Expert Systems With Applications 201 (2022) 117078

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Reduce unrelated Knowledge through Attribute Collaborative signal for knowledge graph recommendation

Fulan Qian *, Yuhui Zhu, Hai Chen, Jie Chen, Shu Zhao, Yanping Zhang

Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui University, Hefei, Anhui Province 230039, China
School of Computer Science and Technology, Anhui University, Hefei 230601, China
Information Materials and Intelligent Sensing Laboratory of Anhui Province, China

ARTICLE INFO

Keywords:
Recommender system
Knowledge graph
Collaborative signal
Attention mechanism

ABSTRACT

Knowledge graph (KG), as an auxiliary information, plays an important role in the recommendation system, which effectively solves the sparsity and cold start problems of collaborative filtering algorithms. The recommendation algorithm that introduces the propagation mechanism on the KG has been a great success, it enriches the representation of users and items by aggregating multi-hop neighbors. However, the existing KG-based propagation recommendation algorithm aggregating all entity information cannot guarantee the improvement of recommendation results, because entity information in KG is not all helpful to recommend appropriate items to users. Indiscriminately aggregating the entity information in the neighborhood allows the learned embedding representation to be influenced by its unrelated entities.

In this paper, we propose a new model named **R***educe unrelated* **K***nowledge through* **A***ttribute* **C***ollaborative signal* (RKAC). Compared to other KG-based propagation methods, RKAC offers a new concept of combining item attributes with collaborative signals to reduce information about unrelated entities. Specifically, the initial entity set of users was obtained by collaborative signals and the initial entity set of items was obtained by filtering redundant collaborative signals based on item attributes, and then they were propagated on KG as seeds to acquire multi-hop neighbor entities. Finally, domain entities of different importance were gathered through attention mechanism to obtain more accurate embedding representation of entities. Experimental results on four benchmark datasets of music, book, movie and restaurant show that the AUC of RKAC on CTR prediction increases by 1.4% , 1.3%, 0.8% and 0.5% respectively, compared with the state-of-the-art existing approaches.

1. Introduction

Recommendation system is based on the user's interaction history and characteristics, attributes and context information modeling, to infer user's interest, and recommend the right items for the user (Cai & Zhu, 2019). Among the many recommendation strategies, the classic collaborative filtering (CF) (He et al., 2017; Koren et al., 2009; Weimer et al., 2008; Zhang et al., 2016) algorithm estimates user's interest by analyzing the user's historical interaction records (Chen et al., 2021a, 2021b). However, traditional collaborative filtering algorithms cannot effectively solve the problem of data sparsity and cold start without auxiliary information. As an emerging auxiliary information, KG contains rich item attribute and relation information. Introducing it into the recommendation system as a kind of auxiliary information can effectively improve the problems existing in the CF algorithm.

KG (Sun & Han, 2012) is a heterogeneous network composed of nodes (items and item attributes) and edges (relations). Specifically, each edge in the knowledge graph is represented as a triplet (head entity, relation, tail entity), indicating that the head entity and tail entity are connected by this relation. For example, as shown on the left side of Fig. 1, a triple (Cast away, film. Director, Robert Zemeckis) indicates that Robert Zemeckis is the director of film Cast away. Multiple triples reveal various relationships between items.

However, it is a challenge to integrate heterogeneous information from knowledge graph into recommendation system effectively. The existing fusion methods of knowledge graph and recommendation system can be roughly divided into three types: embedding-based (Cao et al., 2019; Huang et al., 2018; Wang, Zhang, Xie, & Guo, 2018; Wang, Zhang, Zhao, et al., 2019; Xin et al., 2019; Zhang, Yuan, Lian,

* Correspondence to: Qingyuan campus of Anhui University, No. 111, Jiulong Road, Shushan District, Hefei, Anhui, China.
E-mail addresses: qianfulan@hotmail.com (F. Qian), zhuyuhuilz@163.com (Y. Zhu), chber_ahu@hotmail.com (H. Chen), chenjie200398@163.com (J. Chen), zhaoshuzs2002@hotmail.com (S. Zhao), zhangyp2@gmail.com (Y. Zhang).

https://doi.org/10.1016/j.eswa.2022.117078
Received 24 October 2021; Received in revised form 22 January 2022; Accepted 28 March 2022
Available online 4 April 2022
0957-4174/

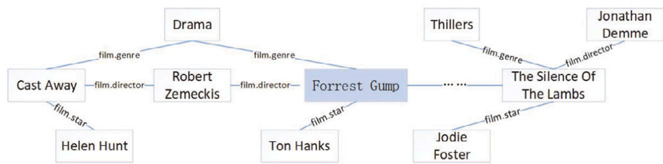

Fig. 1. The actual scene of introducing unrelated entities in MovieLens-20M. *Forrest Gump* is the same genre and director as *Cast Away*, but completely different from *The Silence Of The Lambs*.

Xie, & Ma, 2016), path-based (Hu et al., 2018; Shi et al., 2019; Sun et al., 2018; Wang, Wang, et al., 2019; Yu et al., 2014, 2013) and propagation-based (Sha et al., 2019; Tang et al., 2019; Wang, He, Cao, Liu, & Chua, 2019; Wang et al., 2018; Wang, Zhang, Zhang, et al., 2019; Wang, Zhao, Xie, Li, & Guo, 2019).

Embedding-based algorithms use knowledge graph embedding (Bordes et al., 2013; Ji et al., 2015; Lin et al., 2015; Wang et al., 2014) algorithms to process KG in advance (Wang et al., 2017; Zhu et al., 2019) and embed the learned entity as the item representation in the recommendation. Nevertheless, the entity embedding learned by this method is more suitable for graph applications, such as link prediction, rather than recommendation; Path-based algorithms seriously depend on artificially designed meta-paths or meta-graphs, lack flexibility and scalability; Propagation-based algorithms mainly focus on effectively encoding the knowledge association in KG, and this key collaboration signals hidden in user–item interactions are not well utilized. There are also studies that use the target entity's collaborative signals to propagate on KG to enrich the embedding representation, however, without distinguishing the propagation field of this collaborative signal, the unrelated entity information will be introduced when aggregating the multi-hop neighbor information.

Toy Example. In the actual scenario shown in Fig. 1. If we want to get a better feature representation of the target movie *Forrest Gump* in the KG, the existing method is to represent it through the historical viewing records of users who have watched the movie (collaboration signals). If user A and user B have watched the target movie *Forrest Gump*, and at the same time user A and user B have seen other movies, such as *Cast Away* and *The Silence Of The Lambs*, in which the movie of *Cast Away* and target movie are both *dramas*, which can provide better information for the feature representation of the target movie, while *The Silence Of The Lambs* is far from the target movie in terms of the genre, director, leading role and so on. Furthermore, the gap between entities diffused by *The Silence Of The Lambs* along with the KG and the target film will be even greater. Therefore, if the feature representation of the target movie is only from the collaborative signal, it will bring more unrelated entities.

