Knowledge-Based Systems 259 (2023) 110078

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

AGRE: A knowledge graph recommendation algorithm based on multiple paths embeddings RNN encoder

Na Zhao [a,1], Zhen Long [a,1], Jian Wang [b,1], Zhi-Dan Zhao [c,d,*]

[a] Engineering Research Center of Cyberspace, School of Software, Yunnan University, Kunming, 650504, Yunnan, China
[b] College of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, 650217, Yunnan, China
[c] Department of Computer Science, School of Engineering, Shantou University, Shantou, 515063, Guangdong, China
[d] Key Laboratory of Intelligent Manufacturing Technology (Ministry of Education), Shantou University, Shantou, 515063, Guangdong, China

ARTICLE INFO

Article history:
Received 14 January 2022
Received in revised form 24 October 2022
Accepted 27 October 2022
Available online 5 November 2022

Dataset link: https://github.com/sunzhuntu/Recurrent-Knowledge-Graph-Embedding, https://github.com/hwwang55/MKR

Keywords:
Knowledge graph
Recommendation algorithm
Multiple paths
MRNN

ABSTRACT

More and more researches have focused on the use of knowledge graphs (KG) to solve the sparsity problem of traditional collaborative filtering recommendation systems. Most KG based recommendation algorithms focus on independent paths connecting users and items, or iteratively propagate user preferences in KG. However, the current approachs that focus on indedpent paths ignore the association between paths. Therefore, in this study, we propose a knowledge graph recommendation system algorithm for the multiple paths RNN encoder (AGRE), which fully considers the association between paths. Specifically, the paths between the user and the item are coded by a specified RNN (MRNN) to accurately learn the user's preferences. Traditional RNNs can encode multiple paths without considering the association between paths, but our RNN can encode multiple paths with considering the association between paths. We have compared AGRE with other state-of-the-art algorithms on three real-world datasets, and achieved good results in terms of AUC and Precision@K. This indicates that AGRE could solve the problem of sparse interaction between users and items, and could make full use of the knowledge graph for recommendation.

1. Introduction

With the rapid growth of data volume, the phenomenon of information overload becomes more and more common, and it is difficult for individual users to quickly find items and content of interest to them. As a technology to solve information overload, recommendation system has received more and more attention and has been applied in some practical application scenarios [1–4]. In order to achieve personalized recommendations, traditional collaborative filtering methods use various online unstructured data of users, such as the number of clicks, ratings, etc., and predict user preferences for items based on these historical information [5]. Generally speaking, a learnable collaborative filtering model consists of embedding and interactive data modeling [6]. Among them, embedding can transform users and items into vector forms, and interactive data modeling can reconstruct historical interactive information of users and items based on embedding vectors. For example, the matrix factorization model directly expresses the IDs of users and items in vector form, and models their interaction through inner products [7]. The neural collaborative filtering model uses a nonlinear neural network instead of the inner product interaction function in the matrix factorization model [6,8]. The collaborative filtering model based on translation uses Euclidean distance as an interactive function to predict recommendation results [9].

In order to solve the problem that the traditional collaborative filtering system only considers descriptive features (user ID and item ID, attributes) to construct the embedding function, which may make it difficult for the collaborative filtering algorithm to generate a satisfactory embedding vector. In addition, when the number of interactions deeply digs into the complex user–item relationship, these methods tend to overfit, which makes the sparsity problem even more serious. Therefore, researchers propose to use auxiliary information in the recommendation system to enrich the semantic representation of items, such as social networks, attributes, and multimedia [10–16]. Recently, the knowledge graph (KG) has attracted more and more attention from the recommendation system research community because it contains more comprehensive and effective recommendation auxiliary data such as networks, item attributes and user information etc [17–24]. The recommendation algorithm considering the knowledge graph is generally considered from the following three

* Corresponding author at: Key Laboratory of Intelligent Manufacturing Technology (Ministry of Education), Shantou University, Shantou, 515063, Guangdong, China.
E-mail address: zzhidanzhao@gmail.com (Z.-D. Zhao).
1 These authors have contributed equally to this work.

https://doi.org/10.1016/j.knosys.2022.110078
0950-7051/

aspects: direct relationship [18,25–30], semantic path [31,31–36] and propagation [37–42].

However, the above algorithms have some problems such as the following: (1) Since only the direct relationship between entities is considered, the method based on direct relationship cannot fully capture the complex semantics of the connection between the user and the item to enhance the embedding learning; (2) the method based on semantic path inevitably leads to information loss to decompose the complicated association between users and items into separate linear paths, so they ignore the association between paths; (3) Due to there is no constraint between the target entity and its neighbors, the propagation-based embedding learning process introduces noise (neighbors unrelated to specific interactions) and heavy computation (a large number of neighbors), thereby reducing the performance of the algorithm. In order to solve the above-mentioned problems, this study proposes a knowledge graph recommendation algorithm based on a multiple paths RNN encoder (AGRE).

AGRE can learn more accurate embedding by mining and connecting specific semantic paths from KG, and making use of the rich semantics of the connection between users and items in an automatic way through considering the association between paths. To sum up, our contribution are as follows: (1) We consider the association between paths (In addition to u and v, there are other common nodes in the path. For example, the path $u \rightarrow a \rightarrow b \rightarrow v$ and the path $u \rightarrow a \rightarrow c \rightarrow v$ have common node a); (2) We design an improved RNN encoder and apply it to encode multiple paths (MRNN). As far as we know, it is the first time that RNN has been applied to encode multiple paths between users and items with considering the association between paths; (3) We conduct experiments on three real datasets, and compared with other state-of-the-art algorithms. The experimental results indicate AGRE's rationality and effectiveness.

