  do
(10)           $\alpha_{u,t} \leftarrow \frac{\exp(\theta(e_u, e_t))}{\sum_{u,t \in N(n)} \exp(\theta(e_u, e_t))}$ 
```

- $$(11) \quad e_{N(n)}^{i-1} \leftarrow \sum_{t \in N(n)} \alpha_{u,t} e_t^{i-1}$$
- $$(12) \quad e_{N(n)}^i \leftarrow g(e_t^{i-1}, e_{N(n)}^{i-1});$$
- $$(13) \quad e_s^l \leftarrow e_{N(n)}^l$$
- $$(14) \quad \text{Prediction function } Y(u, s) \leftarrow \sigma(e_u, e_s^l)$$
- $$(15) \quad \text{Update parameters};$$
- $$(16) \quad \mathbf{return} \ Y;$$

4. EXPERIMENTS

In this section, we will verify the effectiveness of the HoPKG model through experiments. By conducting experiments on the target dataset, obtain the optimal experimental parameters, and compare the average optimal results with the baselines.

4.1 Experimental Dataset and Baselines

In this section, we will introduce the dataset and baselines in experiments.

4.1.1 Experimental Dataset

This paper uses the background data from SCHOLAT¹ (Xu, Qiu, Lin, Tang, He, & Yuan, 2021) as the target dataset. It is the real scholar data saved in the SCHOLAT, including user interaction history, scholar's name, research field, work unit and title, etc. After data processing, the user-scholar interaction and knowledge graph are constructed, and a total of 11,256 interaction histories and 9,054 pairs of triples are generated. The detailed information of the dataset is shown in Table 1.

4.1.2 Baselines

In order to further illustrate the effectiveness of the method proposed in this paper, we will compare our model with several current classic recommendation models:

1. **PER(Yu, Ren, Sun, Gu, Sturt, Khandelwal, Norick, & Han, 2014):** It uses external information networks to construct meta-paths, defines global and personalized recommendation modules, and uses Bayesian ranking optimization to estimate the effect of the model.
2. **MCRec(Hu, Shi, Zhao, & Yu, 2018):** It Constructs the meta-path by deep neural network and a priority based sampling method, uses meta-path context to design a model and predict the results.

Table 1. Statistics of Dataset

Graph	Attributes	Scholar
User-Scholar Interaction	# Users	1,000
	# Scholars	1,589
	# Interactions	11,256
Knowledge Graph	# Entities	6,622
	# Relations	5
	# Triples	9,054

3. **CKE(Zhang, Yuan, Lian, Xie, & Ma, 2016):** It obtains the semantic representation of the project by extracting the structured content, text content and image content in the knowledge base, and finally assembles them into a CKE framework.
4. **RKGE(Sun, Yang, Zhang, Bozzon, Huang, & Xu, 2018):** It proposes a recurrent structure to model the semantic paths of the same entities which connected with each other, and then, puts them into the recommendation.
5. **RippleNet(Wang, Zhang, Wang, Zhao, Li, Xie, & Guo, 2018):** It uses historical interaction records to obtain high-order entity representations through the iterative propagation of connections in the knowledge graph, and then predicts users' potential interests.

4.2 Parameter Settings and Evaluating Indicators

4.2.1 Parameter Settings

All methods in the experiments are set and reproduced according to the best parameters in the original paper. It is worth noting that some methods used multiple sources of information in previous studies. Therefore, for the dataset used in this paper, we will only use this part as input. For our proposed HoPKG model, we use Adam (Kingma & Ba, 2015) as the algorithm optimizer. After preliminary experiments, we select the best learning rate δ in $\{0.001, 0.002, 0.01, 0.02\}$, and search the best coefficient ε of L_2 normalization in $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. Furthermore, we select the best dimension d of entity embedding in $\{16, 32, 64, 128, 256\}$, and select the best layer l of high-order propagation. At the same time, we repeat each group of experiments 10 times, and then take the average of best results.

Remark 1: About parameter setting, we set a rough parameter range based on the parameter settings of related papers. Preliminary experiments were conducted on this basis, and the parameter range was adjusted through the experimental results. Furthermore, parameters in this range can make the model converge and make the loss smaller, so a wider range was not considered.

