Knowledge Graph Bidirectional Interaction Graph Convolutional Network for Recommendation

Zengqiang Guo[1], Yan Yang[1,2](✉), Jijie Zhang[1], Tianqi Zhou[1], and Bangyu Song[1]

[1] School of Computer Science and Technology, Heilongjiang University, Harbin, China
2201831@s.hlju.edu.cn, yangyan@hlju.edu.cn
[2] Key Laboratory of Database and Parallel Computing of Heilongjiang Province, Harbin, China

Abstract. Recently, the recommended method based on the Knowledge Graph (KG) has become a hot research topic in modern recommendation systems. Most researchers use assistive information such as entity attributes in KG to improve recommendation performance and alleviate Collaborative Filtering (CF) sparsity and cold start problems. The most recent technical trend is to develop end-to-end models based on the Graph Convolutional Network (GCN). In this paper, we propose a Knowledge Graph Bidirectional Interaction Graph Convolution Network for recommendation (KBGCN). This method is used to refine the embedded representation of node by recursively delivering messages from the neighbors (attributes or items) of the node (entity) and applies the knowledge aware attention mechanism to distinguish the contributions of different neighbors based of the same node. It uses neighbors of each entity in KG as the view of this entity, which can be extended by expanding the view of Multi-hop neighbors to mine high-order connectivity information existing in KG automatically. We apply the proposed method to three real-world datasets. KBGCN is better than seven KG-based baselines in recommendation accuracy and the two state-of-the-art GCN-based recommendations frameworks.

Keywords: Recommender systems · Knowledge graph · Graph convolutional networks

1 Introduction

With the success of the recommendation system, people can search for a large number of interesting content in various applications on the network, such as searching for commodities [12] on E-commerce platforms, news [23] on news portals, and movies [3] on video websites. Early recommendation systems were devoted to collaborative filtering [5,7]. Product based on the ID of the user and

E. Pimenidis et al. (Eds.): ICANN 2022, LNCS 13530, pp. 532–543, 2022.
https://doi.org/10.1007/978-3-031-15931-2_44

the item itself or trained through the neural network. The main disadvantage of collaborative filtering [2,4,24] is that it cannot effectively model user's and items' assistive information (attributes and context). Researchers have turned to bring recommendations into richer scenarios in recent years, such as building Knowledge Graph. The assistive information in KG can be used to model the user and item sides. Compared with the KG free method, the KG method can deeply mine the high-order information of users and items, and the path information based on KG improves the accuracy of the recommendation system. But at the same time, the heterogeneity and high dimensionality of the KG make it difficult to combine with the recommendation system. A feasible method is to use the *Knowledge Graph Embedding* (KGE) [16] method to pretrain the KG to obtain the embedding vector. For example, CKE [22] uniformly collects various types of assistive information such as entity attributes and text information and uses the TransE [1] algorithm to encode latent vectors and extract features. MKR [14] combines KGE and recommendation to learn the potential representation of items on the one hand and the semantic matching of entities related to items on the other hand. However, KGE is more often used in graph classification tasks, such as link prediction. HeteRec [21] uses meta path similarity to enrich the user-item interaction matrix so as to extract a more comprehensive representation of users and items. Both the above embedding-based and path-based approaches utilize only one aspect of the information in the graph. The idea of embedding-based propagation that combines the semantic representation of relationships and connection information is proposed to make full use of the information in KG.

In this paper, inspired by the idea of message passing in GCN, We propose Knowledge graph Bidirectional interaction Graph Convolutional Network for recommendation (KBGCN). Our purpose is to automatically capture the high-order connected information in the KG to enrich the representation of entities. The core idea of KBGCN is to have biased aggregation of neighborhood information in the knowledge graph, learn item representations of distant neighbors inward, and aggregate higher-order information using a bidirectional interaction module. We release the codes and datasets at https://github.com/zq816/KBGCN.

Our contributions in this paper are summarized as follows:

- We propose a simple yet effective GCN-based knowledge graph recommendation framework. We use the bidirectional interaction which can better aggregate neighborhood information to obtain good entity representation.
- We design a novel neural knowledge aware attention mechanism to learn the knowledge-based weights of entities in the same entity set and generates weighted representation of entities.
- Experimental results on three public benchmark datasets have demonstrated that the KBGCN outperforms recent state-of-the-art models.