The remainder of this work is organized as follows: In Section 2, we introduce related work. In Section 3, we specify Framework of AGRE 3.1, and MRNN 3.2, attention mechanism 3.3, prediction 3.4, model optimization 3.5 and experimental setup 3.6. In Section 4, we conduct extensive experiments to evaluate the proposed method for AGRE. Finally, in Section 5, we discuss and conclude this work.

2. Related work

Recommendation algorithms based on knowledge graphs are mainly considered from the following three aspects: methods based on direct relationships, methods based on semantic paths, and methods based on propagation. First, the method based on direct relationship is to use the relationship between directly connected entities in KG for embedding learning. For example, CKE [25] uses TransR [26] to learn items' embeddings with the participation of KG. DKN [27] uses KG to generate news embedding through TransD [28]. KTUP [18] complements recommended model and knowledge map through TransH [29] joint learning. MKR [30] uses KG embedding to standardize recommendation tasks. Additionally, the semantic path-based method measures the connectivity between users and items through the similarity of meta-paths [31], such as PER [31], HeteCF [32] and SemRec [33]. Some researchers use random walk based on meta-path to achieve better embedding learning, such as Metapath2vec [34]. However, methods based on meta-path rely on manual features, which greatly hinders the universality of these methods. Therefore, RKGE [35] and KPRN [36] automatically extract the paths that have length constraints connecting users and items, and model these paths through recurrent neural network (RNN). Moreover, the strategy of the propagation-based method is to iteratively propagate across the entire KG to help recommendations. RippleNet [37], for example, expands user's interest along the edge in KG to discover user's potential interest. KGCN [38] and KGCN-LS [39] use Graph convolution Network (GCN [40]) to calculate items' embeddings through the propagation between neighbors in KG. KGAT [41] recursively performs propagation on KG through GNN to refine entity embedding. These methods, which take the whole KG as input, learn entities' embedding by aggregating the information from its neighbors.

3. Model

This section introduces the advantages of the ARGE algorithm and its detailed construction process. In this study, the knowledge graph is an directed graph. Its construction process is as follows: (1) Add all users, items and their attributes to the knowledge graph; (2) Connect items and their corresponding attributes; (3) Connect users and items that interact with users.

Compared with the traditional knowledge graph algorithms [35–41,43–46], we believe that multiple paths in the knowledge graph are beneficial for recommendation, mainly from the following two aspects. On the one hand, as a nonlinear combination of independent paths, multi-paths contain rich semantic and topological information and are more expressive than independent paths. On the other hand, the paths only retains the entities and relationships related to the specific (u, v) interaction, which can effectively avoid the noise caused by KG.

However, applying multiple paths to infer users' preferences is not an easy task, and it mainly faces the following two challenges. The first challenge is how to consider the associations between paths. Association is universal, and better results are usually obtained when association are taken into account. The GNN is a good example, which introduces the association between samples. But the current approaches ignore the association between paths. The second challenge is how to encode the extracted multiple paths in an efficient way to enhance user preference inference. In the literatures [35,36], RNN has been used to generate representation vectors for paths, but the first challenge prompted us to design a RNN model that could be used to encode multiple paths and take into account the associations between paths. The RNN generates multiple path representation vectors that consider the association between paths from each independent path representation.

Table 1 is a description of symbols commonly used in this study.

3.1. Framework

This section introduce the overall framework of AGRE, which applies an improved RNN to encode multiple paths. The entire working process of the model is shown in Fig. 1. AGRE mainly includes four modules, followed by Path Sampling, MRNN, Attention and MLP. The specific input and output of each module are shown in Fig. 1. Path sampling is to extract the paths between user u and item v. The length of the paths is l. We bring the paths in $\mathcal{P}_{u,v}$ into the model, and this process incurs huge computational cost. Additionally, too many paths may introduce noise. In order to balance the computational cost and possible noise problems, we set a fixed number of paths to n during the calculation. These paths form the set $\mathcal{P}_{u,v}$. MRNN can encode multiple paths, taking into account the associations between paths. Attention can aggregate multiple pieces of information to obtain the embedding of multiple paths. The MLP outputs the predicted probability of user u interacting with the item v.