In order to handle the limitation of above methods of KG-based recommendations, we propose a new end-to-end neural recommendation method, RKAC, which can effectively reduce the unrelated entities when aggregating multi-hop neighbors. This paper mainly explores the association behind the item by combining the item attribute in KG with the collaborative signals displayed in user–item interaction, filters the unrelated auxiliary knowledge information in the process of aggregating neighbor entities, to obtain the accurate representation of users and items. Specifically, first of all, through collaborative signals and item attribute information for the user and the item of the initial propagation collection, and then spread the collection on KG to explore multi-layer potential associated knowledge information, using attention mechanism to generate each layer entity weighted embedding, finally gathering from different propagation layer embedding in order to obtain more accurate representation of a user and a item.

Our method applies to four baseline data scenarios: music, books, movies and restaurants. Experimental results show that our method is superior to the several state-of-the-art existing methods. In conclusion, our contribution is as follows:

- We propose a new end-to-end model, RKAC, which extracts collaborative signals and attribute information in KG to obtain a diffusion model with high attribute correlation, which is used to mine users' higher-order interests and preferences on the KG.
- In the module of attribute collaborative propagation and knowledge graph propagation of the model, we improve the recommendation effect through reducing unrelated entities.
- We conduct experiments in four real-world recommendation scenarios, and the experimental results demonstrate that our approach is superior to existing advanced methods.

2. Related work

Based on the idea of embedding propagation, the common method is based on Graph Neural Networks (GNNS) (Dwivedi et al., 2020; Velickovic et al., 2018). These methods provide an embedding representation of the entity, predict user's preference, and make effective recommendations mainly through two methods of aggregating all entities information and aggregating entities information with collaborative signals. The following will introduce the research progress of related work from these two aspects.

2.1. Aggregating all entities information

These methods aggregate information from its multi-hop neighbors to update the embedding representation.

Ripplenet (Wang et al., 2018) propagates users' historical preferences along with the link on KG to mine users' deeper interests, but this method fails to effectively represent the importance of relations. KGCN (Wang, Zhao, Xie, Li, & Guo, 2019) and KGNN-LS (Wang, Zhang, Zhang, et al., 2019) are end-to-end models based on graphical neural networks (GNNs). The key idea is to integrate multi-hop neighbors into the representation by using information aggregation schemes. AKGE (Sha et al., 2019) first automatically extracts the high-order subgraphs of user–item pairs with rich link semantics and then encodes the subgraphs through the proposed attention graph neural network to learn accurate user preferences. However, the above methods fail to effectively utilize the collaborative signal in user–item interaction, resulting in the deficiency of item embedding.

2.2. Aggregating entities information with collaborative signals

These methods enrich entity representation by using multi-hop neighbor information and user–item interaction information (Wang et al., 2021).

KGAT (Wang, He, Cao, Liu, & Chua, 2019) integrates *user–item bipartite graph* (UIG) and *knowledge graph* (KG) into a *collaborative knowledge graph* (CKG) to supplement entity embedding. KGIN (Wang et al., 2021) considers the user–item relation at a more fine-grained intention level, regards the relation path as an information channel, and embeds each channel into the representation vector. The entity to be merged in AKUPM (Tang et al., 2019) is initialized as the item that the user clicks and then propagates along with the relation in the KG from near to far to introduce rich entities. Then weighted entities are aggregated to the user to show the user's attention to the input item. CKAN (Wang et al., 2020) combines collaborative signals and knowledge association naturally, uses KG to propagate and extract users' potential interests, and iteratively injects them into users' features with attention bias. In particular, CKAN does not make full use of the knowledge association in KG. There are two main points: (1) It regards KG as a directed graph, which makes the connected entities fair to exert influence on each other along with the link and lose some effective neighborhood information. (2) It only propagates collaborative signals on KG to aggregate neighbor information without considering the attribute characteristics of the target item itself. It introduces the interference of unrelated entity

Table 1
Datasets and evaluation indicators used in related work.

Model	Datasets									Indicators					
	Last.FM	Book-Crossing	MovieLens-1M	MovieLens-20M	Dianping-Food	Bing-News	Yelp	Amazon-book	Alibaba-iFashion	AUC	ACC	F1	recall	ndcg	hit
Ripplenet		✓	✓			✓				✓	✓		✓		
KGCN	✓	✓		✓						✓		✓	✓		
KGNN-LS	✓	✓		✓	✓					✓			✓		
AKGE	✓		✓				✓							✓	✓
KGAT	✓						✓	✓					✓	✓	
KGIN	✓							✓	✓				✓	✓	
AKUPM		✓	✓							✓	✓		✓		
CKAN	✓	✓		✓	✓					✓		✓	✓		

information beyond multi-hop in the process of learning target item representation.

All the above methods show that the entity information of multi-hop neighbors plays an important role in improving the efficiency of the recommendation system. However, None of the above methods take into account both the collaborative signals and the item attributes on the KG, which is equivalent to the undifferentiated aggregation of neighbor information from different neighborhoods in the process of enriching entity representation and finally have an effect on embedding of users and items. Our method can alleviate this effect by reducing unrelated auxiliary entity information. The Table 1 summarizes the datasets used by these methods and their metrics.

3. Problem formulation

We describe the recommendation problem based on KG as follows.

User–item interaction. We assume that existing user $u \in U = \{u_1, u_2, \ldots, u_m\}$ and item $v \in V = \{v_1, v_2, \ldots, v_n\}$, the user–item interaction matrix based on users' implicit feedback is defined as $\mathbf{Y} \in \mathbb{R}^{m\times n}$, where m is the number of users, n is the number of items, $y_{uv} = 1$ means that user u has interacted with item v, such as clicking, watching, buying, etc., otherwise, it is 0. $y_{uv} = 0$ does not mean that the user u does not like item v. It is just that the user u cannot touch item v for some reason, user u may like item v.

Knowledge Graph. We obtain a heterogeneous graph $G = \{h, r, t | h, t \in E, r \in R\}$ with item and its attributes as entities, where E and R are the set of entities and relations, respectively. For each triplet (h, r, t), it means that there exists relation r which links entity h and tail entity t. For example, (Steven Spielberg, director, Schindler's list) describes the fact that Steven Spielberg directed the movie Schindler's list. The mapping between item and KG can be aligned through the set $A = \{(v, \mathrm{e}) | v \in V, \mathrm{e} \in E\}$, and (v, e) represents the entity e in the knowledge graph corresponding to item v. For example, in movie recommendation, the item "Schindler's list" also appears in the knowledge graph as an entity with the same name. In this way, KG can describe items and provide supplementary information for interactive data. It is worth noting that (1) G is treated as an undirected graph because connected entities influence each other along with the links. (2) Through knowledge association and user–item interactions in G, the collaborative signals are propagated (Wang et al., 2018);

Task Description. According to the user–item interaction matrix $\mathbf{Y}$ and the knowledge graph G, predicting the probability that user u will select items that have not interacted with each other. Our primary goal is to learn a prediction function $\hat{y}_{uv} = \Gamma(u, v|\theta)$, where $\hat{y}_{uv}$ is the prediction probability, θ represents the model parameters of function Γ.