2 Related Work

Knowledge Graph based RS (KGRS) usually constructs KG based on external knowledge (such as edge information), explores the implicit or high-order connection relationship between users or items. Due to the use of assistive knowledge, KGRS can better understand user behaviour and item characteristics. RippleNet [12] is the first method to use preference propagation. It uses KG to spread information and get the embedded representation of high-order neighbors. The AKUPM [10] model is combined with the TransR [6] algorithm to simulate users' click history and spread users' preferences for different entities. Like RippleNet, our work can be viewed as a KG-based neighborhood propagation method.

Graph Convolution Network (GCN) usually learns how to use graph structure and node feature information and uses neural network to iteratively aggregate feature information from local graph neighborhoods. For example, GCN are used for influence diffusion on social graphs in social recommendations [19]. Mining the user-item connection information hidden in the user-item interaction graph to alleviate the problem of data sparsity in collaborative filtering [11]. KGCN [15] first samples the neighbors of the candidate items in KG and then aggregates the information of multi-hop neighbors and propagates it inward to the candidate items. KGNN-LS [13] further adds a label smoothing (LS) mechanism to the KGCN model. Our method also connects to KGCN [15] and CKAN [18]. But the major difference between our work and the literature is the use of a bidirectional interaction method to aggregate entity neighbors and the design of a novel neural knowledge-aware attention mechanism to learn knowledge-based entity weights.

3 Methodology

We introduce the framework of the proposed KBGCN in detail. As shown in Fig. 1, the model framework consists of four main parts: (1) Embedded Layer. (2) Knowledge Graph Awareness Propagation Layer. This is divided into neighborhood view propagation and knowledge awareness attention embedding. (3) BI-Interactional Aggregation Layer. (4) The Prediction Layer.

3.1 Problem Formulation

In a typical recommendation scene, we have a set of users $U = \{u_1, u_2, u_3 \dots u_m\}$, and a set of items $V = \{v_1, v_2, v_3 \dots v_n\}$. We get the user-item interaction matrix $Y (Y \in R^{m \times n})$ based on user's implicit feedback (click or purchase), the user's implicit feedback is defined as $Y_{uv} = \{y(u, v)|u \in U, v \in V\}$, where $y(u, v)$ equals 0 or 1. Knowledge Graph G consists of triples $\{(h, r, t)|t \in E, r \in R\}$. for example, in the triplet (The Avengers, film.film.actor, Robert Downey Jr.), the actor of The Avengers is Robert Downey Jr. In most recommended scenes, items are represented in KG with the corresponding entities. Our goal is to learn a prediction function $\hat{y}_{uv} = F(u, v)$ to predict whether user u will click the item i that has never been clicked, where $\hat{y}_{uv}$ is a probability value.

$$Y_{uv} = \begin{cases} 1 & \text{if (u,v) has an interaction} \\ 0 & \text{else} \end{cases} \tag{1}$$

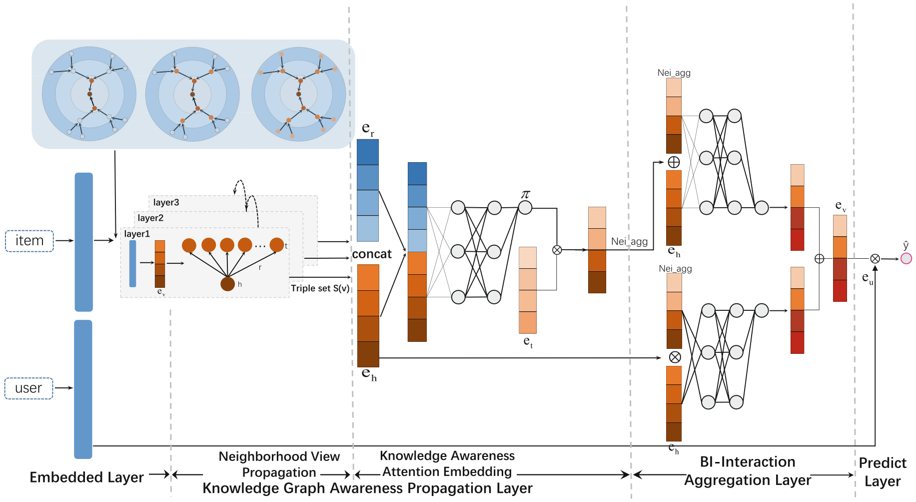


Fig. 1. The KBGCN framework consists of four main parts: (1) Embedded layer (2) Knowledge graph awareness propagation layer (3) BI-Interaction aggregation layer (4) Prediction layer.