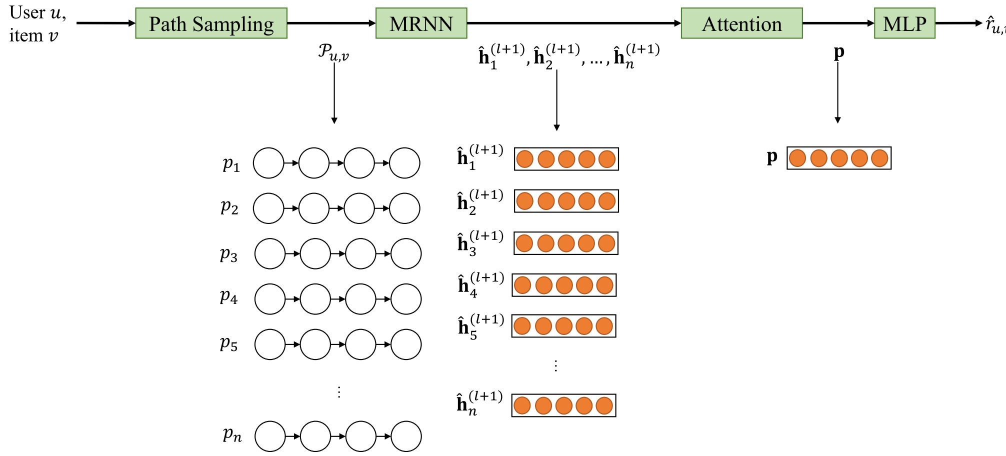


Fig. 1. The user u and item v are input into path sampling to obtain the paths between user u and item v. And the path set $\mathcal{P}_{u,v}$ is input into MRNN to obtain the hidden state $\hat{\mathbf{h}}_j^{(l+1)}$ that is the embedding of path. Attention aggregates information from multiple paths. Finally, the embedding $\mathbf{p}$ of multiple paths is input into the MLP, and the probability $\hat{r}_{u,v}$ of the user u clicking on the item v is obtained.

Table 1
Model symbol introduction.

Notation	Description
$\mathbf{W}_*$	Learnable parameter
$\mathbf{H}_*$	Learnable parameter
$\mathbf{b}_*$	Learnable parameter
$\mathcal{P}_{u,v}$	The set of path between u and v
$r_{u,v}$	$r_{u,v}=1$ represents that user u has interacted with item v, $r_{u,v}=0$ indicates that user u has not interacted with item v
u	User
v	Item
l	The length of paths
n	The size of $\mathcal{P}_{u,v}$
p_i	The path in $\mathcal{P}_{u,v}$
$\mathbf{x}_i^j$	The relation-embedding of the j-th node in path p_i
$\mathbf{h}_i^j$	The hidden state of the j-th node in path p_i
$\hat{\mathbf{h}}_i^j$	The path-association hidden state of the j-th node in ith path
σ	Sigmoid function
$\hat{r}_{u,v}$	The predicted probability of user u interacting with item v
$\mathcal{J}$	Loss function
$\mathcal{F}$	The parameters of AGRE
$\mathcal{D}_{train}$	Training set
λ	L2 regularization coefficient
BCELOSS	Cross entropy

3.2. MRNN

The traditional path-based method only encodes one path, and ignores the association between paths. In this work, we proposed a MRNN method, compared with RNN can only encode one path, MRNN considers the association between paths and can encode multiple paths. The MRNN is shown in Fig. 2.

3.2.1. Relation perception

Because relationships have different meanings for entities, relationships can help models express clearer semantics. For example, singing relationship and composition relationship have different meanings for a piece of music. And relationships between entities are used in many literatures [30,37,47].

Eq. (1) is the relation-embedding of the j-th node in the path p_i, where $\mathbf{e}_i^j$ is the embedding of e_i^j, and $\mathbf{r}_i^j$ is the embedding of r_i^j.

$$\mathbf{x}_i^j = \begin{bmatrix} \mathbf{e}_i^j \\ \mathbf{r}_i^j \end{bmatrix}, \tag{1}$$

For examples, $\mathcal{P}_{u,v} = \{p_1, p_2, p_3, \ldots, p_n\}$, $p_i = \{e_i^1, r_i^1, e_i^2, r_i^2, e_i^3, \ldots, r_i^l, e_i^{(l+1)}\}$, where $\mathcal{P}_{u,v}$ is the set of the paths between user u and item v, p_i is the path in $\mathcal{P}_{u,v}$, e_i^j is the j-th node in the path p_i, r_i^j is the relation between e_i^j and $e_i^{(j+1)}$. In particular, $r_i^{(l+1)}$ is a relation that does not exist. For consistency with other entities, we give an empty relationship. An empty relationship means no relationship. The input of RNN is $[\mathbf{x}_1^j, \mathbf{x}_2^j, \mathbf{x}_3^j, \ldots \mathbf{x}_n^j]$.

3.2.2. Path association

To effectively apply KGs for recommendation, obtaining embeddings for multiple paths is a key aspect to consider. Since there are also associations between different paths, in order to obtain a better embedding of multiple paths in the knowledge graph, we must add the association between paths in the AGRE algorithm. In traditional relation mining methods, semantic path-based methods do not consider the associations among paths [35, 36,48]. Here, we comprehensively considered the associations between paths.

$$\mathbf{h}_i^j = \sigma(\mathbf{W}\mathbf{x}_i^j + \mathbf{H}\mathbf{h}_i^{(j-1)}), \tag{2}$$

$$[\hat{\mathbf{h}}_1^{(l+1)}, \hat{\mathbf{h}}_2^{(l+1)}, \ldots, \hat{\mathbf{h}}_n^{(l+1)}] = \sigma([\mathbf{h}_1^{(l+1)}, \mathbf{h}_2^{(l+1)}, \ldots, \mathbf{h}_n^{(l+1)}]\mathbf{P}), \tag{3}$$

In Eqs. (2) and (3), $\mathbf{W} \in \mathbb{R}^{2d\times 2d}$ and $\mathbf{H} \in \mathbb{R}^{2d\times 2d}$ are the learnable matrixes, $\hat{\mathbf{h}}_i^{(j-1)} \in \mathbb{R}^{2d\times 1}$ is the hidden state of the $(j-1)$-th node in the path p_i. $\mathbf{P} \in \mathbb{R}^{n\times n}$ is a learnable matrix, and $\mathbf{h}_i^{(l+1)}$ is the embedding of the path p_i. We use a normal RNN to obtain embeddings for each path, and use the matrix $\mathbf{P}$ to obtain path-association embeddings of the paths. The path-association embeddings fully considers the connections between paths. The path-association embeddings are generated by the embeddings of the paths, so path-association embeddings take into account the association between paths. The matrix $\mathbf{P}$ can distinguish the influence of each path embedding, which acts as an attention mechanism.