4. The RKAC model

We propose a model named **R**educe *unrelated* **K**nowledge *through* **A**ttribute **C**ollaborative *signal* (RKAC), shown in Fig. 2 as a model framework. Specifically, RKAC includes four modules: (1) Attribute collaborative propagation. We get two sets, the initial entity set of user was obtained by collaborative signals and the initial entity set of item was obtained by filtering redundant collaborative signals based on item attributes from the first low order propagation on KG. (2) Knowledge graph propagation. The user initial entity set and the item initial entity set obtained in the previous module are used as seeds on KG, and the second propagation was carried out on KG respectively to capture more potential neighbors to enrich the embedding of target entities. (3) Attention embedding. The attention mechanism is further used to distinguish the importance of different neighbors in the obtained domain neighbor embedding representation. By aggregating domain neighbors with different importance, a more accurate embedding representation of user and item is obtained. (4) Prediction. We aggregate the representations of users and items from different propagation layers and finally output the predicted click probability.

The formal description of the above steps is presented in Algorithm 1. Where $N(e_u^l)$ means the embedding representation of user u's lth neighbor entity. Note that Algorithm 1 traverses all possible user–item pairs (line 3). Next we will describe our model in more details

Algorithm 1: RKAC algorithm

Input: Interaction matrix $\mathbf{Y}$; knowledge graph G;
Output: Prediction function $\Gamma(u, v|\theta)$;
1 Initialize all parameters;
2 **while** *RKAC not converge* **do**
3   **for** (u, v) *in* $\boldsymbol{Y}$ **do**
4     $\xi_u^0 \leftarrow$ calculate user u initial entity sets;
5     $\delta_v^0 \leftarrow$ calculate item v collaborative propagation set;
6     $\omega \leftarrow$ calculate item v attribute propagation set;
7     $\xi_v^0 \leftarrow \delta_v^0 \cap \omega$; calculate item v initial entity sets;
8     $\xi_u^L, S_u^L \leftarrow$ Sample_neighbor(ξ_u^0, L, G);
9     $\xi_v^L, S_v^L \leftarrow$ Sample_neighbor(ξ_v^0, L, G);
10   **for** *l=1,...,L* **do**
11   **for** $e \in \xi_u^l$ **do**
12       $N(e_u^{l-1}) \leftarrow \sum_{(h,r,t)\in S_u^l} \tilde{\gamma}(e^h, r)e^t$
13   **for** $e \in \xi_v^l$ **do**
14       $N(e_v^{l-1}) \leftarrow \sum_{(h,r,t)\in S_v^l} \tilde{\gamma}(e^h, r)e^t$;
15   $e_u \leftarrow e_u^L$,$e_v \leftarrow e_v^L$;
16   Calculate predicted probability $\hat{y}_{uv} = e_u^\top e_v$;
17   Update parameters by gradient descent;
18 **return** Γ;
19 **Function** Sample_neighbor(x, L, G)**:**
20   $n^0 \leftarrow x$;
21   **for** $l = 1, \ldots, L$ **do**
22   **for** $y \in n^{l-1}$ **do**
23       $\xi^l \leftarrow n^l \cup \{$ z | sampled neighbors of y in G' $\}$
24       $S^l \leftarrow (h, r, t)^l \cup \{$ (h', r', t') | sampled neighbor triplets of y in G' $\}$
25   **return** $\{\xi^i\}_{i=1}^L, \{S^i\}_{i=1}^L$

