

# Knowledge Graph Representation Reasoning for Recommendation System

Tao Li, Hao Li\*, Sheng Zhong, Yan Kang, Yachuan Zhang, Rongjing Bu and Yang Hu

Department of Software, Yunnan University, Kunming, 650500, China

\*Corresponding Author: Hao Li. Email: lihao707@ynu.edu.cn

Received: 17 January 2020; Accepted: 31 January 2020

**Abstract:** In view of the low interpretability of existing collaborative filtering recommendation algorithms and the difficulty of extracting information from content-based recommendation algorithms, we propose an efficient KGRS model. KGRS first obtains reasoning paths of knowledge graph and embeds the entities of paths into vectors based on knowledge representation learning TransD algorithm, then uses LSTM and soft attention mechanism to capture the semantic of each path reasoning, then uses convolution operation and pooling operation to distinguish the importance of different paths reasoning. Finally, through the full connection layer and sigmoid function to get the prediction ratings, and the items are sorted according to the prediction ratings to get the user's recommendation list. KGRS is tested on the movielens-100k dataset. Compared with the related representative algorithm, including the state-of-the-art interpretable recommendation models RKGE and RippleNet, the experimental results show that KGRS has good recommendation interpretation and higher recommendation accuracy.

**Keywords:** Knowledge graph; collaborative filtering; deep learning; interpretable recommendation; knowledge representation learning

## 1 Introduction

With the promotion and popularization of information society, the rapid development of Internet technology makes information present in front of users in an explosive growth trend. It is difficult for users to obtain the really useful part of information from the problem of information overload. How to effectively screen information for users is a topic in the era of big data. The main problem of recommendation system research is how to find the content that every user is interested in from the overloaded information and push the content to the user. Recommendation algorithms mainly include collaborative filtering recommendation (CF), content-based recommendation and so on [1].

In recent years, deep learning has made great breakthroughs and achievements in image processing, natural language processing, speech recognition and other fields due to its strong representation learning ability, which also brings new opportunities for the research of recommendation system. However, most of these recommendations based on deep learning are based on the idea of matrix decomposition, only considering the rating data of users and items, which restrains the recommendation effect of the models [2].

Collaborative filtering recommendation algorithm attempts to use the willingness of the crowd for recommendation [3]. Although collaborative filtering has achieved significant results in recommendation accuracy, it often faces problems such as cold start and lack of interpretability [4]. Content-based recommendation attempts to model users and items with various available content information. The content of an item is usually easy for users to understand, so in content-based recommendation, users can usually intuitively explain why the item is recommended. However, it is a time-consuming task to collect the required content information in different recommendation contexts [4]. In contrast, the construction of knowledge graph only needs to use entities and the relationships between entities, which will greatly reduce



the workload of extracting content information. As a new auxiliary data, knowledge graph has also attracted wide attention of academia and industry [5].

Based on the above background, this paper proposes KGRS (Knowledge Graph Representation Reasoning for Recommendation System) model. Experimental results on the MovieLens-100K dataset show that compared with the recent advanced interpretable recommendation model RKGE (Recurrent Knowledge Graph Embedding) [6] and RippleNet [7], the proposed model can achieve higher recommendation accuracy and it have good interpretability.

## **2 Related Work**

### ***2.1 Recommendation Based on Deep Learning***

Salakhutdinov first applied deep learning to recommendation system in [8], and proposed a collaborative filtering model based on Restricted Boltzmann machines. Huang proposed a deep structured semantic model in [9]. In 2017, He X proposed a neural collaborative filtering algorithm combining multi-layer perceptron model and generalized matrix decomposition model in [10], further improving the performance of the recommended model. However, most of these algorithm models are based on the idea of matrix decomposition, only considering the rating data of users and items, which restrains the recommendation effect of the models.