3.3. Attention mechanism

The MRNN finally outputs the hidden states of node v in each path. It is the embeddings for these paths. In order to obtain embedding for multiple paths, these hidden states must be aggregated. There are two ways of aggregation. One is to add all hidden states and take the average. The other is to use an attention mechanism to distinguish the effects of these hidden states. Intuitively, different hidden states have different effects

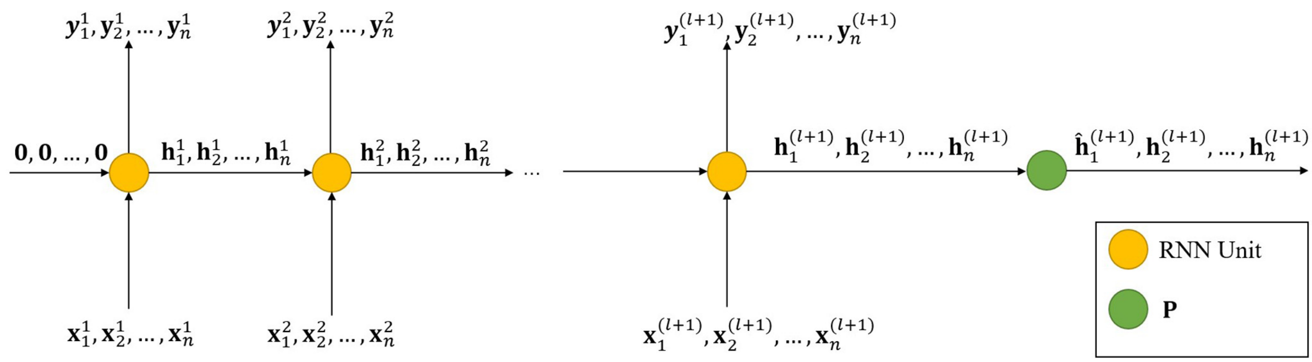


Fig. 2. The difference between MRNN and ordinary RNN is the matrix **P**, we consider the association between paths through the learnable matrix **P**. To obtain the hidden state of each node, each node of the path is input into the RNN unit. The embedding of the path p_i is obtained. Finally, the path-association embeddings are obtained by the matrix **P**.

on the embedding of multiple paths, so the attention mechanism should be adopted.

$$a_i = \sigma(\mathbf{W}_a\hat{\mathbf{h}}_i^{l+1} + \mathbf{b}_a), \tag{4}$$

$$\mathbf{p} = \sum_{i=1}^{i=n} a_i \hat{\mathbf{h}}_i^{(L+1)}, \tag{5}$$

In Eq. (4), a_i is the attention weight for i-th path in $\mathcal{P}_{u,v}$, $\mathbf{W}_a \in \mathbb{R}^{2d\times 1}$ and $\mathbf{b}_a \in \mathbb{R}^{1\times 1}$ are the learnable matrix. In Eq. (5), $\mathbf{p} \in \mathbb{R}^{2d\times 1}$ is the embedding for multiple paths in $\mathcal{P}_{u,v}$.

3.4. Prediction

By applying AGRE, we can better learn the representation of multiple paths. The representation of multiple paths is further used to predict user's preference for item. We use a MLP to simulate the interaction between user and item [35], which is shown as Eq. (6):

$$\hat{r}_{uv} = \sigma(\mathbf{W}_p\mathbf{p} + \mathbf{b}_p), \tag{6}$$

Where $\hat{r}_{uv}$ is the predicted probability of user u interacting with item v, $\mathbf{W}_p \in \mathbb{R}^{2d\times 1}$ and $\mathbf{b}_p \in \mathbb{R}^{1\times 1}$ are learnable parameters. We use sigmoid function to control the predicted probability within the range of 0–1.

3.5. Model optimization

Objective function: Many studies [6,36,41,47–49] treat top-k recommendation as a binary classification task. Therefore, cross-entropy is the objective function, the formula is as follows Eq. (7)

$$\mathcal{J} = \sum_{(u,v)\in\mathcal{D}_{train}} \mathrm{BCELOSS}(\hat{r}_{u,v}, r_{u,v}) + \lambda\|\mathcal{F}\|_2^2 \tag{7}$$

where λ is the L2 regularization coefficient, $\mathcal{F}$ is the parameters of the model, and $\mathcal{D}_{train}$ is training set. The first term represents cross-entropy, and the second term represents L2 regularization.

3.6. Experienment setup

3.6.1. Dateset

Datasets: (1) Last.FM [30] belongs to music recommendation. (2) Movielens-100K [35] is the record of user movies. (3) Yelp [35] is the record of user reviews for businesses.

Data processing: If an item is in a user's record, the user has interacted with the item. According to literature [27,37,38], the number of negative samples is the same as the number of positive samples. We randomly divide the records of each user into training set and test set according to 7:3. The number of attributes extracted from different data sets is displayed in Table 2, including: users, items, relations, triples, interactions and entities.

Table 2
Dataset attributes.

Dataset	Last.FM	Movielens-100K	Yelp
#users	1876	943	16308
#items	3846	1674	11516
#interactions	21173	99974	199422
#entities	9366	6800	12088
#triples	15518	24686	85758
#relations	60	6	4

3.6.2. Evaluation

According to the literature [6,36,50], To reduce the complexity of evaluating the algorithm, we randomly select 100 users, and we randomly select 100 items that have not interacted with each user, and then randomly select 1 item in the test set. The recommendation list for each user consists of 101 items.