3.2 Embedded Layer

We take the one-layer structure as an example. Assuming a pair of user-item pairs (u, v), we use $p(v)$ to represent all the direct neighbors of item v, $r(a, b)$ represents the relationship between a and b. Firstly, initialize the user embedding vector $e_u \in R^d$, the entity embedding vector E and relationship embedding vector R in the knowledge graph G, the initialization embedding vector of the item is mapped from the knowledge graph G.

$$e_v = E * v_v, \tag{2}$$

where $E \in R^d$ is the entity embedding matrix, v_v is the ID representation of item v, and $e_v \in R^d$ represents the initial embedding representation of item v.

3.3 Knowledge Graph Awareness Propagation Layer

As shown in Fig. 1, the Knowledge Graph Awareness Propagation Layer is mainly composed of two modules: Neighborhood View Propagation for entities seeking information about their neighbors in the view. Knowledge Awareness Attention Embedding provides a novel neural knowledge aware attention mechanism to learn the knowledge-based weights of entities in the view.

Neighborhood View Propagation. In the real world KG, each entity will not have only one neighbor node. That is, $P(v)$ is greater than or equal to 1. In order to ensure the computational efficiency and space utilization of the model, the number of neighbors of each entity takes a fixed size during model training. We set the number of neighbors K as a hyperparameter.

$$N(v) = \{e_n | e_n \in p(v) \text{ and } |N(v)| = K\}, \tag{3}$$

$$S(v) = \{(e_h, e_r, e_t) | e_h = e_v, e_r \in R, e_t \in N(v) \text{ and } (e_h, e_r, e_t) \in G\}, \tag{4}$$

where e_n is the neighbor entity, K is the number of neighbors sampled each time, and $N(v)$ is the neighbor sampling set of item v. $S(v)$ is a set of triples, and the size is also K.

Knowledge Awareness Attention Ambedding. When the same head entity is connected with different tail entities, each tail entity may have different meanings according to the different relationship r. for example, The Legend Of 1900 and Rob Roy have similarities in terms of directors and actors, the similarity in topic materials or types is basically zero, which leads to users choosing different items in different relationship spaces. Therefore, We propose a knowledge aware attention embedding method to reveal the different meanings of tail entities in different head relationships through different attention weights.

$$A_i = \pi\left(e_i^h, e_i^r\right) e_i^t, \tag{5}$$

where e_i^h is the header entity embedding, e_i^r is the relational embedding, e_i^t and A_i are the tail entity embedding and weighted representation of the $i-th$ triplet. π is a matrix controlling the impact of the weights generated by the head entity and related entity on the tail entity. We design such a neural network function similar to the attention mechanism to realize π :

$$c_0 = \mathrm{ReLU}\left(w_0\left(e_i^h \| e_i^r\right) + b_0\right), \tag{6}$$

$$\pi\left(e_i^h, e_i^r\right) = \sigma\left(w_2 \,\mathrm{ReLU}\left(w_1 c_0 + b_1\right) + b_2\right), \tag{7}$$

we use a three-layer depth neural network model to generate the weight of the tail entity. e_i^h and e_i^r are first concat operated, the activation function is relu, and the last layer is sigmoid. $w_0 \in R^{d\times 2d}$, $w_1 \in R^{2d\times 2d}$ and $w_2 \in R^{2d\times d}$ are all trainable parameters. Finally, we normalize the weight score:

$$\pi\left(e_i^h, e_i^r\right) = \frac{\exp\left(\pi\left(e_i^h, e_i^r\right)\right)}{\sum_{(h',r',t')\in\mathcal{S}_{(v)}} \exp\left(\pi\left(e_i^{h'}, e_i^{r'}\right)\right)}. \tag{8}$$

Finally, we sum the representations of the K triples obtained:

$$neighbor_{agg} = \sum_{i=1}^{K} A_i. \tag{9}$$

3.4 BI-Interaction Aggregation Layer

The final stage is to reaggregate the header entity and neighborhood representation into a new entity representation, that is, the final representation of the item. Here, we propose a new aggregator, the BI-Interactional aggregator:

$$\begin{aligned} f_{\text{BI-Interactional}} &= \text{LeakyReLU}\left(w_3\left(e_h + neighbor_{agg}\right)\right) \\ &\quad + \text{LeakyReLU}\left(w_4\left(e_h \odot neighbor_{agg}\right)\right), \end{aligned} \tag{10}$$

where $w_3 \in R^{d\times d}$ and $w_4 \in R^{d\times d}$ is the trainable parameter matrix, and $\odot$ is the element product. The BI-Interactional aggregator consists of two parts: the sum of head entity and neighborhood information and the other part is the element product of head entity and neighborhood information. Adding the two parts, we realize bidirectional interaction, breaking the previous aggregation methods such as concat or addition. Achieved a significant improvement.