4.2.2 Evaluating Indicators

We use CTR prediction and Top-K recommendation to evaluate our proposed model. During the CTR prediction, we implement the trained HoPKG model in test set, and evaluate the prediction results by evaluating indicators AUC and F_1 . During Top-K recommendation, we adopt Pre@K, Rec@K to evaluate the results.

AUC: Area under the ROC curve, it is defined as following Equation 12:

$$AUC = \frac{\sum_{i \in positive} rank_i - \frac{n_{pos}(1 + n_{pos})}{2}}{n_{pos} \times n_{neg}} \quad (12)$$

where n_{pos} denotes the number of positive samples, n_{neg} denotes the number of negative samples, $rank_i$ denotes the probability of the sample being classified as a positive sample.

F_1 : The harmonic mean of precision and recall, it is defined as following Equation 13:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (13)$$

4.3 Performance Comparison

The final experimental comparison results are shown in Table 2 and Figure 5. HoPKG performs best among all methods in AUC and F_1 . Specifically, as shown in Table 2, HoPKG outperforms the best baselines RippleNet by 3.81% in F_1 . The reason is that the BiPart Aggregator can make better use of the information during the propagation. Compared with RKGE, CKE, MCRRec, PER, HoPKG improves by 4.13% to 25.99% in F_1 respectively, HoPKG improves by 3.33% to 22.57% in AUC respectively. The reason why HoPKG performance is better than CKE, PER, MCRRec and RKGE is that they fail to make good use of the high-order semantic information in the knowledge graph. As shown in Figure 5, Compared with others, HoPKG also performs best in terms of Rec@K, Pre@K. The reason of PER and MCRRec perform not better than others is that they need to manually construct meta-paths. This will cause some deterministic factors. Therefore, the performance is worse when the dataset is sparse.

Moreover, different aggregators will produce different results. Our proposed BiPart aggregator has a certain improvement over GraphSage in both AUC and F_1 . Because it combines 2 types of results and propagates entity information more effectively.

In general, as the propagation layer l increases, the number of nodes adopted increases sharply and the computational complexity also increases accordingly. Through comparative experiments, we found that with the increase of l , the performance increases and reaching a peak when $l=3$, and then there is a decreasing trend. The specific results are shown in Figure 6. The reason is that with the