AUC [51] measures the extent to which a recommender system can distinguish items that users like from items they do not like. Precision@K [51] is the proportion of items that users like among the multiple items recommended by the system (K={1, 2, 5, 6, 7, 8, 9, 10}). The higher these metrics, the better the performance of the algorithm.

3.6.3. Baselines

Some state-of-the-art algorithms recently proposed are baselines. These algorithms include methods based on direct relationships, methods based on semantic paths and methods based on propagation.

- Methods based on direct relationships: CKE applies TransR and has no text and image data. MKR [30] applies TransH [29] to combine recommendation module and KG embedding module. For MKR, we search for best L in {1, 2, 3}, best T in {1, 2, 3, 4, 5}. According the code of MKR, the KGE weight is fixed to 10^{-6}.
- Methods based on semantic paths: MCRec [48] uses meta-path. We design two meta-paths: *user* → *item* → *attribute* → *item* and *user* → *item* → *user* → *item*. The number of paths is set by the original paper. RKGE [35] automatically extracts the paths that have length constraints connecting users and items. RKGE encodes the paths by RNN. The number of paths is set by the code of RKGE and the length of path is set by the original paper. KARN [44] integrates the user click history sequence and

Table 3
The setting of some hyper parameters.

Dataset	n	d	l	η
Last.FM	30	20	3	0.001
Movielens-100K	30	40	3	0.01
Yelp	20	20	3	0.001

the path connectivity between the user and the recommended items. Except for the embedding dimension, other parameters follow the original paper settings.

- Methods based on propagation: RippleNet [37], KGCN [38] and CKAN [47] iteratively propagate across the KG to improve recommendation. We search best H or L in {1, 2, 3, 4}, and the size of set or neighbors in {5, 10, 20, 30, 40, 50}. The KGE weight in RippleNet is set by the code of RippleNet.

For all algorithms, the optimizer is Adam, and mini-batch gradient descent algorithm is applied for model optimization. They are widely used optimization strategies [37,38,41]. The batch size is fixed to 1024, the L2 regularization coefficient is fixed to 10^{-5}, and the learning rate adjusted based on experience. We search for best embedding dimension in {5, 10, 20, 30, 40, 50}.

3.6.4. Parameter settings

For AGRE, d (the embedding dimension of entity) and n (the number of paths between user and item), l(the length of path) are important hyperparameters. Initially, We search for the best n in {5, 10, 20, 30, 40, 50} and search for the best d in {5, 10, 20, 30, 40, 50}, search for the best l in {3, 4, 5}. The roles of these hyperparameters are discussed in Section 4.3. For comparison with other state-of-the-art algorithms, other parameters are the same as these algorithms. Table 3 shows the hyperparameters in this work, where η is the learning rate.

4. Experiment and analysis

In this section, the following three questions are answered through experiments on Last.FM, Movielens-100K and Yelp datasets: (1) Does the AGRE outperforms state-of-the-art algorithms? (2) Do some modules of AGRE affect the model? (3) How does the AGRE is affected by the hyperparameters?

4.1. Compare to baselines

In order to better understand the performance of the AGRE algorithm, we compare the performance of the eight different state-of-the-art algorithms on three different datasets, and Fig. 3 shows the results. It is apparent that in majority of cases, the performance of the AGRE is better than other eight different algorithms. Comparing Figs. 3(a), 3(b) and 3(c), it can be observed that as K increases, the precision decreases. The performance of the AGRE algorithm is generally better than other algorithms. For example, in Last.FM, when K={1, 2, 5, 6, 7, 8, 9, 10}, compared with the best results of state-of-the-art algorithms, AGRE outperforms by 8.91% on average. Additionally, compared with the worst case of these algorithms, when K={1, 2, 5, 6, 7, 8, 9, 10}, AGRE outperforms by up to 65.02% on average. Compared with the average values of other algorithms when K={1, 2, 5, 6, 7, 8, 9, 10}, AGRE outperforms by 34.27%, 34.55%, 24.59%, 25.13%, 27.54%, 22.51%, 21.43% and 18.45%, respectively. These results demonstrate that the competitive prediction precision of the AGRE algorithm. It should, however, be noted that K could affect the precision of AGRE and other algorithms.

Generally speaking, AUC is one of the most important evaluation metrics for measuring the performance of model. Table 4 illustrates the AUC values of each model on the three datasets. As can be seen in Table 4, in the majority of cases, the AGRE is revealed a considerably higher AUC values than other algorithms. For instance, compared with all other 8 top algorithms, the AUC value of the AGRE algorithm has achieved good performance in three different datasets.

Table 4
The comparison of AUC in different model.

Model	Dataset		
	Last.FM	Movielens-100K	Yelp
AGRE	0.853	0.886	0.878
MKR	0.796	0.812	0.858
CKE	0.828	0.845	0.842
RKGE	0.802	0.839	0.858
MCRec	0.839	0.845	0.869
CKAN	0.763	0.797	0.806
Ripplenet	0.805	0.873	0.825
KGCN	0.781	0.865	0.812
KARN	0.842	0.856	0.872

Table 5
The results for AGRE comparing to variants.