### ***2.2 Collaborative Filtering Recommendation***

The earliest collaborative filtering recommendation algorithm is user-based collaborative filtering proposed by Resnick in [11]. Later, Sarwar introduced the collaborative filtering algorithm based on items in [12]. When the collaborative filtering is combined with the latent factor model (LFM) introduced by Koren in [13], the collaborative filtering is further successful. To some extent, collaborative filtering can be explained according to the principle of algorithm design. For example, users-based collaborative filtering can be interpreted as: the recommended items are the items that users like similar to the user. Item-based CF can be interpreted as: the recommended items are similar to the items the user like. However, compared with many content-based recommendation algorithms, it lacks interpretability. In addition, a user often only comments on a few movies, and CF will face the problem of cold start.

### ***2.3 Content Based Recommendation***

Content based recommendation attempts to model users and items with various available content information. The content-based recommendation includes the content-based collaborative filtering system proposed by Ricci [14] and Pazzani [15]. Although content-based recommendation has better interpretability, it is a time-consuming task to collect the required content information under different background of recommendation.

### ***2.4 Recommendation Based on Knowledge Graph***

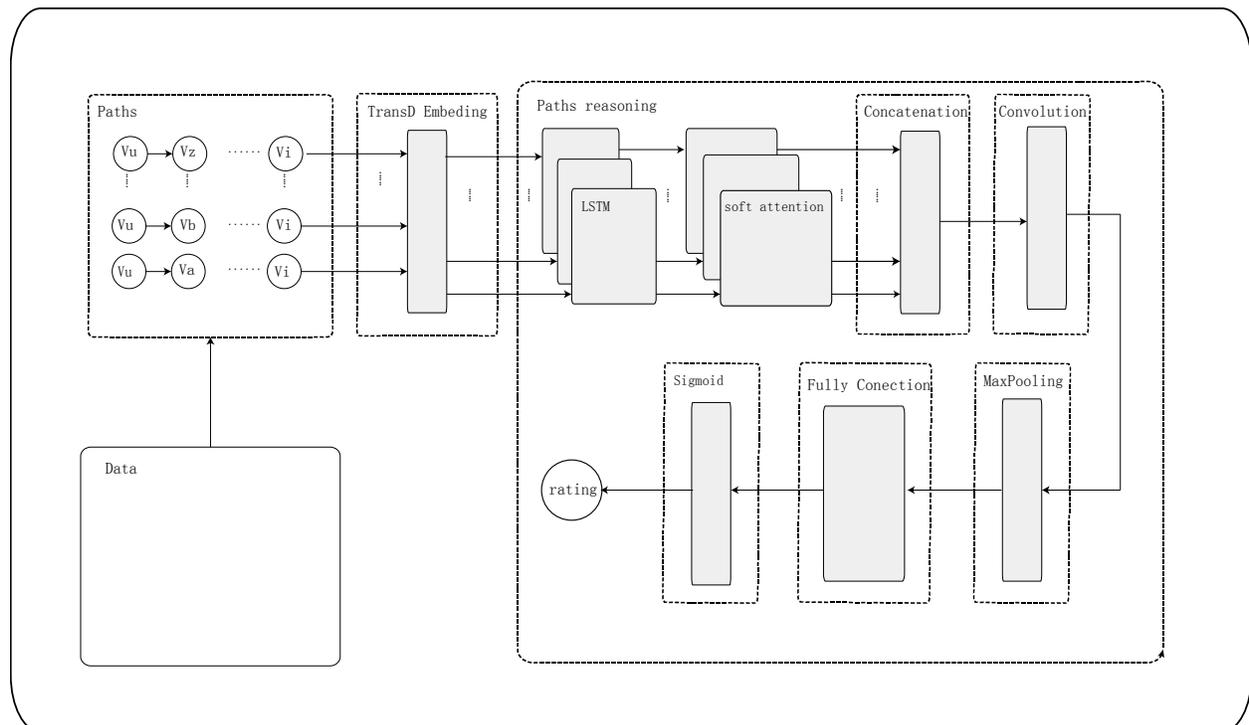
Knowledge graph contains rich information of users and items, which also provides more intuitive and targeted interpretation for the generation of recommended items. And building knowledge graph only needs to use the relationship between entities, which will greatly reduce the workload of content-based recommendation to extract content information.