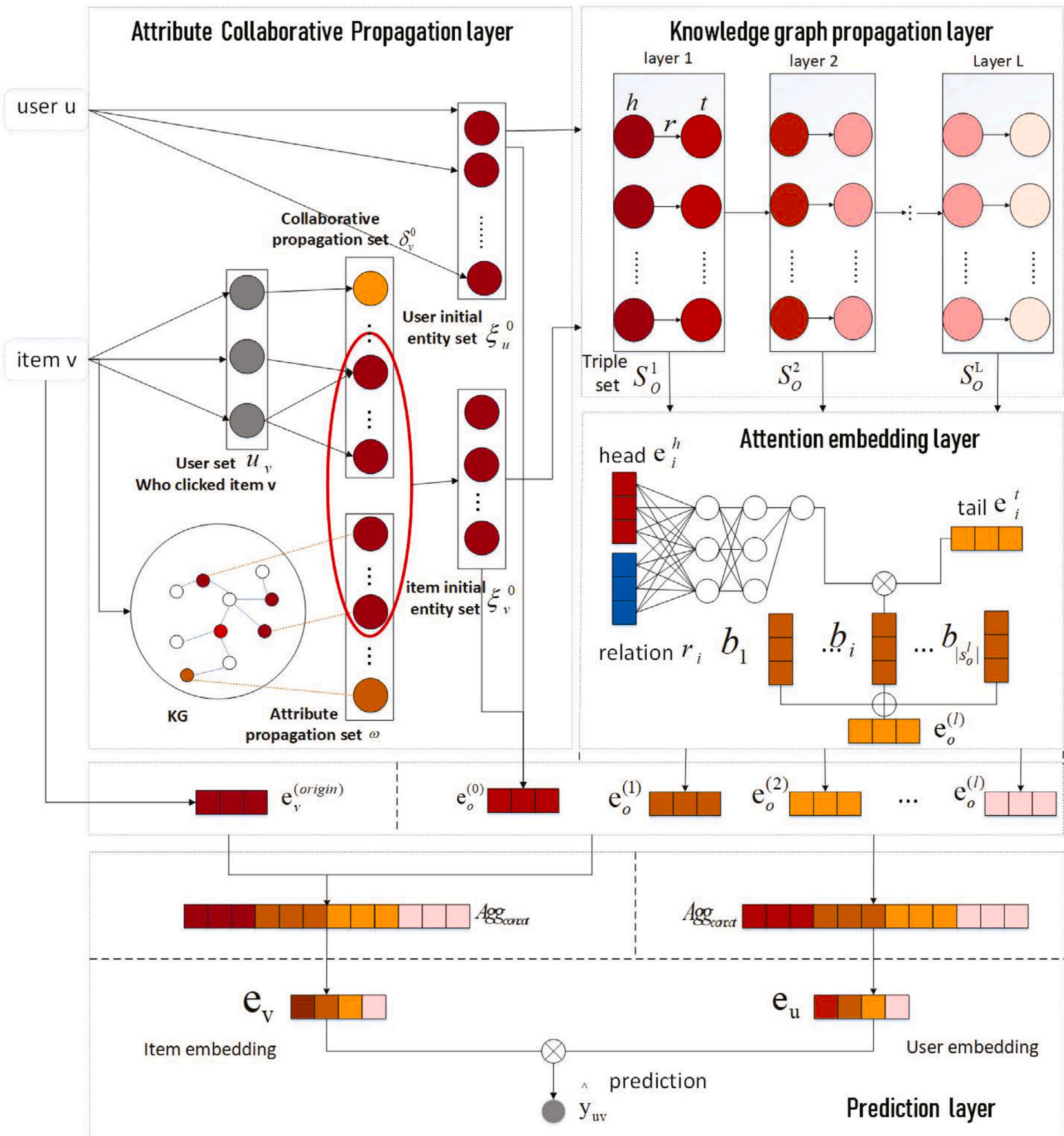

Fig. 2. Illustration of the proposed RKAC model, which consists of four modules: attribute collaborative propagation layer, knowledge graph propagation, attention embedding layer and prediction layer. the initial set of the target item in the attribute collaborative layer is the overlapping part of the item collaboration set and the item attribute set, as shown in the figure with a red ellipse.

4.1. Attribute collaborative propagation layer

Theoretically, the user's historical interaction items reflect the user's preferences to a certain extent. Here, we express user through its interaction items. Historical interactive items are converted to the initial seed set propagated in KG through alignment between items and entities. The initial entity set ξ_u^0 for user u is defined as follows:

$$\xi_u^0 = \{\mathrm{e}|(v,\mathrm{e}) \in A \ and \ v \in \{v|y_{uv} = 1\}\} \tag{1}$$

Similarly, users who have interacted with item v can contribute to the item's feature because of their similar behavior preferences. We define a collaborative neighbor as a user who has already interacted with the same item v and use the collaborative neighbor's historical interactive item as a collection δ_v^0 of collaborative propagation of the item v, formulated as follows:

$$V_v = \{v_u|u \in \{u|y_{uv} = 1\} \ and \ y_{uv_u} = 1\} \tag{2}$$

The corresponding alignment set on KG is as follows:

$$\delta_v^0 = \{\mathrm{e}|(v_u,\mathrm{e}) \in A \ and \ v_u \in V_v\} \tag{3}$$

At the same time, the highly related items propagated by item v's neighbors on the KG can contribute to the item's feature representation in terms of attribute similarity. We formulate the kth level attribute propagation set ω_v^k from item v propagated on the KG as follows:

$$\omega_v^k = \{t|(h,r,t) \in G \ and \ h \in \omega_v^{k-1}\} \ k = 1,2,\ldots,L \tag{4}$$

Where $\omega_v^0 = \{\mathrm{e}|(v,\mathrm{e}) \in A\}$, ω_v^0 is the corresponding entity of item v on the KG. Notice that, since we want to get other items with the same attribute association degree as the target item, the tail entity that is propagated once is the attribute value of the item, two propagations may get other items with the same attribute association degree so that $2k$th propagations can get the attribute propagation set ω of the k layers.

$$\omega = \{\omega_v^0, \omega_v^1, \dots, \omega_v^k\} \tag{5}$$

The intersection of formula formula (3) and formula formula (5) give the collaborative propagation set ξ_v^0, the initial entity set of the item, which reflects both user's preference and the attribute feature of the item. Formulated as follows:

$$\xi_v^0 = \{\mathrm{e} | \mathrm{e} \in \delta_v^0 \ and\ \mathrm{e} \in \omega\} \tag{6}$$

The attribute collaborative propagation layer not only explicitly codes the potential first-order interactive information into the initial entity set, but also enhances the representation of users and items and improves the recommendation effect by extracting the attribute characteristics of items to control the propagation trend of preferences over KG.

4.2. Knowledge graph propagation layer

The degree of association of connected entities in KG is always high. By propagating the initial set of entities for the user and the item along with the links in KG, we can obtain extended entity sets and triples set at different distances, which can effectively enrich the latent vector representation of the user and the item. The entity set ξ_o^l definitions for user u and item v are recursively defined as:

$$\xi_o^l = \{t | (h, r, t) \in G\ and\ h \in \xi_o^{l-1}\}, l = 1, 2, \dots, L \tag{7}$$

Where l represents the distance from the initial entity set, subscript o is a uniform placeholder for the symbol u and v since the calculation and formulation for user and item in this paper are similar in many scenarios. Given the definition of the entity set, the lth triplet set S_o^l of user u and item v is as follows:

$$S_o^l = \{(h, r, t) | (h, r, t) \in G\ and\ h \in \xi_o^{l-1}\}, l = 1, 2, \dots, L \tag{8}$$

In the KG, adjacent entities can be regarded as an intuitive extension of user's preference and item features. As shown in Fig. 2, the initial entity set obtained by attribute collaborative propagation can be used as seed to spread in KG layer by layer from near to far, so as to obtain knowledge-based high-order interaction information and attribute feature information with high correlation degree, which can effectively improve the ability of the model to represent users and items with latent vectors.

4.3. Attention embedding layer

Because an entity can be contained in multiple KG triples, when different head entities and relation join tail entities, they have different latent semantic representations. For example, "Farewell My Concubine" and "Jingke assassinates the king of Qin" are both directed by Chen Kaige. However they are not related in terms of film themes, so it is necessary to distinguish the different semantic information contained in different head entities and relations. Here we use attention embedding to measure the different attention weights of tail entities. For the ith (h, r, t) triplet of layer l, we can get the attentive embedding of tail entity $\mathbf{a_i}$ as follows:

$$\mathrm{a_i} = \gamma(\mathrm{e_i^h}, \mathrm{r_i})\mathrm{e_i^t} \tag{9}$$

Where $\mathrm{e_i^h} \in \mathbb{R}^\mathrm{d}$, $\mathrm{r_i} \in \mathbb{R}^\mathrm{d}$ and $\mathrm{e_i^t} \in \mathbb{R}^\mathrm{d}$ are the embedding representations of the head entity h, relation r and tail entity t in the ith triple, respectively, d is the dimension of representations. We implement $\gamma(\mathrm{e_i^h}, \mathrm{r_i})$ through attention mechanism (Vaswani et al., 2017), and the formula is as follows:

$$z_0 = \mathrm{ReLU}(\mathrm{W_0}(\mathrm{e_i^h} \parallel \mathrm{r_i}) + \mathrm{b_0}) \tag{10}$$

$$\gamma(\mathrm{e_i^h}, \mathrm{r_i}) = \sigma(\mathrm{W_2}\mathrm{ReLU}(\mathrm{W_1}z_0 + \mathrm{b_1}) + \mathrm{b_2}) \tag{11}$$