	Last.FM	Movielens-100K	Yelp
AGRE	0.853	0.886	0.878
AGRE-P	0.846	0.872	0.873
AGRE-a	0.846	0.879	0.873
AGRE-r	0.827	0.884	0.875

Yelp is the most sparse of all datasets. However, AGRE can still outperform other algorithms. This shows that AGRE can solve the data sparsity problem very well.

These results indicate that the effectiveness of MRNN in the AGRE algorithm. Beacause MRNN takes into account the association between paths.

4.2. Compare to variants

To investigate the underlying reasons for the performance advantages of the AGRE algorithm, we study the role of relation perception and attention mechanism, the matrix **P** in detail in this section.

(1) AGRE-r removes relation perception: $\mathbf{x}_i^j = \mathbf{e}_i^j$;

(2) AGRE-a removes attention mechanism: $\mathbf{p} = \frac{\hat{\mathbf{h}}_i^{(l+1)}}{n}$;

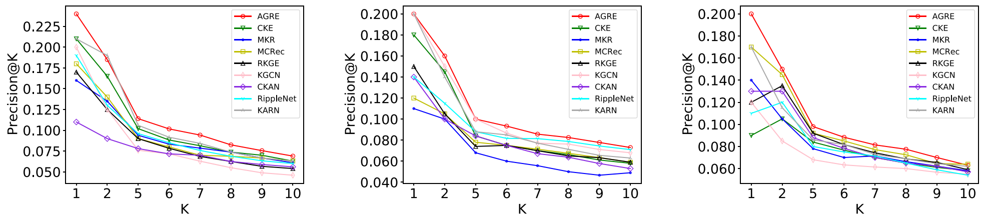
(3) AGRE-P removes the matrix **P**: $\hat{\mathbf{h}}_i^{(l+1)} = \mathbf{h}_i^{(l+1)}$.

The Table 5 shows the result for AGRE comparing to its variants. AGRE outperforms three variants, which shows that relation perception, attention mechanism and the matrix **P** are important roles for AGRE.

4.3. Parameter sensitivity analysis

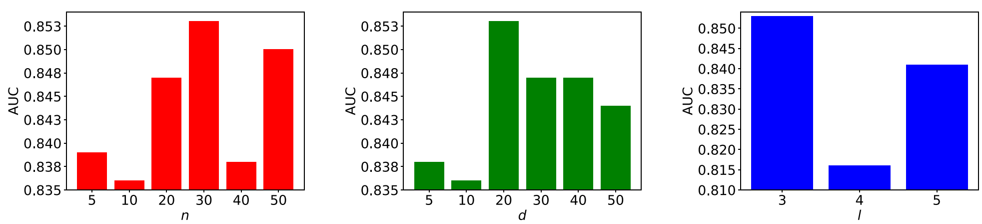
In order to explore the role of some hyperparameters in the AGRE algorithm, here we study the impact of the following three parameter changes on the performance of the AGRE algorithm on Last.FM. These parameters are: d (the embedding dimension of entity), n (the number of paths between user and item), and l (the length of path). We take the Last.FM dataset as an example to illustrate the impact of these hyperparameters.

It is known from the literature [49] that the number of paths between user and item is a significant influencing factor. It is evident from the results that the number of paths between user and item cannot be too large or too small, too large is easy to introduce noise, too small is easy to make MRNN become a RNN that encodes paths. As shown in Fig. 4(a), we observe that as n increases, the performance of the model is greatly improved at



(a) Last.FM (b) Movielens-100K (c) Yelp

Fig. 3. These figures show Precision@K of different algorithms in 3 datasets. We get the values of Precision@K by taking K in {1, 2, 5, 6, 7, 8, 9, 10}. Figure (a) is the Precision@K in Last.FM. Figure (b) is the Precision@K in Movielens-100K. Figure (c) is the Precision@K in Yelp. The meaning of each curve is shown in the upper right corner of each subfigure.



(a) The number of paths between user and item (b) The embedding dimension of entity (c) The length of path

Fig. 4. AUC reach max value when $n = 30, d = 20, l = 3$. AUC increases then decrease as d and n increase.

the beginning. Because more paths can help to encode the rich knowledge from KG. At the same time, the paths collected at the beginning may be independent (the nodes of each path do not have the same node except u and v). However, as n increases further, it gradually tends to a stable value in Last.FM, which means that too much semantic path integration introduce noise [49]. The results seem to indicate that the more paths between the user and the item, the more useful it is to improve the performance of the knowledge graph in terms of recommendation performance in a certain range.

It is known from the literature [37,41,49] that the appropriate embedding dimension is a significant influencing factor. Generally speaking, the embedding dimension cannot be too large or too small, too large is easy to overrepresent the entity, too small is easy to encode less useful information. Fig. 4(b) illustrate the findings of the impact of entity embedding dimension d on model performance. As can be seen, in the cases of data Last.FM, the performance of the AGRE starts to improve with the increase of d, reaches the best performance when d reaches a certain value, and then decreases as d continues to increase. This demonstrates that a larger embedding dimension significantly help to encode more useful information, but too large embedding dimensions may overrepresent the entity and thereby introducing noise [37,41,49]. In addition, we should also note that the change of AUC value and d is a fluctuating trend. These results imply that more underlying complex mechanisms need to be explored in future.

We also study the effect of path length on AGRE. The results are shown in Fig. 4(c). As can be seen from it, AGRE performs best when the path length is 3. Because shorter paths contain clearer and interpretable semantics [35].