The extended latent factor model [16], which represents different semantics according to the path connecting two entities, is introduced into the recommendation system, such as the Meta-Path in [17]. This method is helpful to infer user preferences based on the similarity of items, so as to generate effective recommendations. However, the Meta-Path method relies heavily on the characteristics of hand-made Meta-Path and requires additional domain knowledge. Moreover, the feature of manual meta-path is often incomplete, so it is difficult to cover all possible entity relationships, which hinders the improvement of recommendation quality. Compared with Meta-Path, knowledge graph representation learning can automatically learn the semantic embedding of entities in knowledge graph, and has better effect than Meta-

Path [6]. The study of knowledge graph-based representation learning has made great progress [18]. Among them, a series of translation representation learning algorithms are concerned. In [19], Borders proposed the TransE algorithm. The basic method of TransE is to give a triple  $(h, r, t)$ , and use the relation  $r$  vector as the translation between the head entity  $h$  vector and the tail entity  $r$  vector. In order to solve the limitation of transE model in dealing with 1-n, n-1, n-n complex relations, TransH [20] model uses both translation vector and normal vector of hyperplane to express the relation  $r$ . Although the TransH model makes each entity have different representations under different relationships, it still assumes that entities and relationships are in the same semantic space, which limits the representation ability of TransH to some extent. TansR [21] model holds that an entity is a complex of multiple attributes, and different relationships focus on different attributes of entities. TransR believes that different relationships have different semantic spaces. For each triplet, the entity should be projected into the corresponding relationship space, and then the translation relationship from the first entity to the last entity should be established. In order to solve the problem that too many parameters in TansR may differ greatly from the types or attributes of the head and tail entities of a relationship, Ji et al. [22] Proposed the TransD model, which set up two projection matrices that project the head entity and the tail entity into the relationship space respectively.

### 3 The Proposed KGRS

The KGRS proposed in this paper combines the knowledge graph and deep learning method, first obtains the reasoning path of the knowledge graph, then uses the knowledge graph representation learning TransD method to turn the entities in the reasoning paths into vectors, and finally uses the deep learning to capture the reasoning semantics of the paths to obtain the prediction ratings. The whole model framework is shown in Fig. 1.



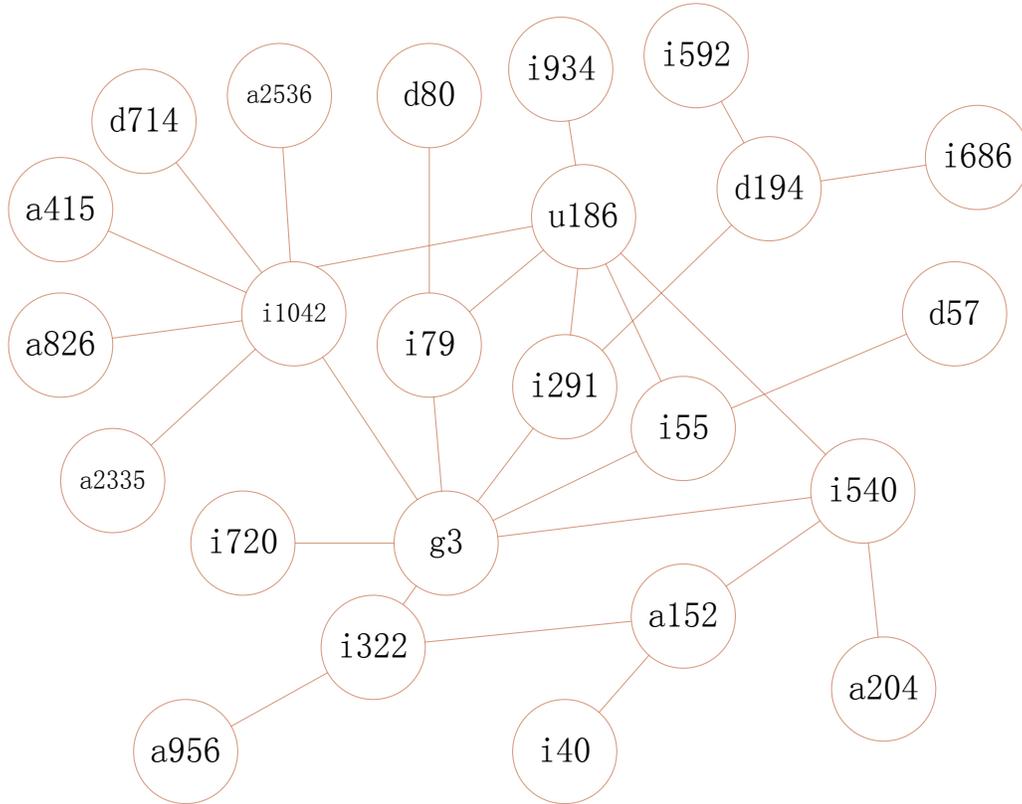
**Figure 1:** Framework of the KGRS model