We use ReLU as the nonlinear activation function (Hahnloser et al., 2000), Sigmoid as the final activation function (Han & Moraga, 1995), $\parallel$ as a concatenation operation, W and b as trainable weight matrices and deviations, and their different subscripts indicate that they are parameters of different layers. After that, we use softmax function to normalize the coefficients of all triples:

$$\gamma(\mathrm{e_i^h}, \mathrm{r_i}) = \frac{\exp(\gamma(\mathrm{e_i^h}, \mathrm{r_i}))}{\sum_{(h', r', t') \in S_o^l} \exp(\gamma(\mathrm{e_i^h}, \mathrm{r_i}))} \tag{12}$$

Where S_o^l is the set of triples at the lth level of users or items, and the attention weight is used to quantify the importance of adjacent tail entities under different conditions. Finally, we get the representation of layer lth triplet set of user u or item v:

$$e_o^{(l)} = \sum_{i=1}^{|S_o^l|} \mathrm{a_i^{(o)}}, \ l = 1, 2, \dots, L \tag{13}$$

$|S_o^l|$ is the number of triples in set S_o^l. It is worth noting that item v has the corresponding original related entity representation, but user u does not. However, the entities in their initial entity set are strongly related to the original user and item. Therefore, we add the representation of the initial entity set to the user and item representation, and the item takes the original related entity representation into account.

$$\begin{aligned} \mathrm{e_o^{(0)}} &= \frac{\sum_{\mathrm{e} \in \xi_o^0} \mathrm{e}}{|\xi_o^0|} \\ \mathrm{e_v^{origin}} &= \sum_{\mathrm{e} \in \{\mathrm{e} | (\mathrm{e}, v) \in A\}} \mathrm{e} \end{aligned} \tag{14}$$

After the knowledge graph embedding layer, we can get the multi-layer embedding representation of user u and item v, which is formulated as follows:

$$\begin{aligned} \varphi_u &= \{\mathrm{e}_u^0, \mathrm{e}_u^1, \dots, \mathrm{e}_u^L\} \\ \varphi_v &= \{\mathrm{e}_v^{origin}, \mathrm{e}_v^0, \mathrm{e}_v^1, \dots, \mathrm{e}_v^L\} \end{aligned} \tag{15}$$

4.4. The prediction layer

We use the *Concat aggregator* (Hamilton et al., 2017) to aggregate the multi-layer embedded representations into a single vector of users and items. Concat first connects two representations before applying the nonlinear transformation.

$$agg_{concat}^{(o)} = \sigma(\mathrm{W_3} \cdot (\mathrm{e}_o^{i_1} \parallel \mathrm{e}_o^{i_2} \parallel \cdots \parallel \mathrm{e}_o^{i_n}) + \mathrm{b_3}) \tag{16}$$

In the formula formula (16), W and b are trainable parameters and biases, and we use $Sigmoid$ as the nonlinear function. Finally, the embedding representation of user vector e_u and item vector e_v can be obtained, and the inner product of the two vectors can be used to predict the user's preference score for the item. Specific details follow the work in Wang et al. (2020).

$$\hat{y}_{uv} = \mathrm{e}_u^\top \mathrm{e}_v \tag{17}$$

4.5. Loss function

In order to improve the computational efficiency, we use the negative sampling strategy in training. The loss function is as follows:

$$\begin{aligned} L = \sum_{u \in U} (&\sum_{v : y_{uv} = 1} J(y_{uv}, \hat{y}_{uv}) \\ &- \sum_{v \in \{v | (u,v) \in P\}} J(y_{uv}, \hat{y}_{uv})) + \lambda \parallel \theta \parallel_2^2 \end{aligned} \tag{18}$$

Where J is the cross entropy loss and P is the negative sampling of user u in the training set. In order to balance the number of positive and negative samples and ensure the effect of model training, each user has the same number of negative samples and positive samples. $\theta = \{\mathrm{E}, \mathrm{R}, W_i, b_i, \forall i \in \{0,1,2,3\}\}$ is the model parameter, E and R is the embedding table of all entities and relations, and λ is the hyper-parameter controlling L2-regularizer $\lambda \parallel \theta \parallel_2^2$.

5. Experiment and analysis

In this section, we evaluate the proposed RKAC model and conduct experiments in four real scenarios: music, book, movie and restaurant recommendations.The goal is to answer three research questions:

• **RQ1**: Does our proposed RKAC outperform the state of-the-art KG-based propagation recommendation methods?

• **RQ2**: How do different choices of hyper-parameters

(the depths of propagation layers, size of triple set.) affect the performance of RKAC?

• **RQ3**: How do our proposed the method construction of KG, the propagation of attribute association collaborative signals and attentive aggregation affect the performance of RKAC, respectively?

5.1. Datasets

We use the following four datasets: Last.FM, Book-Crossing, Movielens-20M and Dianping Food. Each dataset contains user–item interaction information and corresponding knowledge associations (presented in KG triples). Details for each dataset are as follows:

• **Last.FM**[1] is provided by last.fm online music system, which contains musicians listening information from a set of 2000 users.

• **Book-Crossing**[2] contains a rating of 1 million books in the book-crossing community (from 0 to 10).

• **MovieLens-20M**[3] contains about 20 million explicit ratings (ranging from 1 to 5) on the MovieLens website. It is a widely used benchmark dataset in movie recommendations.

• **Dianping-Food**[4] is provided by dianping.com, which includes more than 10 million interactions between about 2 million users and 1000 restaurants (including clicks, purchases, and adding to favorites)

The Last.FM, Book-Crossing, and MovieLens-20M datasets contain explicit feedback. We convert them into implicit feedback, where 1 represents the user's positive evaluation of the item, which is a positive sample. The score threshold of Movielens-20M is 4, while Last.FM and Book-Crossing have no threshold due to their sparsity. In addition, we randomly select the items that each user does not pay attention to, marked as 0, as his/her negative sample, and the number is equal to his/her positive sample size.

We used Microsoft satori[5] to build the KGs of Movielens-20M, Book-Crossing and Last.FM datasets. Following the works in Wang, Zhang, Zhang, et al. (2019), Wang, Zhao, Xie, Li, and Guo (2019) and Yu et al. (2014), first select a subset of triplet format with confidence greater than 0.9 from the whole KG, align and match the tail of the triplet with Satori IDs of all valid films, books and musicians collected, and exclude items with multiple matched or mismatched entities. For the Dianping-Food dataset, its KG is built by the internal toolkit of the group of Meituan-dianping (Wang, Zhang, Zhang, et al., 2019). The detailed statistics of the four datasets are shown in Table 2.

Table 2
Statistics of four datasets: Last.FM (Music), Book-Crossing (Books), MovieLens-20M (Movies), Dianping-Food (Restaurants). (# means 'the number of').