Table 2. Overall Performance Comparison

Model	AUC	F_1
PER	0.6472	0.5964
MCRRec	0.7137	0.6278
CKE	0.7354	0.6849
RKGE	0.7677	0.7216
RippleNet	0.7886	0.7238
HoPKG-GraphSage	0.7906	0.7302
HoPKG-BiPart	0.7933	0.7514

Figure 5a. Comparison of different models in Top-K recommendation (Rec@K)

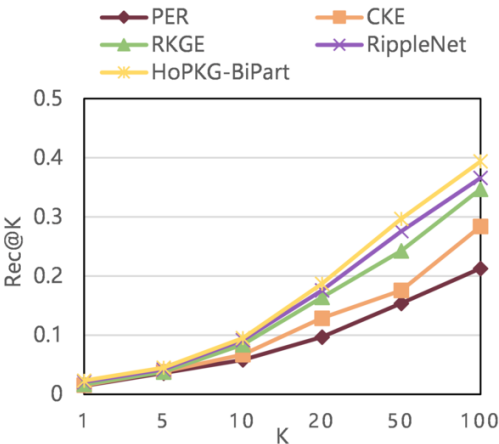


Figure 5b. Comparison of different models in Top-K recommendation (Pre@K)

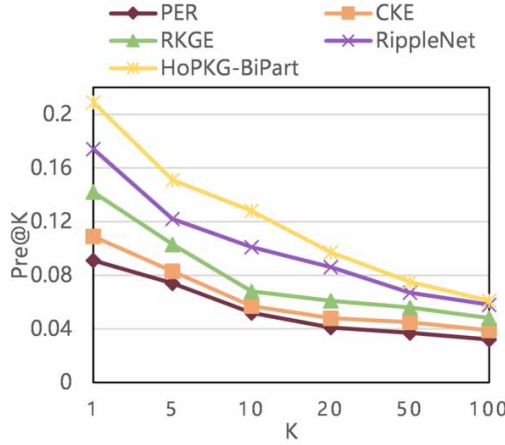
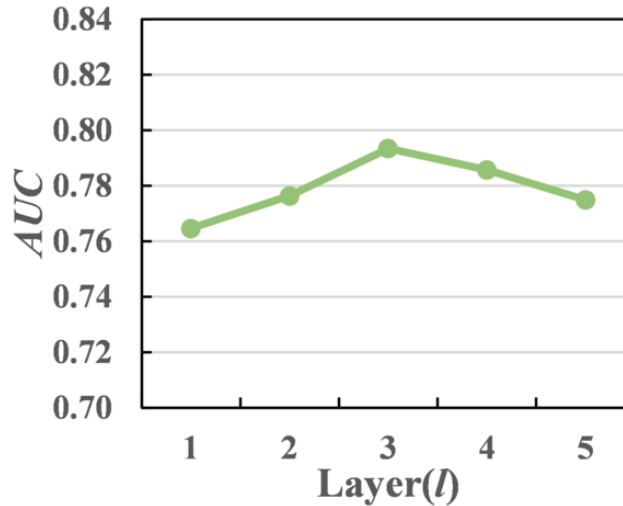


Figure 6a. The Influence of Layer (AUC)



increase of the propagation layer, the entity information is enriched, but too many irrelevant entities are adopted. The result is to increase the computational complexity and reduce the accuracy. In terms of time complexity, as l increases, the time complexity initially increases slowly and increases sharply when $l=4$. This is because as the propagation depth increases, the number of adopted nodes increases sharply, resulting in an increase in computational complexity. In view of the above situation, the performance and time complexity are comprehensively considered, and $l=3$ is selected as the optimal propagation layer in the following experiments.

Moreover, in the target knowledge graph, the size of neighbor sampling directly determines the richness of information in the propagation process. Therefore, how to choose an appropriate neighbor sampling size is significant. In the experiment, we set the neighbor sampling size n as $\{2, 4, 6, 8, 10, 12, 14, 16\}$, repeat each experiment 5 times, and take the average results. The results are shown in Figure 7. AUC reaches the peak when the neighbor sampling size n is 6. F_1 reaches the peak when the

Figure 6b. The Influence of Layer (F_1)

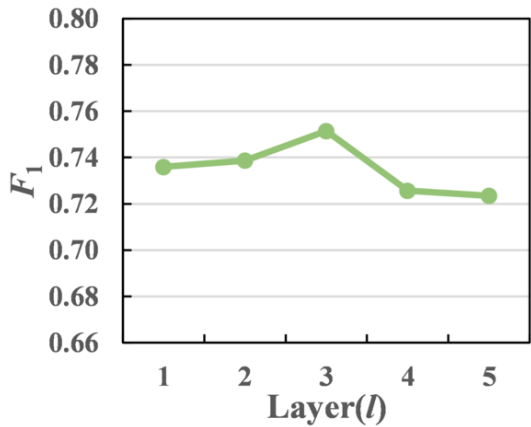
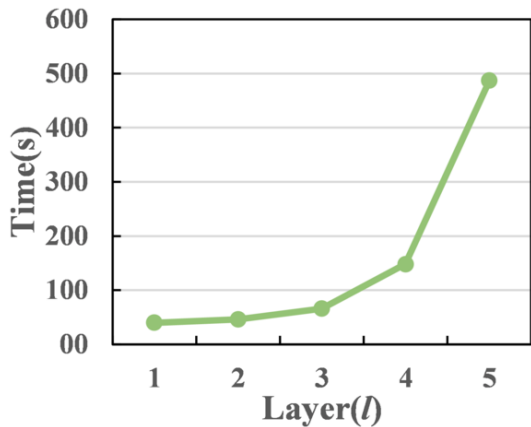


Figure 6c. The Influence of Layer (Time)



neighbor sampling size n is 4. Later, with the increase of neighbor sampling size, AUC and F_1 have shown varying degrees of reduction. The reason for this situation may be because the target dataset is relatively sparse. The number of neighbor nodes of each node is close to the optimal sampling size, so there is a large increase. Later, with the increase of sampling size, the noise also increases, that's why there is a decreasing trend.