#### 3.1 The Reasoning Paths of Interpretability

We use all triples  $(h, r, t)$  to construct the graph, and then use the method of constructing graph to search the reasoning paths in the knowledge graph. Specifically, entities are mapped to vertices in the graph,

and relationships are mapped to edges in the graph. By specifying two entities to traverse the whole graph, all paths of the two entities can be obtained. These paths are the reasoning paths between two entities. Fig. 2 is a partial diagram.

The reasoning path “u186→i540→g3→i322” can indicate that the user u186 likes the movie i540, the movie i540 belongs to g3 genre, and the movie i322 also belongs to g3 genre, so the user u186 may like the movie i322; the reasoning path “u186→i540→a152→i322” can indicate that the user u186 likes the movie i540, the movie i540 have been acted by a152, and the movie i322 has also been played by a152, so the user u186 can also like the movie i322. This example emphasizes that different paths connecting the same entity pair usually have different semantic relationships. Generally, they are of different importance in describing the taste of the user for the item. The importance of these paths is distinguished and the prediction rating is obtained by synthesizing the reasoning results of these paths. This reasoning process has better interpretability.



**Figure 2:** A partial diagram of knowledge graph

KGRS model is to use deep learning to simulate the reasoning process of these paths, and then distinguish the importance of these paths, and finally integrate the reasoning semantics of these paths to get the prediction ratings.

### 3.2 Reasoning Paths Embedding Generation Based on TransD

The great breakthrough of deep learning in natural language processing, image processing and other fields is largely due to its strong representation learning ability in the field. Therefore, we get the knowledge graph reasoning path embedding based on the knowledge representation learning TransD algorithm. Given the triples (h, r, t) in TransD, two projection matrices  $M_{rh}$  and  $M_{rt}$  sum of head entities and tail entities are respectively set in TransD model. The specific definitions are as follows:

$$M_{rh} = I_r I_{h_p} + I^{d \times k} \quad (1)$$

$$M_{rt} = I_r I_{t_p} + I^{d \times k} \quad (2)$$

where  $\mathbf{l}_h, \mathbf{l}_t \in \mathbb{R}^d, \mathbf{l}_r \in \mathbb{R}^k$ , the subscript p represents that the vector is a projection vector. Obviously, it is related to the entities and relationships. Finally, the following loss functions are defined in the TransD model:

$$f_r(h, t) = \|\mathbf{l}_h \mathbf{M}_{rh} + \mathbf{l}_r + \mathbf{l}_t \mathbf{M}_{rt}\|_{L_1/L_2} \quad (3)$$

Through TransD training for all triples, each entity K can be represented as a vector  $\mathbf{e}_k$ . The embedded representation of the m-th path of entity pair u-i is shown in Eq. (4).

$$\mathbf{path}_{uim} = [\mathbf{e}_{uim1}, \mathbf{e}_{uim2}, \dots, \mathbf{e}_{uimn}, \dots] \quad (4)$$

where  $\mathbf{e}_{uimn}$  is the embedding vector representing the n-th entity in the m-th path from user u to item i.