5. Conclusion and discussion

Due to the knowledge graph (KG) contains recommendation auxiliary data such as network, item attributes and user information, it has received more and more attention from the recommendation system research community. In order to solve the problem that the methods based on semantic path ignore the association between paths, this work proposes AGRE, a knowledge graph recommendation algorithm based on multiple paths RNN encoder. It integrates the structured information of the knowledge graph into the algorithm. The improved RNN is conducive to encoding the multiple paths between users and items at the same time. This considers the association between paths. In addition, AGRE effectively alleviates the sparsity problem and improves the accuracy of recommendations. Experimental results on three real-world datasets indicate that AGRE is generally better than other state-of-the-art algorithms in AUC and Precision@K. Meanwhile, AGRE provides users with more accurate recommendation results, and it also has strong flexibility in fusing heterogeneous data.

The AGRE algorithm uses the paths between user and item. Therefore, the algorithm relies heavily on the degree of association between sampling paths. For example, if there is no association in the sampling paths, the AGRE algorithm degenerate into an algorithm for encoding one path. Future research should adopt universally relevant sampling strategies, and consider users' online ratings [42], temporal information [52] and Artificial intelligence [53] etc.

CRediT authorship contribution statement

Na Zhao: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Funding acquisition, Roles/Writing – original draft. **Zhen Long:** Data curation, Formal analysis, Methodology, Roles/Writing – original draft. **Jian Wang:** Formal analysis, Methodology, Roles/Writing – original draft, Funding acquisition. **Zhi-Dan Zhao:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Funding acquisition, Roles/Writing – original draft, Writing – review & editing.

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.

Data availability

Movielens-100K and Yelp are RKGE dataset, which can be obtained from the following website. If you want to use this data, please refer to the official instructions https://github.com/sunzhuntu/Recurrent-Knowledge-Graph-Embedding. The Last.FM is MKR dataset, which is available from the https://github.com/hwwang55/MKR.

Acknowledgments

This research was funded by the Key Research and Development Program of Yunnan Province (202102AA100021), the National Natural Science Foundation of China (Grant Nos. 62066048, 62176148), the Science Foundation of Yunnan Province (202101 AT070167), the Open Foundation of Key Laboratory in Software Engineering of Yunnan Province (2020SE311), the Scientific Research Foundation of Shantou University (Grant No. NTF19015), the 2020 Li Ka Shing Foundation Cross-Disciplinary Research (Grant No. 2020LKSFG09D), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2021A1515012294) and the Basic and Applied Basic Research of Colleges and Universities in Guangdong Province (Special Projects in Digital Economic. 2021ZDZX3025).

References

[1] Linyuan Lü, Tao Zhou, Link prediction in complex networks: A survey, Physica A 390 (2011) 1150–1170.
[2] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, Tao Zhou, Recommender systems, Phys. Rep. (ISSN: 0370-1573) 519 (1) (2012) 1–49.
[3] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, Guangquan Zhang, Recommender system application developments: A survey, Decis. Support Syst. 74 (2015) 12–32.
[4] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al., Wide & deep learning for recommender systems, in: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016 pp. 7–10.
[5] Hanwang Zhang, Fumin Shen, Wei Liu, Xiangnan He, Huanbo Luan, Tat-Seng Chua, Discrete collaborative filtering, in: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2016, pp. 325–334.
[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, Tat-Seng Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.
[7] Yehuda Koren, Robert Bell, Chris Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.
[8] Ruiping Yin, Kan Li, Guangquan Zhang, Jie Lu, A deeper graph neural network for recommender systems, Knowl.-Based Syst. 185 (2019) 105020.
[9] Yi Tay, Luu Anh Tuan, Siu Cheung Hui, Latent relational metric learning via memory-based attention for collaborative ranking, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 729–739.
[10] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, Qing He, A survey on knowledge graph-based recommender systems, IEEE Trans. Knowl. Data Eng. (2020).
[11] Lei Sang, Min Xu, ShengSheng Qian, Xindong Wu, Multi-modal multi-view Bayesian semantic embedding for community question answering, Neurocomputing 334 (2019) 44–58.
[12] Guangneng Hu, Yu Zhang, Qiang Yang, Conet: Collaborative cross networks for cross-domain recommendation, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 667–676.
[13] Jun Li, Chaochao Chen, Huiling Chen, Changfei Tong, Towards context-aware social recommendation via individual trust, Knowl.-Based Syst. 127 (2017) 58–66.
[14] Weike Pan, A survey of transfer learning for collaborative recommendation with auxiliary data, Neurocomputing 177 (2016) 447–453.
[15] Q. Zhang, P. Hao, J. Lu, G. Zhang, Cross-domain recommendation with semantic correlation in tagging systems, in: 2019 International Joint Conference on Neural Networks, IJCNN, pp. 1–8.
[16] Lei Sang, Min Xu, Shengsheng Qian, Matt Martin, Peter Li, Xindong Wu, Context-dependent propagating based video recommendation in multimodal heterogeneous information networks, IEEE Trans. Multimed. (2020).
[17] Qingyao Ai, Vahid Azizi, Xu Chen, Yongfeng Zhang, Learning heterogeneous knowledge base embeddings for explainable recommendation, Algorithms 11 (9) (2018) 137.
[18] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, Tat-Seng Chua, Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences, in: The World Wide Web Conference, 2019, pp. 151–161.
[19] Lei Sang, Min Xu, Shengsheng Qian, Xindong Wu, Knowledge graph enhanced neural collaborative filtering with residual recurrent network, Neurocomputing 454 (2021) 417–429.
[20] Zhisheng Yang, Jinyong Cheng, Recommendation algorithm based on knowledge graph to propagate user preference, Int. J. Comput. Intell. Syst. 14 (1) (2021) 1564–1576.
[21] Yang Yang, Yi Zhu, Yun Li, Personalized recommendation with knowledge graph via dual-autoencoder, Appl. Intell. (2021) 1–12.
[22] Bei Hui, Lizong Zhang, Xue Zhou, Xiao Wen, Yuhui Nian, Personalized recommendation system based on knowledge embedding and historical behavior, Appl. Intell. (2021) 1–13.
[23] Zhiqiang Pan, Honghui Chen, Collaborative knowledge-enhanced recommendation with self-supervisions, Mathematics 9 (17) (2021) 2129.
[24] Guibing Guo, Huihuai Qiu, Zhenhua Tan, Yuan Liu, Jing Ma, Xingwei Wang, Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems, Knowl.-Based Syst. 138 (2017) 202–207.
[25] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, Wei-Ying Ma, Collaborative knowledge base embedding for recommender systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 353–362.
[26] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, Xuan Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[27] Hongwei Wang, Fuzheng Zhang, Xing Xie, Minyi Guo, DKN: Deep knowledge-aware network for news recommendation, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1835–1844.
[28] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, Jun Zhao, 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 (Volume 1: Long Papers), 2015, pp. 687–696.
[29] Zhen Wang, Jianwen Zhang, Jianlin Feng, Zheng Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 no. 1, 2014.
[30] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, Minyi Guo, Multi-task feature learning for knowledge graph enhanced recommendation, in: The World Wide Web Conference, 2019, pp. 2000–2010.
[31] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, Tianyi Wu, Pathsim: Meta path-based top-k similarity search in heterogeneous information networks, Proc. VLDB Endowment 4 (11) (2011) 992–1003.
[32] Chen Luo, Wei Pang, Zhe Wang, Chenghua Lin, Hete-cf: Social-based collaborative filtering recommendation using heterogeneous relations, in: 2014 IEEE International Conference on Data Mining, IEEE, 2014, pp. 917–922.
[33] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S Yu, Yading Yue, Bin Wu, Semantic path based personalized recommendation on weighted heterogeneous information networks, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015 pp. 453–462.
[34] Yuxiao Dong, Nitesh V. Chawla, Ananthram Swami, Metapath2vec: Scalable representation learning for heterogeneous networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 135–144.
[35] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, Chi Xu, Recurrent knowledge graph embedding for effective recommendation, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 297–305.
[36] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, Tat-Seng Chua, Explainable reasoning over knowledge graphs for recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, 2019, pp. 5329–5336.
[37] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, Minyi Guo, Ripplenet: Propagating user preferences on the knowledge graph for recommender systems, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018 pp. 417–426.