5. CONCLUSION

With the advent of the era of big data, the research on knowledge graphs has attracted more and more scholars' attention. Researching the relationship between scholars based on the knowledge graph, and implementing scholar recommendations based on this, has certain significance for grasping academic trends, scientific and technological frontiers, as well as the development of research work and the introduction of talents. Taking this as a starting point, this paper proposes a scholar recommendation method based on the high-order propagation of knowledge graphs. Starting from the history of users' interaction, this method uses the rich information in the knowledge graph to obtain users' preferences, which effectively solves the problem of data sparseness and has certain interpretability. In addition,

Figure 7a. The Influence of Neighbor Sampling Size (AUC)

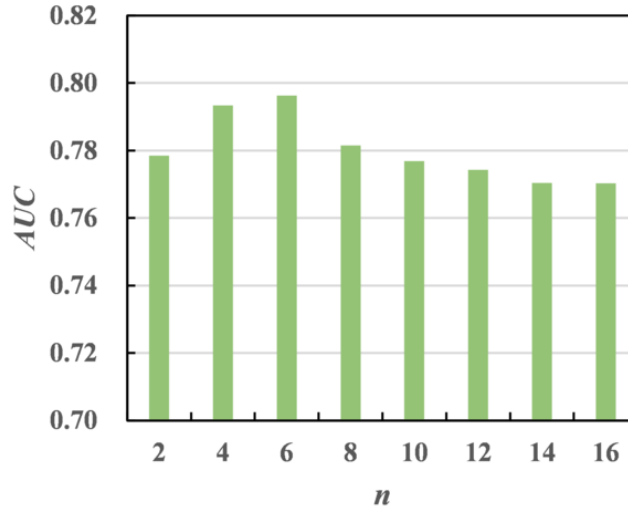
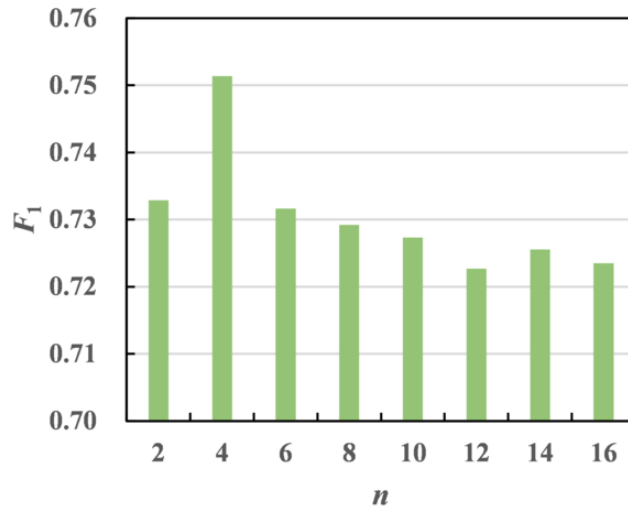


Figure 7b. The Influence of Neighbor Sampling Size (F_1)



the BiPart aggregator proposed in this paper aggregates from two aspects, which can aggregate entity information more effectively. By extracting real scholar data from the SCHOLAT and using this as the target dataset, the experimental analysis shows that the method proposed in this paper has certain advantages compared with some current recommendation methods. Therefore, we can also apply it in other fields, such as restaurant recommendation, music recommendation, book recommendation, etc. But it still has some problems, it does not make good use of the semantic relationship information in the knowledge graph. In future work, we will study and analyze more implicit semantic relationships in the knowledge graph. Moreover, it is hoped that a larger knowledge graph in cyberspace can be cited as a background knowledge source to obtain more semantic information and further enhance the recommendation effect.

ACKNOWLEDGMENT

The works described in this paper are supported by The National Natural Science Foundation of China under Grant Nos. 61802352, 61772211, 61772210, 62072414, 61975187, 62006212, 62102372; The Program for Young Key Teachers of Henan Province under Grant No. 2021GGJS095; The Project of collaborative innovation in Zhengzhou under Grant No. 2021ZDPY0208; The Ph.D. Research Foundation of Zhengzhou University of Light Industry under Grant No. 2017BSJJ052; The Project of Science and Technology in Henan Province under Grant Nos. 202102210178, 222102210030, 212102210418, 222102210027, 212102210104.