### 3.3 Recommendation Generation by Hybridizing TransD and LSTM

In this part, KGRS first captures the reasoning semantics of paths through LSTM (long short term memory) [23] and soft attention [24], then distinguishes the importance of different reasoning semantics of paths through convolution and maxpooling operations. Finally, it uses the full connection aggregation pooling vector to generate the prediction ratings through sigmoid function.

There is a reasoning path “u186→i540→g3→i322”, which means that user u186 likes movie i540, movie i540 belongs to g3 genre, movie i322 also belongs to g3 genre, so user u186 may also like movie i322. The reasoning process is obtained step by step, and the semantics of reasoning can be captured by RNN (recurrent neural network). We choose the improved network model of LSTM which effectively solves RNN gradient disappearance and gradient explosion.

In order to avoid over-fitting, we use only one LSTM capture semantics for all paths. When the embedded path  $\mathbf{path}_{uim}$  vector is input into LSTM, the reasoning semantics of every iteration in LSTM reasoning can be obtained, and the output semantics of entity to the m-th path of entity pair u-i, can be expressed as Eq. (5).

$$\mathbf{h}_{uim} = [\mathbf{h}_{uim1}, \mathbf{h}_{uim2}, \dots, \mathbf{h}_{uimn}, \dots] \quad (5)$$

In order to make the semantics of the final reasoning more relevant to the reasoning semantics of each iteration in the reasoning process, we use the attention mechanism. We chose the soft attention mechanism. In this layer, we can get the influence degree of the semantics of each iteration of LSTM on the semantics of the last iteration, and summarize the final semantics of this path. Suppose there is the m-th path of entity pair u-i, and the attention weight  $W_{uimn}$  of the n-th iterative reasoning semantics of LSTM can be obtained by Eq. (6) and Eq. (7).

$$s_{uimn} = \mathbf{h}_{uiml} \cdot \mathbf{h}_{uimn} \quad (6)$$

$$W_{uimn} = \frac{\exp(s_{uimn})}{\sum_{k=1}^l \exp(s_{uimk})} \quad (7)$$

The final semantics of this path after being processed by soft attention can be obtained by Eq. (8).

$$\mathbf{pr}_{uim} = \sum_{n=1}^l \mathbf{h}_{uimn} W_{uimn} \quad (8)$$

In order to distinguish the different importance of path reasoning semantics, we choose convolution operation and pooling operation. First, we combine the reasoning semantic vectors of all paths to form the whole reasoning semantics. Then the key semantic information is obtained by convolution operation according to different modes. Finally, in order to obtain the most important or comprehensive features of vectors, we choose the pooling operation. Experiments show that maxpooling is better than avgpooling. We choose maxpooling pool operation. Finally, through the full connection layer to gather the pooling vectors of all paths, and through sigmoid function to generate the prediction rating  $KGRS_u$  of entity pair u-i.

The results can be obtained from Eq. (9), where  $cnv$ ,  $mpooling$  and  $con$  of  $cnv\_mpooling\_con$  represent convolution operation, maxpooling operation and full connection operation, respectively.

$$KGRS_{ui} = \frac{1}{1 + e^{-cnv\_mpooling\_con([pr_{ui}, \dots, pr_{in}, \dots])}} \quad (9)$$

## 4 Experiment and Analysis Results

### 4.1 Dataset

The dataset used in this article is the movielens-100k dataset. On this basis, we crawled to the movie genre, director, actor information on IMDB. The crawled information is used as the auxiliary information of the movie. The auxiliary information expands the knowledge base, so that the knowledge graph can get better performance.

The movielens-100k dataset contains 943 users, 1682 movies and 100000 ratings. After removing the movies not on IMDB, the data is shown in Tab. 1. After mapping these data to the knowledge graph, there are 7746 entities, 8 relationships and 202183 triples. The triples do not contain the triples of the relationship between the positive sample set in the test set.