3.5 Predict Layer

$$\hat{y}_{uv} = e_u^{\top} f_{\text{BI-Interactional}}, \tag{11}$$

where e_u and $f_{\text{BI-Interactional}}$ are the final vector representation of user and item, and $\hat{y}_{uv}$ is the prediction probability value.

3.6 Optimization

We train all user-item pairs. During the training, we adopt the negative sampling strategy, and the loss function is as follows:

$$\mathcal{L} = \sum_{u\in U}\left(\sum_{v:y(u,v)=1} \mathcal{J}\left(y_{uv}, \hat{y}_{uv}\right) - \sum_{i=1}^{N^u} \mathcal{J}\left(y_{uv_i}, \hat{y}_{uv_i}\right)\right) + \lambda\|\mathcal{W}\|_2^2, \tag{12}$$

where $\mathcal{J}$ is the cross entropy loss, $N^u = |v : y(u,v) = 1|$ is the number of user negative samples, and the last item is L2-regulator.

4 Experiments

In this section, we evaluate the performance of KBGCN through three real-world datasets.

4.1 Datasets

- **MovieLens-20M** is a data set widely used in the field of movie recommendation, including 138000 users and 20 million ratings for 27000 movies. The rating is 1–5.

- **Book-Crossing** contains more than 1 Million Book rating information. The rating ($'Book - Rating'$) is either explicit, expressed in a rating of 1–10, or implicit, expressed in 0.
- **Last.FM** contains 92,800 artist recordings from 1,872 users.

The basic information of the three datasets is shown in Table 1:

Table 1. Basic statistics for the three datasets

		MovieLens-20M	Book-Crossing	Last.FM
User-Item interaction	#Users	138159	17860	1872
	#Items	16954	14967	3846
	#Interactions	13501622	139746	42346
Knowledge graph	#Entities	102569	77903	9366
	#Relations	32	25	60
	#Triplets	499474	151500	15518

4.2 Baselines

- **PER** [20] is a path based method that uses the relationship heterogeneity to disperse user preferences to different meta paths in the information network.
- **CKE** [22] introduces structural information, text data, image data and other information in the knowledge base to improve the quality of the recommendation system.
- **BPRMF** [9] is a classical CF method, which proposes the sampling of positive and negative samples (triples) in the recommendation field, and then trains the model through BPR loss.
- **LibFM** [8] + **TransE** [1] is a feature-based factorization model for CTR scene takes user ID, item ID and corresponding entities learned through TransE algorithm as the input of the model.
- **RippleNet** [12] is a classic recommendation method based on propagation, which enriches the user representation by propagating the user's potential preferences in the knowledge graph.
- **KGCN** [15] applies the graph convolutional networks to the recommendation system, aggregates the neighborhood information with deviation in combination with the knowledge graph, and refines the embedding of items.
- **KGNN-LS** [13] transforms heterogeneous KG into user specific weighted graph, and adds label smoothing mechanism to KGCN framework to calculate personalized item embedding.
- **KGAT** [17] is a state-of-the-art propagation-based method that collectively refines embeddings of users, items, and KG entities via training interaction and KG embeddings jointly.
- **CKAN** [18] is another state-of-the-art propagation-based method employing a heterogeneous propagation strategy to encode diverse information for better recommendation.

4.3 Experiments Settings

In the experiment, we divide the data set according to the ratio of 6:2:2. We provide two actual recommended scenario predictions: (1) CTR prediction: we use AUC and F1 scores as the evaluation indexes. (2) Top-K recommendation, we use the recall@k Index evaluation model. Adam optimizer is used to optimize all trainable parameters. The hyperparameters are set as shown in Table 2, where d is the feature vector dimensions, K is the number of randomly sampled neighbor nodes, H is the view size in the neighborhood aggregation, and BS is the batch size, η is learning rate, λ is L2 regularization coefficient.