ADDITIONAL FUNDING INFORMATION

The publisher has waived the Open Access Publication fee for this article.

REFERENCES

- Gesese, G. A., Biswas, R., & Sack, H. (2019). A comprehensive survey of knowledge graph embeddings with literals: Techniques and applications. *International Workshop on Deep Learning for Knowledge Graphs (DLKG 2019)*, 31-40.
- Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., & He, Q. (2020). A Survey on Knowledge Graph-Based Recommender Systems. *IEEE Annals of the History of Computing*, (1), 1–1.
- Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*(pp. 1024-1034). Curran Associates, Inc.
- Heist, N., Hertling, S., Ringler, D., & Paulheim, H. (2020). Knowledge Graphs on the Web – An Overview. *Knowledge Graphs for eXplainable Artificial Intelligence*, 3–22.
- Hu, B., Shi, C., Zhao, W. X., & Yu, P. S. (2018). Leveraging Meta-path based Context for Top- N Recommendation with A Neural Co-Attention Model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1531-1540). Association for Computing Machinery. doi:10.1145/3219819.3219965
- Indra, R., & Thangaraj, M. (2019). An Integrated Recommender System Using Semantic Web With Social Tagging System. *International Journal on Semantic Web and Information Systems*, 15(2), 47–67. doi:10.4018/IJSWIS.2019040103
- Jadhav, S., & Patil, T. R. (2017). Human Computer Interaction. *International Journal of Advanced Research in Computer and Communication Engineering*, 6(3), 141–142. doi:10.17148/IJARCCCE.2017.6329
- Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015). Knowledge Graph Embedding via Dynamic Mapping Matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (pp. 687-696). Association for Computational Linguistics. doi:10.3115/v1/P15-1067
- Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2021). A Survey on Knowledge Graphs: Representation, Acquisition and Applications. *IEEE Transactions on Neural Networks*, 1–21. PMID:33900922
- Kingma, D. P., & Ba, J. L. (2015). *Adam: A Method for Stochastic Optimization*. arXiv preprint arXiv:1412.6980.
- Kipf, T. N., & Welling, M. (2016). *Semi-Supervised Classification with Graph Convolutional Networks*. arXiv preprint arXiv:1609.02907.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. doi:10.1145/3065386
- Lee, Z.-J., He, H.-M., Su, Z.-Y., Chen, Y., Xiao, J.-Y., & Yang, J.-C. (2020). Design an Online Aquarium with Intelligent Recommend System. In *Proceedings of 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)* (pp. 217-220). IEEE. doi:10.1109/ICBAIE49996.2020.00052
- Li, P., Jiang, Y., Wang, J., & Yin, Z. (2017). Semantic Extension of Query for the Linked Data. *International Journal on Semantic Web and Information Systems*, 13(4), 109–133. doi:10.4018/IJSWIS.2017100106
- Li, P., Li, T., Zhang, S., Li, Y., Tang, Y., & Jiang, Y. (2020). A semi-explicit short text retrieval method combining Wikipedia features. *Engineering Applications of Artificial Intelligence*, 94, 103809. doi:10.1016/j.engappai.2020.103809
- Li, P., Xiao, B., Akram, A., Jiang, Y., & Zhang, Z. (2018). SESLDS: An Extension Scheme for Linked Data Sources Based on Semantically Enhanced Annotation and Reasoning. *International Journal of Intelligent Systems*, 33(2), 233–258. doi:10.1002/int.21926
- Li, P., Xiao, B., Ma, W., Jiang, Y., & Zhang, Z. (2017). A graph-based semantic relatedness assessment method combining wikipedia features. *Engineering Applications of Artificial Intelligence*, 65, 268–281. doi:10.1016/j.engappai.2017.07.027

- Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (pp. 2181-2187). AAAI Press.
- Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., & Tang, J. (2018). DeepInf: Social Influence Prediction with Deep Learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2110-2119). Association for Computing Machinery. doi:10.1145/3219819.3220077
- Sun, Z., Yang, J., Zhang, J., Bozzon, A., Huang, L.-K., & Xu, C. (2018). Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (pp. 297-305). Association for Computing Machinery. doi:10.1145/3240323.3240361
- Tang, X., Wang, T., Yang, H., & Song, H. (2019). AKUPM: Attention-Enhanced Knowledge-Aware User Preference Model for Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1891-1899). Association for Computing Machinery. doi:10.1145/3292500.3330705
- Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., & Guo, M. (2018). RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 417-426). Association for Computing Machinery. doi:10.1145/3269206.3271739
- Wang, H., Zhang, F., Xie, X., & Guo, M. (2018). DKN: Deep Knowledge-Aware Network for News Recommendation. In *Proceedings of the 2018 World Wide Web Conference* (pp. 1835-1844). International World Wide Web Conferences Steering Committee. doi:10.1145/3178876.3186175
- Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., & Guo, M. (2019). Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In *Proceedings of the 2019 World Wide Web Conference* (pp. 2000-2010). International World Wide Web Conferences Steering Committee. doi:10.1145/3308558.3313411
- Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12), 2724–2743. doi:10.1109/TKDE.2017.2754499
- Wu, S., Shen, Q., Deng, Y., & Cheng, J. C. P. (2019). Natural-language-based intelligent retrieval engine for BIM object database. *Computers in Industry*, 108, 73–88. doi:10.1016/j.compind.2019.02.016
- Xu, Q., Qiu, L., Lin, R., Tang, Y., He, C., & Yuan, C. (2021). An Improved Community Detection Algorithm via Fusing Topology and Attribute Information. In *Proceedings of 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 1069-1074). IEEE. doi:10.1109/CSCWD49262.2021.9437681
- Yu, X., Ren, X., Sun, Y., Gu, Q., Sturt, B., Khandelwal, U., Norick, B., & Han, J. (2014). Personalized entity recommendation: a heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining* (pp. 283-292). Association for Computing Machinery. doi:10.1145/2556195.2556259
- Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W.-Y. (2016). Collaborative Knowledge Base Embedding for Recommender Systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 353-362). Association for Computing Machinery. doi:10.1145/2939672.2939673
- Zhao, H., Yao, Q., Li, J., Song, Y., & Lee, D. L. (2017). Meta-Graph Based Recommendation Fusion over Heterogeneous Information Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 635-644). Association for Computing Machinery.

ENDNOTE

¹ www.scholat.com

Pu Li, PhD, the associate professor in Software Engineering College, Zhengzhou University of Light Industry. His researches mainly focus on data intelligence, semantic computing, and semantic search. You can browse the homepage <https://www.scholix.com/superlipu> for more information.

Tianci Li received the B.S. degree in Software Engineering from Zhongyuan University of Technology, Zhengzhou, China, in 2019. He is currently working toward the M.S. degree in Software Engineering with the College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou, China. His research interests include Knowledge Graph, Semantic Analysis and Recommended System.

Xin Wang received the B.S. degree in Software Engineering from Zhengzhou University of Light Industry, Zhengzhou, China, in 2019. He is currently working toward the M.S. degree in Electronic Information with the College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou, China. His research interests include Knowledge Graph, Semantic Analysis and Recommended System.

Suzhi Zhang received the Ph.D. degree from School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. He is currently a professor in Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou, China. His major field of study includes data mining, machine learning, etc. You can browse the homepage <http://soft.zzuli.edu.cn/2020/0609/c5885a214315/page.htm> for more details.

Yuncheng Jiang is a Professor and Pearl River Scholar at the School of Computer Science, South China Normal University, Guangzhou, China. He received his Ph.D. from the Institute of Computing Technology, Chinese Academy of Sciences. His work is published in journals and conferences such as Information Processing & Management, Information Sciences, Fuzzy Sets and Systems, Knowledge-Based Systems, Engineering Applications of Artificial Intelligence, International Journal of Intelligent Systems, International Journal on Semantic Web and Information Systems, Soft Computing, Multimedia Tools and Applications, Cluster Computing, AAAI, and KSEM. His current research interests include semantic search, semantic computing, knowledge graphs, and data science.

Yong Tang received the Ph.D. degree from School of Computer Science, University of Science and Technology of China, Hefei, China. He is currently a professor and doctoral supervisor in School of Computer Science, South China Normal University, Guangzhou, China. His major field of study includes data intelligence, social network, etc. You can browse the homepage <https://www.scholix.com/ytang> for more details.