	Music	Book	Movie	Restaurant
#users	1,872	17,860	138,159	2,298,698
#items	3,846	14,967	16,954	1,362
#interactions	42,346	139,746	13,501,622	23,416,418
#entities	9,366	77,903	102,569	28,115
#relations	60	255	32	7
#KG triples	15,518	151,500	499,474	160,519

5.2. Baselines

We compare the proposed RKAC model with three types of KG-based recommendation methods: embedding-based method (CKE), path-based method (PER) and propagation-based method (RippleNet, KGCN, KGNN-LS, KGAT, CKAN).

• **CKE** (Zhang, Yuan, Lian, Xie, & Ma, 2016) is a typical embedded model, which combines CF module with structural, textual and visual items knowledge embedding in a unified Bayesian framework.

• **PER** (Yu et al., 2014) is a typical path-based method, which regards KG as a heterogeneous information network and extracts the potential features based on meta-path to represent the connectivity between users and items.

• **RippleNet** (Wang et al., 2018) is an advanced model based on propagation, which propagates users' potential preferences in KG to enrich users' representation.

• **KGCN** (Wang, Zhao, Xie, Li, & Guo, 2019) is another new model based on propagation, which extends the non-spectral GCN method to KG by selectively and biasedly aggregating the neighborhood information, and can learn the structural and semantic information of KG, as well as the personalized and potential interests of users.

• **KGNN-LS** (Wang, Zhang, Zhang, et al., 2019) is another state-of-the-art propagation-based KG model, which transforms heterogeneous KG into user-specific weighted graphs, and uses label smoothness regularization to calculate personalized item embedding in graph neural network.

• **KAGT** (Wang, He, Cao, Liu, & Chua, 2019) is also the most advanced model based on propagation, which combines UIG and KG to form a unified graph, called CKG. In the process of propagation, an attention mechanism is used to distinguish the importance of neighbors.

• **CKAN** (Wang et al., 2020) is also the state-of-the-art propagation-based model, which combines collaborative signals and knowledge association in a natural way, propagates and extracts users' potential interests on KG, and then iteratively injects them into users' features with attention bias.

5.3. Parameter setting

For each dataset, the ratio of the training set, evaluation set and test set is 6:2:2. Each experiment is repeated 5 times, and the average performance is reported. We evaluate our algorithm in the experimental scenario of click prediction (CTR). We use the trained model to predict each interaction in the test set, and use AUC and F1 to evaluate CTR prediction. The default Xavier initializer (Glorot & Bengio, 2010) to initialize the model parameters. Adam is used to optimize all models, other settings follow the work in (Wang et al., 2020).

We implement our RKAC model in PyTorch,[6] and use grid search to confirm the best settings of each method. The learning rate is adjusted between $\{10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}\}$, the normalization coefficient $L2$ is searched in $\{10^{-3}$, 10^{-4}, $10^{-5}, 10^{-6}, 10^{-7}\}$, and the number of triple set for users and items is adjusted from {8, 16, 32, 64}, respectively. The number of layers of the attention network is adjusted from {1, 2, 3,

1 https://grouplens.org/datasets/hetrec-2011/
2 http://www2.informatik.uni-freiburg.de/cziegler/BX/
3 https://grouplens.org/datasets/movielens/
4 https://www.dianping.com/
5 https://searchengineland.com/library/bing/bing-satori
6 https://pytorch.org

Table 3
AUC and F1 results in CTR prediction. The imp-avg represents the average improvement of all baselines. bold indicates the best result on different datasets. -X% in the parentheses means the difference between our result and the corresponding algorithm. Statistically significant improvement t-test with p = 0.1.

Model	Last.FM		Book-Crossing		MovieLens-20M		Dianping-Food	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
CKE (Wang et al., 2020)	0.747 (−10.7%)	0.674 (−12.6%)	0.676 (−11.4%)	0.623 (−8.2%)	0.927 (−5.8%)	0.874 (−7.3%)	0.812 (−7.9%)	0.741 (−8.1%)
PER (Wang et al., 2020)	0.641 (−24.9%)	0.603 (−21.8%)	0.605 (−20.7%)	0.572 (−15.7%)	0.838 (−14.8%)	0.792 (−16.0%)	0.766 (−13.2%)	0.697 (−13.5%)
RippleNet (Wang et al., 2020)	0.776 (−9.1%)	0.702 (−9.0%)	0.721 (−5.5%)	0.647 (−4.7%)	0.976 (−0.8%)	0.927 (−1.7%)	0.863 (−2.2%)	0.783 (−2.9%)
KGCN (Wang et al., 2020)	0.802 (−6.0%)	0.708 (−8.2%)	0.684 (−10.3%)	0.631 (7.0%)	0.977 (−0.7%)	0.930 (−1.4%)	0.845 (−4.2%)	0.774 (−4.0%)
KGNN-LS (Wang et al., 2020)	0.805 (−5.7%)	0.722 (−6.4%)	0.676 (−11.4%)	0.631 (7.0%)	0.975 (−0.9%)	0.929 (1.5%)	0.852 (−3.4%)	0.778 (−3.5%)
KGAT (Wang et al., 2020)	0.829 (−2.9%)	0.742 (−3.8%)	0.731 (−4.1%)	0.654 (−3.6%)	0.976 (−0.8%)	0.928 (−1.6%)	0.846 (−4.1%)	0.785 (−2.6%)
CKAN (Wang et al., 2020)	0.842 (−1.4%)	0.769 (−0.3%)	0.753 (−1.3%)	0.673 (−0.8%)	0.976 (−0.8%)	0.929 (−1.5%)	0.878 (−0.5%)	0.802 (−0.5%)
RKAC	**0.854**	**0.772**	**0.763**	**0.679**	**0.984**	**0.943**	**0.882**	**0.806**

4}, and the dimension of the hidden layer is set to the same dimension of embedding size. In addition, we find that the best parameter setting for attribute propagation is 3.

5.4. Experimental results (RQ1)

The comparison results of different methods on the four datasets are shown in Table 3. The best results in all comparison methods follow the papers in Wang et al. (2020). Our method RKAC is much better than the most advanced baseline in term of the results of CTR prediction. According to the experimental results, we can draw the following conclusions:

(1) As can be seen from the Table 3, our method performs best on all four datasets. Specifically, as far as AUC is concerned, RKAC is superior to the state-of-the-art baselines $w.r.t.$ AUC by 1.4%, 1.3%, 0.8% and 0.5% in the four datasets, which are Last.FM, Book-Crossing, Movielens-20 m and Dianping-Food, respectively; As far as index is concerned, it is better than the best in the baseline by 0.3%, 1.3%, 1.5% and 0.5%, respectively.

(2) The performance of all models on the Book-Crossing dataset is worse than that on the other three datasets. The main reason is that the average user feedback on the Book-Crossing dataset is less than that on other datasets, that is to say, the other three datasets have more abundant interactive information. Therefore, there is not enough information on the Book-Crossing dataset for the model to understand the user's interests.