**Table 1:** All data removes the data after the movie is not on IMDB

Data name	Number
users	943
movies	1675
ratings	99975
directors	1154
actors	3948
genres	26

### 4.2 Data Partitioning

In our experiment, we did not consider the rating value, that is, if the user has a rating for a movie. We assume that the user likes the movie, and the rating is set to 1. Otherwise, the movie rating set to 0.

In order to make the experiment comparable, we use the same training set and test set for all models. The positive samples of the training set and the test set are obtained by splitting 99975 rating data with auxiliary information in a ratio of 4:1. The positive sample of the test set is used to check whether the items in the recommended list generated by KGRS are accurate.

For the selection of negative samples in the training set, we randomly select movies that users have no rating. In order to make KGRS learn more negative sample information, the negative samples count we extract for each user is 120% of the positive samples count in the training set.

Because some negative samples are randomly selected as the training set, if these negative samples are not included in the test set. Compared with other methods, the test set may contain fewer negative samples and the prediction results are not true. In order to avoid such a situation, we will use all samples except the positive samples in the training set as the test set.

### 4.3 Path Extraction

The paper [17] has proved that the shorter path is more important to the recommendation result, and the noise will be introduced when the path length exceeds 5. In order to speed up the process of path extraction and training, this paper only mines 5 shortest paths for each entity pair  $u-i$ , that is, 5 paths with length of 3 at most.

#### 4.4 Evaluating Indicator

The evaluation indicators we used include  $\text{precision}@N$  and  $\text{MMR}@N$ .  $\text{precision}@N$  which represent the mean value of the probability that the first N items of each user's recommendation list appear in the positive samples of the test set. The N value of  $\text{precision}@N$  includes 1, 5 and 10.

$\text{MMR}@N$  definition is shown in Eq. (10).

$$\text{MMR}@N = \frac{1}{N} \sum_{i=1}^N \sum_{v_j \in \text{test}(u_i)} \frac{1}{\text{rank}(u_i, v_j)} \quad (10)$$

where we set n to 10,  $v_j$  is the item that appears correctly in the top-N recommendation list in the positive sample of the test set.  $\text{rank}(u_i, v_j)$  represents the number of  $v_j$  positions in the user  $u_i$  recommendation list.

#### 4.5 Model Setting

Considering the uncertainty of the number of all user-item paths, using the same pooling window and pooling step size in pooling operation will cause data dimension inconsistency. Therefore, it is necessary to dynamically adjust the pooling window and the pooling step size according to the number of paths, both of which are (paths\_size, 1).