Table 2. Hyperparameter settings for three datasets

Datasets	d	K	H	BS	η	λ
MovieLens-20M	32	4	1	65536	5×10^{-3}	1.8×10^{-7}
Book_Crossing	4	8	1	256	1.5×10^{-4}	4.5×10^{-5}
Last.FM	8	8	1	128	2×10^{-4}	1×10^{-4}

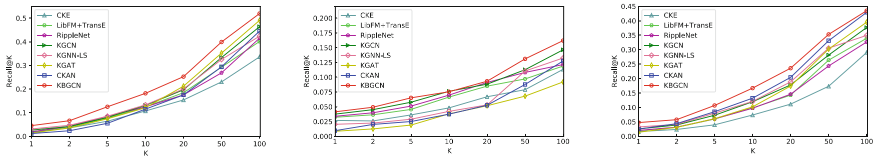
4.4 Results

Table 3 and Fig. 2 show the results of CTR prediction and Top-K recommendation for two recommended scenarios respectively. We have the following conclusions:

Table 3. The results of AUC and $F1$ in CTR prediction

Model	MovieLens-20M		Book-Crossing		Last.FM	
	AUC	F1	AUC	F1	AUC	F1
PER	0.832	0.788	0.617	0.562	0.633	0.596
CKE	0.924	0.871	0.677	0.611	0.744	0.673
BPRMF	0.958	0.914	0.658	0.611	0.756	0.701
LibFM+TransE	0.966	0.917	0.659	0.622	0.777	0.709
RippleNet	0.968	0.912	0.672	0.625	0.780	0.702
KGCN	0.970	0.923	0.651	0.613	0.780	0.702
KGNN-LS	0.975	0.929	0.676	0.631	0.803	0.722
KGAT	0.976	0.928	0.652	0.619	0.805	0.725
CKAN	0.971	0.922	**0.711**	0.634	0.811	0.731
KBGCN	**0.983**	**0.942**	0.695	**0.660**	**0.825**	**0.747**

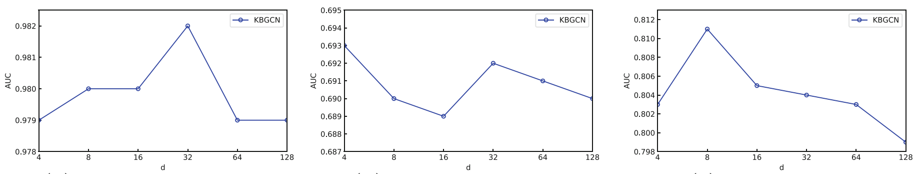
– Compared with these state-of-the-art baselines, KBGCN achieves competitive performance on all three datasets, especially in movie and music datasets, which shows that KBGCN is better at dealing with datasets with strong sparsity.



(a) MovieLens-20M (b) Book-Crossing (c) Last.FM

Fig. 2. The results of Recall@K in Top-K recommendation

- The performance of KBGCN compared to RippleNet demonstrates the effectiveness of the knowledge-aware attention mechanism. Comparative results with KGAT indicate the importance of explicitly bidirectional interactions.
- CKAN has a more prominent result on the Book-Crossing dataset. In the Top-K recommendation, when the k value is larger, KBGCN effect is equal to CKAN. It can be inferred that CKAN will introduce more noise when the dataset becomes larger and sparser, while KBGCN's bidirectional interaction aggregation method will be more effective.
- In the Top-K recommendation, no matter what the K value is, the experimental effect of KBGCN is superior to all baseline models. With the increase of the K value, the value of recall tends to rise. The *recall*@K of KBGCN on the three datasets is higher than KGAT and CKAN. This illustrates the effectiveness of bidirectional interactions aggregation and knowledge-aware attention mechanism.



(a) MovieLens-20M (b) Book-Crossing (c) Last.FM

Fig. 3. The influence of embedding vector dimensions d

The Influence of Embedding Vector Dimensions. As shown in Fig. 3, when the dimension is increased with a certain range in the lower dimension, the performance of KBGCN will improve, but when it exceeds a certain threshold, the performance will decline. We guess that too large d value will encode more entity information to a certain extent, resulting in a slight improvement in performance, but it will also cause overfitting.

The Influence of Neighbor Sampling Size. As shown in the Table 4. We find that the K value is relatively average in all datasets, and the promotion effect of the model is the best when it is in the middle position ($k = 4$ or 8). On the one hand, because the small number of samples can not provide sufficient information, the large number will bring unnecessary noise interference to the model.

Table 4. AUC result of KBGCN with different neighbor sampling size K.