(3) It can be seen from the experimental results that the propagation-based method is better than the embedding-based and path-based methods in all KG-based methods, which shows the importance of spreading user's preferences and aggregating multi-hop neighbor information in KG. The poor performance of embedding-based (CKE) may be due to the unavailability of textual and visual data. At the same time, the performance of embedding-based (PER) is lower than other methods because it is difficult to find the best meta-paths. Compared with RippleNet, the importance of knowledge aware attention module is proved. The results compared with KGCN and KGNN-LS show the effectiveness of explicit modeling collaboration signals. Compared with KGAT and CKAN results, the effectiveness of attribute-based collaborative knowledge association and modeling of high-order neighbor information on KG is obtained.

5.5. Parameters sensitive study (RQ2)

In this section, we analyze the influence of hyperparameters on the performance of the proposed model. First, in Section 5.5.1, we investigate the impact of the number of layers on model accuracy. In Section 5.5.2, we discussed the impact of size of triple set.

5.5.1. Impact of the depths of layers

As shown in Table 4, we change the depth of knowledge aware propagation to study the performance changes in RKAC and explore more related items connected by KG triples, which can deepen the understanding of user's interests. RKAC-1 considers first-order connectivity, while RKAC-2 and above explore more potentially high correlation information. We can observe that music, books, movies and restaurants have the best propagation layer depth of 2, 2, 1 and 4, respectively. The possible reason for this phenomenon is that reasonable deep propagation can provide rich knowledge information. In contrast, long-distance propagation can not only provide high-quality knowledge information, but also bring some negative information.

5.5.2. Impact of size of triple set

We change the size of the triple set of users and items to explore their impact on RKAC. Here we select a small dataset (Last.FM) and a large dataset (Movielens-20M) for the experiment. Tables 5 and 6 are the experimental results, respectively. It can be found from the experimental results that increasing the size of the triple sets of users and items on music datasets can improve performance, but if the triple sets are too large, the results will decrease. When the size of the triple sets of users and items is taken as 16, the performance is the best. For movie datasets, the performance of the user triplet is the best when 64, while the size of the item triplet has no impact. One possible reason is that for different datasets, the difference between the number of initial entity sets of users and the initial entity sets of the item is different, which affects the number of triples that can be associated, thus affecting performance.

5.6. Performance comparisons with variants (RQ3)

In order to explore the influence of the method construction of KG, the propagation of attribute association collaborative signals and attentive aggregation on RKAC, we conduct three variants on the model.

- $RKAC_{(directed)}$: Only consider the impact of the model built in a directed graph of KG to explore the influence that connected entities exert on each other along with the link.
- $RKAC_{(no\_attr)}$: Consider the model which propagates only collaborative signals without attribute association to explore the influence of attribute association propagation based on collaborative signals propagation.
- $RKAC_{(no\_atte)}$: In order to explore the effect of attention mechanism in our model, We disable the attention mechanism (formula (9)) and set $\gamma(e_i^h, r_i)$ as $1/|S_o^l|$.

Table 7 shows that the effect of $RKAC_{(directed)}$ on the four datasets is generally not as good as that of RKAC, which shows that undirected graph KG can effectively transfer the information of connected entities. By comparing the effects of variant experiment $RKAC_{(no\_attr)}$ with RKAC, it can be concluded that aggregation of multi-hop and high-correlation attribute neighbor information plays an important role

Table 4
The result of AUC and F1 *w.r.t.* different depth of layer on all four datasets.

Depth of layer L	1		2		3		4	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
Music	0.8555	0.7821	**0.8633**	**0.7888**	0.8580	0.7728	0.8607	0.7898
Book	0.7671	0.6805	**0.7675**	**0.6817**	0.7641	0.6765	0.7647	0.6809
Movie	**0.9845**	**0.9431**	0.9834	0.9415	0.9832	0.9412	0.9828	0.9406
Restaurant	0.8816	0.8052	0.8827	0.8069	0.8826	0.8066	**0.8827**	**0.8070**

Table 5
The result of AUC on Last.FM *w.r.t.* different sizes of the triple set.

	Item				
User	4	8	16	32	64
4	0.842	0.849	0.846	0.849	0.859
8	0.855	0.859	0.860	0.856	0.854
16	0.855	0.862	**0.863**	0.861	0.862
32	0.855	0.857	0.857	0.860	0.859
64	0.854	0.849	0.855	0.852	0.853

Table 6
The result of AUC on MovieLens-20M *w.r.t.* different sizes of the triple set.

	Item				
User	4	8	16	32	64
4	0.978	0.978	0.978	0.977	0.977
8	0.981	0.980	0.980	0.980	0.978
16	0.982	0.982	0.982	0.982	0.982
32	0.983	0.983	0.983	0.983	0.983
64	**0.984**	0.984	0.984	0.984	0.984

Table 7
Comparison among RKAC and its two variants in CTR prediction scenario.

	Music	Book	Movie	Restaurant
$RKAC_{(directed)}$	0.853	0.763	0.853	0.879
$RKAC_{(no\_attr)}$	0.856	0.753	0.984	0.882
$RKAC_{(no\_atte)}$	0.759	0.700	0.977	0.863
$RKAC$	**0.863**	**0.767**	**0.984**	**0.882**

in improving the performance of the recommendation system. The experimental results of $RKAC_{(no\_atte)}$ are worse than those of RKAC on all datasets. This supports that attention mechanism is a powerful determinant when aggregating information of multi-hop neighbors.

Here are two points worth noting:

(1) The experimental results of $RKAC_{(no\_attr)}$ on the datasets of movies and restaurants are the same as those of RKAC. One of the main reasons for this result is that these two datasets have a large number of users and user interactions. However, the corresponding knowledge graph contains relatively simple information (the number of KG triples is small), and the information is not redundant but also very scarce. Therefore, whether to remove attribute association propagation only collaboration signals has little impact on its accuracy. In contrast, undirected graph propagation is more effective. In the case of less entity information, undirected graph propagation actually plays a supplementary role and better transmits the information of adjacent entities, this also explains why the effect of variant experiment $RKAC_{(directed)}$ is lower than that of variant experiment $RKAC_{(no\_attr)}$ on the two datasets of movies and restaurants.

(2) In the Book-Crossing dataset on the experiment result, we can see that the effect of $RKAC_{(no\_attr)}$ is much lower than that of the other two experimental results, $RKAC_{(directed)}$ and RKAC, mainly because the knowledge graph corresponding to book dataset is rich in information, so the collaborative signals propagation model with attribute association can filter the redundant neighbor information well.

6. Conclusion and future work

In this paper, we propose an attribute collaborative knowledge-aware attentive network algorithm, which is called RKAC for short. RKAC uses the item attribute in KG and the user–item's collaborative signals to filter the unrelated auxiliary knowledge information and extract neighbor entities with high correlation, so as to obtain the accurate representation of users and items. Experimental results on four benchmark datasets show that RKAC is superior to other similar methods.