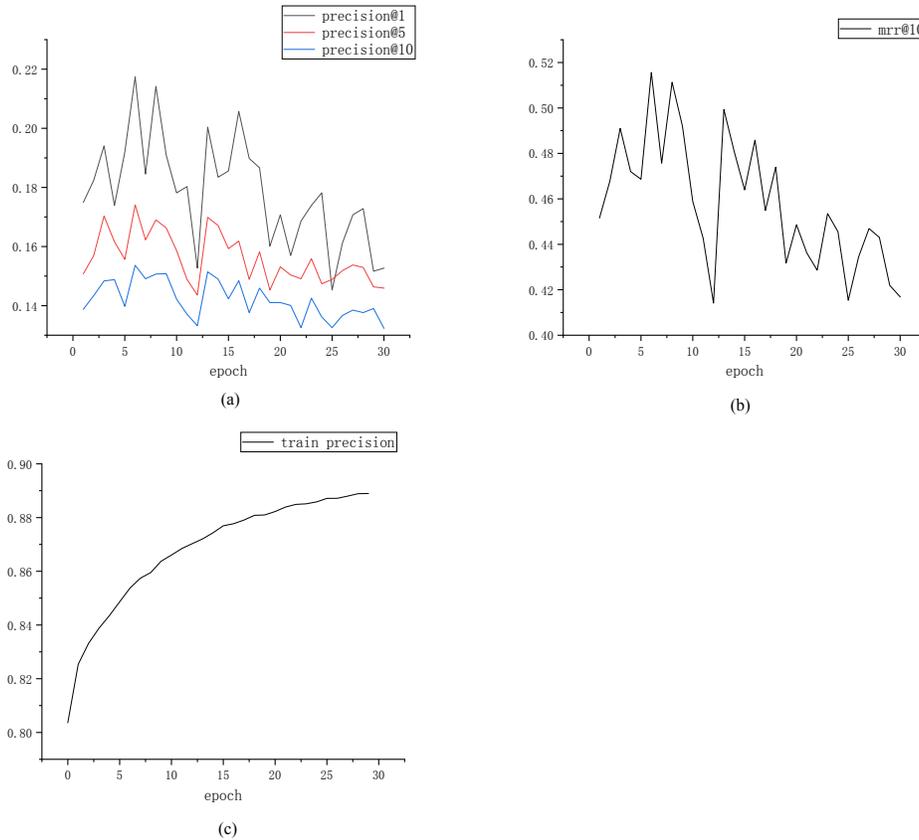
The number of hidden units of LSTM is set to 64, number of convolution kernels set to 1, the size of convolution kernel is set to 1x5, TransD embedding size set to 100 dimensions and the learning rate is set to 0.1. The optimization method adopts SGD.

#### 4.6 Experimental Results and Analysis

The comparative baseline models include GraphLF (graph latent factor) [25], CKE (collaborative knowledge base embedding) [26], and the recently advanced interpretable recommendation methods: RKGE model, RppleNet. The experimental results are shown in Tab. 2. It can be seen that the recommended accuracy of KGRS proposed in this paper is the highest among all comparison models. In the case of good interpretability at the same time, the recent advanced interpretable recommendation algorithms RippleNet and RKGE recommendation efficiency is far lower than KGRS. It is proved that our model is advanced. Compared with the related models, KGRS increased 37.59%, 26.25%, 18.23% and 25.51% respectively in  $\text{precision}@1$ ,  $\text{precision}@5$ ,  $\text{precision}@10$  and  $\text{MRR}@10$ . The interpretability of KGRS is well described in Section 3. According to the above analysis, KGRS has good recommendation interpretation and higher recommendation accuracy.

**Table 2:** Experimental comparison result

Methods	GraphLF	CKE	RippleNet	RKGE	KGRS	Improve
$\text{precision}@1$	0.1069	0.0954	0.1548	0.1580*	0.2174	37.59%
$\text{precision}@5$	0.0360	0.0781	0.1379*	0.1094	0.1741	26.25%
$\text{precision}@10$	0.0581	0.0682	0.1300*	0.0956	0.1537	18.23%
$\text{MRR}@10$	0.1524	0.2440	0.4108*	0.3399	0.5156	25.51%



**Figure 3:** (a) Change of precision@N during KGRS training; (b) Changes of MRR@10 during KGRS training; (c) The change of training set accuracy in KGRS training

The training process of KGRS is shown in Fig. 3. It can be found that MRR@10 and precision@N are the best when epoch is about 6. After that, the accuracy of the training set is still rising, while the accuracy of the test set fluctuates downward, and begins to over-fitting.

## 5 Conclusion

KGRS is proposed in this paper. In KGRS, the reasoning in the knowledge graph is simulated by deep learning to get the prediction rating, and then the final recommendation list is obtained by ranking a user's preference from high to low. The experimental results show that compared with the recently advanced interpretable models RKGE and RippleNet, KGRS has higher recommendation accuracy and advanced nature. In the future, we will try to add information such as movie posters into the recommendation system.

**Acknowledgement:** I am deeply thanking to my tutors and my parents for their continuous support and encouragement.

**Funding Statement:** This work has been supported by the National Science Foundation of China Grant No. 61762092, "Dynamic multi-objective requirement optimization based on transfer learning", No. 61762089, "The key research of high order tensor decomposition in distributed environment", and the Open Foundation of the Key Laboratory in Software Engineering of Yunnan Province, Grant No. 2017SE204," Research on extracting software feature models using transfer learning".