K	2	4	8	16	32
MovieLens-20M	0.980	**0.983**	0.980	0.980	OOM
Book-Crossing	0.690	0.690	**0.695**	0.690	0.689
Last.FM	0.804	0.805	**0.825**	0.804	0.806

Table 5. AUC result of KBGCN with different view size H.

H	1	2	3	4
MovieLens-20M	**0.983**	0.978	0.977	0.899
Book-Crossing	**0.695**	0.689	0.689	0.678
Last.FM	**0.825**	0.795	0.594	0.586

The Influence of View Size. As shown in the Table 5. On the music dataset, the influence of the model on the view size fluctuates greatly. The model effect decreases sharply when the view size is high-level. With the increase of view size, the effect of other datasets also decreases to varying degrees. As the path becomes longer, the information too late is meaningless to the head entity but will mislead the original entity representation.

5 Conclusions

In this paper we propose knowledge graph bidirectional interaction graph convolutional network for recommendation. Through knowledge-based deep propagation and utilizes a knowledge-aware attention mechanism to discriminate the contribution of different knowledge-based neighbors. The knowledge based high-order interaction information of item is successfully captured, effectively improves the model's ability to represent item with latent vectors. Experimental results on three public benchmark datasets the KBGCN is superior to state-of-the-art models.

References

1. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: 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 (2013)
2. Cheng, H., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016, Boston, MA, USA, 15 September 2016, pp. 7–10 (2016)
3. Diao, Q., Qiu, M., Wu, C., Smola, A.J., Jiang, J., Wang, C.: Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, New York, NY, USA - 24–27 August 2014, pp. 193–202 (2014)
4. He, X., Chua, T.: Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, 7–11 August 2017, pp. 355–364 (2017)
5. Juan, Y., Zhuang, Y., Chin, W., Lin, C.: Field-aware factorization machines for CTR prediction. In: Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016, pp. 43–50 (2016)
6. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 25–30 January 2015, Austin, Texas, USA, pp. 2181–2187 (2015)
7. Rendle, S.: Factorization machines. In: ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14–17 December 2010, pp. 995–1000 (2010)
8. Rendle, S.: Factorization machines with libfm. ACM Trans. Intell. Syst. Technol. **3**, 57:1–57:22 (2012)
9. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–21 June 2009, pp. 452–461 (2009)
10. Tang, X., Wang, T., Yang, H., Song, H.: AKUPM: attention-enhanced knowledge-aware user preference model for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, 4–8 August 2019, pp. 1891–1899 (2019)
11. Wang, H., Lian, D., Ge, Y.: Binarized collaborative filtering with distilling graph convolutional network. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, 10–16 August 2019, pp. 4802–4808 (2019)
12. Wang, H., et al.: Ripplenet: propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, 22–26 October 2018, pp. 417–426 (2018)
13. Wang, H., et al.: 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, KDD 2019, Anchorage, AK, USA, 4–8 August 2019, pp. 968–977 (2019)

14. Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., Guo, M.: Multi-task feature learning for knowledge graph enhanced recommendation. In: The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, 13–17 May 2019, pp. 2000–2010 (2019)
15. Wang, H., Zhao, M., Xie, X., Li, W., Guo, M.: Knowledge graph convolutional networks for recommender systems. In: The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, 13–17 May 2019, pp. 3307–3313 (2019)
16. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**, 2724–2743 (2017)
17. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.: KGAT: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, 4–8 August 2019, pp. 950–958 (2019)
18. Wang, Z., Lin, G., Tan, H., Chen, Q., Liu, X.: 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, SIGIR 2020, Virtual Event, China, 25–30 July 2020, pp. 219–228 (2020)
19. Wu, L., Sun, P., Fu, Y., Hong, R., Wang, X., Wang, M.: A neural influence diffusion model for social recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, 21–25 July 2019, pp. 235–244 (2019)
20. Yu, X., et al.: Personalized entity recommendation: a heterogeneous information network approach. In: Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, 24–28 February 2014, pp. 283–292 (2014)
21. Yu, X., et al.: Recommendation in heterogeneous information networks with implicit user feedback. In: Seventh ACM Conference on Recommender Systems, RecSys 2013, Hong Kong, China, 12–16 October 2013, pp. 347–350 (2013)
22. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016, pp. 353–362 (2016)
23. Zheng, G., et al.: DRN: a deep reinforcement learning framework for news recommendation. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, 23–27 April 2018, pp. 167–176 (2018)
24. Zhou, G., et al.: Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, 19–23 August 2018, pp. 1059–1068 (2018)