[38] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, Minyi Guo, Knowledge graph convolutional networks for recommender systems. CoRR abs/1904.12575 (2019), 2019, arXiv preprint arXiv:1904.12575.

[39] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, Zhongyuan Wang, 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, 2019, pp. 968–977.

[40] Thomas N. Kipf, Max Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.

[41] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, Tat-Seng Chua, Kgat: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 950–958.

[42] Lina Wang, Xingshu Chen, Rui Tang, Yawei Yue, Yi Zhu, Xuemei Zeng, Wei Wang, Improving adversarial robustness of deep neural networks by using semantic information, Knowl.-Based Syst. 226 (2021) 107141.

[43] Xintao Ma, Liyan Dong, Yuequn Wang, Yongli Li, Hao Zhang, MNI: An enhanced multi-task neighborhood interaction model for recommendation on knowledge graph, PLoS One 16 (10) (2021) e0258410.

[44] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, Li Guo, A knowledge-aware attentional reasoning network for recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34 no. 04, 2020, pp. 6999–7006.

[45] Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xia, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, Yanfang Ye, Knowledge-aware coupled graph neural network for social recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, 2021 pp. 4115–4122.

[46] Zhiyi Deng, Changyu Li, Shujin Liu, Waqar Ali, Jie Shao, Knowledge-aware group representation learning for group recommendation, in: 2021 IEEE 37th International Conference on Data Engineering, ICDE, IEEE, 2021 pp. 1571–1582.

[47] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, Xiyang Liu, 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, 2020, pp. 219–228.

[48] Binbin Hu, Chuan Shi, Wayne Xin Zhao, Philip S. Yu, 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 &Amp; Data Mining, KDD '18, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450355520, 2018, pp. 1531–1540, http://dx.doi.org/10.1145/3219819.3219965.

[49] Xiao Sha, Zhu Sun, Jie Zhang, Attentive knowledge graph embedding for personalized recommendation, 2019, arXiv preprint arXiv:1910.08288.

[50] Ali Mamdouh Elkahky, Yang Song, Xiaodong He, A multi-view deep learning approach for cross domain user modeling in recommendation systems, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 278–288.

[51] Lv Linyuan Zhu Yuxiao, Overview of recommendation system evaluation indicators, J. Univ. Electron. Sci. Technol. China 02 (2012) 163–175.

[52] Pengpeng Shao, Dawei Zhang, Guohua Yang, Jianhua Tao, Feihu Che, Tong Liu, Tucker decomposition-based temporal knowledge graph completion, Knowl.-Based Syst. 238 (2022) 107841.

[53] Qian Zhang, Jie Lu, Yaochu Jin, Artificial intelligence in recommender systems, Complex Intell. Syst. 7 (1) (2021) 439–457.