**Conflicts of Interest:** We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

## References

- [1] W. Y. Zhu, W. Liu, Z. T. Liu, "Personalized news recommendation based on event ontology," *Computer Engineering*, vol. 45, no. 6, pp. 267-272 + 279, 2019.
- [2] T. H. Li, Y. Tang, B. Liu, "Multiple interactive hybrid recommendation model based on deep learning," *Computer Engineering and Application*, vol. 55, no. 01, pp. 141-147, 2019.
- [3] M. D. Ekstrand, "Collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5-53, 2007.
- [4] Y. Zhang, X. Chen, "Explainable recommendation: a survey and new perspectives," *Information Retrieval*, 2018.
- [5] S. P. Guan, X. L. Jin, Y. T. Jia, Y. Z. Wang, X. Q. Cheng, "Research progress of knowledge reasoning oriented to knowledge map," *Journal of Software*, vol. 29, no. 10, pp. 2966-2994, 2018.
- [6] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L. Huang et al., "Recurrent knowledge graph embedding for effective recommendation," *Conference on Recommender Systems*, pp. 297-305, 2018.
- [7] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li et al., "RippleNet: propagating user preferences on the knowledge graph for recommender systems," in *the 27th ACM International Conference*, 2018.
- [8] R. Salakhutdinov, A. Mnih, G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *International Conference on Machine Learning*, pp. 791-798, 2007.
- [9] P. Huang, X. He, J. Gao, L. Deng, A. Acero et al., "Learning deep structured semantic models for web search using clickthrough data," in *Proc. of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, 2013.
- [10] X. He, L. Liao, H. Zhang, L. Nie, X. Hu et al., "Neural collaborative filtering," *Information Retrieval*, 2017.
- [11] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *ACM Conference on Computer Supported Cooperative Work*, 1994.
- [12] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *International Conference on World Wide Web*, 2001.
- [13] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, USA, 2008.
- [14] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, "*Recommender Systems Handbook*," Springer US, 2011.
- [15] M. J. Pazzani, D. Billsus, "Content-based recommendation systems," in *Adaptive Web*, pp. 325-341, Springer-Verlag, 2007.
- [16] Y. Shi, M. Larson, A. Hanjalic, "Collaborative filtering beyond the User-Item matrix: a survey of the state of the art and future challenges," *ACM Computing Surveys*, vol. 47, no. 1, pp. 1-45, 2014.
- [17] Y. Sun, J. Han, X. Yan, P. Yu, T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proc. of the VLDB Endowment*, vol. 4, no. 11, pp. 992-1003, 2011.
- [18] Z. Liu, M. Sun, T. Lin, R. Xie, "Research progress of knowledge representation learning," *Computer Research and Development*, vol. 53, no. 2, pp. 247-261, 2016.
- [19] A. Bordes, N. Usunier, A. Garciaduran, J. Weston, O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *International Conference on Neural Information Processing Systems*, 2013.
- [20] J. Zhang, "Knowledge graph embedding by translating on hyperplanes," in *Association for the Advancement of Artificial Intelligence*, 2015.
- [21] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- [22] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Meeting of the Association for Computational Linguistics & the International Joint Conference on Natural Language Processing*, 2015.
- [23] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [24] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville et al., "Show, attend and tell: neural image caption generation with visual attention," in *International Conference on Machine Learning*, pp. 2048-2057, 2015.

- [25] R. Catherine, W. W. Cohen, “Personalized recommendations using knowledge graphs: a probabilistic logic programming approach,” in *Conference on Recommender Systems*, pp. 325-332, 2016.
- [26] F. Zhang, N. J. Yuan, D. Lian, X. Xie, W. Ma, “Collaborative knowledge base embedding for recommender systems,” in *Knowledge Discovery and Data Mining*, pp. 353-362, 2016.