Besides, our research has some practical significance. We do some filtering through attributes to remove some unrelated information from the propagation process of collaborative signals. In this way, we can quickly obtain other highly relevant items of users' favorite items. However, for the problem of a bubble of recommendations limited by a KG short path, you may lose some needs of diversity in life.

Therefore, there are two promising directions in future work: (1) Knowledge graph is the auxiliary information of the item. We can also integrate all kinds of user-related ancillary information into knowledge recommendation, such as social networks, which will help describe user preferences more accurately. (2) In the behavior of users purchasing items, there are many potential advantages of items that affect users' decisions, such as brand awareness, quality and appearance of items. Therefore, the influencing factors of potential advantages of items can be added from the decision-making level to improve the embedding representation of users.

CRediT authorship contribution statement

Fulan Qian: Conceptualization, Methodology, Writing – original draft, Software. **Yuhui Zhu:** Methodology, Writing – review & editing. **Hai Chen:** Investigation, Supervision. **Jie Chen:** Investigation. **Shu Zhao:** Supervision. **Yanping Zhang:** Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the General Project of the National Natural Science Foundation of China (61876001), and the the Natural Science Foundation for the Higher Education Institutions of Anhui Province of China (KJ2021A0039). In addition, we acknowledge the High performance Computing Platform of Anhui University for providing computing resources.

References

Bordes, A., Usunier, N., García-Durán, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems 26: 27th annual conference on neural information processing systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States* (pp. 2787–2795).

Cai, M., & Zhu, J. (2019). Knowledge-aware graph collaborative filtering for recommender systems. In *15th international conference on mobile ad-hoc and sensor networks* (pp. 7–12). IEEE.

Cao, Y., Wang, X., He, X., Hu, Z., & Chua, T. (2019). Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The World Wide Web conference* (pp. 151–161).

Chen, H., Qian, F., Chen, J., Zhao, S., & Zhang, Y. (2021a). Attribute-based neural collaborative filtering. *Expert Systems with Applications*, Article 115539.

Chen, H., Qian, F., Chen, J., Zhao, S., & Zhang, Y. (2021b). FG-RS: Capture user fine-grained preferences through attribute information for recommender systems. *Neurocomputing*, 195–203.

Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y., & Bresson, X. (2020). Benchmarking graph neural networks. URL: https://arxiv.org/abs/2003.00982.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*: *Vol. 9*, (pp. 249–256). JMLR.org.

Hahnloser, R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 947–951.

Hamilton, W. L., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in neural information processing systems 30: annual conference on neural information processing systems 2017* (pp. 1024–1034).

Han, J., & Moraga, C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. In *From natural to artificial neural computation, international workshop on artificial neural networks, proceedings*: *Vol. 930*, (pp. 195–201).

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on World Wide Web* (pp. 173–182).

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).

Huang, J., Zhao, W. X., Dou, H., Wen, J., & Chang, E. Y. (2018). Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR conference on research & development in information retrieval* (pp. 505–514).

Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015). Knowledge graph embedding via dynamic mapping matrix. In *Long Papers*: *Vol. 1*, *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing of the Asian federation of natural language processing* (pp. 687–696).

Koren, Y., Bell, R. M., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. (pp. 30–37).

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).

Sha, X., Sun, Z., & Zhang, J. (2019). Attentive knowledge graph embedding for personalized recommendation. URL: http://arxiv.org/abs/1910.08288.

Shi, C., Hu, B., Zhao, W. X., & Yu, P. S. (2019). Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 357–370.

Sun, Y., & Han, J. (2012). Mining heterogeneous information networks: A structural analysis approach. *SIGKDD Explorations*, 20–28.

Sun, Z., Yang, J., Zhang, J., Bozzon, A., Huang, L., & Xu, C. (2018). Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the 12th ACM conference on recommender systems* (pp. 297–305).

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).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems 30: annual conference on neural information processing systems 2017* (pp. 5998–6008).

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *6th international conference on learning representations, conference track proceedings*.

Wang, X., He, X., Cao, Y., Liu, M., & Chua, T. (2019). KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 950–958).

Wang, X., Huang, T., Wang, D., Yuan, Y., Liu, Z., He, X., & Chua, T. (2021). Learning intents behind interactions with knowledge graph for recommendation. In *WWW '21: The web conference 2021* (pp. 878–887).

Wang, Z., Lin, G., Tan, H., Chen, Q., & Liu, X. (2020). CKAN: Collaborative knowledge-aware attentive network for recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in information retrieval* (pp. 219–228).

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*.

Wang, X., Wang, D., Xu, C., He, X., Cao, Y., & Chua, T. (2019). Explainable reasoning over knowledge graphs for recommendation. In *The thirty-third AAAI conference on artificial intelligence, the thirty-first innovative applications of artificial intelligence conference, the ninth AAAI symposium on educational advances in artificial intelligence* (pp. 5329–5336).

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the twenty-eighth AAAI conference on artificial intelligence* (pp. 1112–1119).

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).

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 on World Wide Web* (pp. 1835–1844).

Wang, H., Zhang, F., Zhang, M., Leskovec, J., Zhao, M., Li, W., & Wang, Z. (2019). Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 968–977).

Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., & Guo, M. (2019). Multi-task feature learning for knowledge graph enhanced recommendation. In *The World Wide Web conference* (pp. 2000–2010).

Wang, H., Zhao, M., Xie, X., Li, W., & Guo, M. (2019). Knowledge graph convolutional networks for recommender systems. In *The World Wide Web conference* (pp. 3307–3313).

Weimer, M., Karatzoglou, A., & Smola, A. J. (2008). Adaptive collaborative filtering. In *Proceedings of the 2008 ACM conference on recommender systems* (pp. 275–282). http://dx.doi.org/10.1145/1454008.1454050.

Xin, X., He, X., Zhang, Y., Zhang, Y., & Jose, J. M. (2019). Relational collaborative filtering: Modeling multiple item relations for recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 125–134).

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 *Seventh ACM international conference on web search and data mining* (pp. 283–292).

Yu, X., Ren, X., Sun, Y., Sturt, B., Khandelwal, U., Gu, Q., Norick, B., & Han, J. (2013). Recommendation in heterogeneous information networks with implicit user feedback. In *Seventh ACM conference on recommender systems* (pp. 347–350).

Zhang, H., Shen, F., Liu, W., He, X., Luan, H., & Chua, T. (2016). Discrete collaborative filtering. In *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval* (pp. 325–334).

Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W. (2016). Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 353–362).

Zhu, M., Zhen, D., Tao, R., Shi, Y., Feng, X. Y., & Wang, Q. (2019). Top-N collaborative filtering recommendation algorithm based on knowledge graph embedding. In *Knowledge management in organizations - 14th international conference, proceedings* (pp. 122–134).