Simplifying Knowledge-Aware Aggregation for Knowledge Graph Collaborative Filtering

Honghai Zhang, Yifan Chen, Xinyi Li, and Xiang Zhao(✉)

Laboratory for Big Data and Decision, National University of Defense Technology, Changsha, China
xiangzhao@nudt.edu.cn

Abstract. Incorporating knowledge graph (KG) for recommendation has been well considered in recent researches since it can alleviate the sparsity and cold-start problem of collaborative filtering. To capture the rich semantics of knowledge graph, existing KG-based models utilize graph neural networks (GNNs). However, we empirically find that the feature transformation and nonlinear activation designs in GNN contribute little to the recommendation performance. We propose *simplified knowledge-aware attention network* (SKAN) that simplifies the knowledge-aware aggregation by removing the two designs. To ensure the personalization during propagation, we apply weighted aggregation with user-specific attentions. We further aggregate the interacted items of users to enhance the user representation learning. We apply the proposed model on three real-world datasets, and the empirical results suggest that simplified knowledge-aware attention network (SKAN) significantly outperforms several compelling state-of-the-art baselines.

Keywords: Recommender system · Knowledge graph · GNNs

1 Introduction

Recommender systems deal with information overload by filtering out irrelevant information and providing only relevant information to users. They have been widely used in various scenarios, such as music, movie and power domain [6,18].

In recent years, in order to alleviate the problems of cold start and sparse data, adding knowledge graph (KG) as side information (by aligning the items with entities in a KG [17]) to the recommender system has proven highly effective. KG is a semantic graph composed of nodes and edges that contain rich semantic knowledge. When applied to recommendation systems, the accuracy, diversity and interpretability can be vastly improved [2]. By integrating multi-source heterogeneous information, the KG represents rich entity relationships, and helps the recommender present accurate items to users.

X. Zhao et al. (Eds.): WISA 2022, LNCS 13579, pp. 52–63, 2022.
https://doi.org/10.1007/978-3-031-20309-1_5

The main challenge of existing KG-based models is how to learn effective user/item representations for recommendation. GNNs have become the dominant method to learn graph representations [5]. KGCN [10] is a representative work. To enhance the representation of users and items, graph convolutional neural network (GCN) is utilized to aggregate information from the neighbors in the KG. Another representative work is KGAT [11], which proposes collaborative knowledge graph (CKG) to combine user-item bipartite graph (UIG) with KG. In these KG-based works, feature transformation and nonlinear activation are two common designs in GNN. However, as suggested by a recent work of lightGCN [4], the two designs play a negative role in graph collaborative filtering (GCF). It is natural to believe that the two designs are also redundant for knowledge graph collaborative filtering (KGCF). For KGCF, we mainly care about the propagation of user personalized preference. We contend that the two designs may hinder the personalized propagation of user preference in KG.

In this light, we put forward *simplified knowledge-aware attention network* (SKAN) to simplify the aggregation of existing KG-based methods, which is to boost personalized preference propagation in KG. In specific, to propagate user preference, we follow the design of KGCN to incorporate user representations into the attention mechanism. Different from KGCN, however, inspired by lightGCN, we remove the feature transformation and nonlinear activation of GNN for knowledge graph collaborative filtering (KGCF). Furthermore, to better propagate user preference, we propose to aggregate user neighbors to form user representations. This is fundamentally different from existing works, since they generally perform aggregation over items [8,9].

Contributions. To summarize, We make the following contributions:

- We propose simplified knowledge-aware attention network, which is an end-to-end framework that learns the effective user/item representations from their interactions and KG by modeling users and items separately. SKAN uses a graph neural network aggregation approach that is more suitable for collaborative filtering.
- SKAN focuses on users' preference for relationships, while we verify the negativity of two common designs of GCNs in KG-based recommender systems—feature transformation and nonlinear activation on recommendation results.
- We perform empirical experiments on three real-world recommendation scenarios whose results demonstrate the superiority of SKAN over compelling state-of-the-art baselines, and we validate the effectiveness of modeling users and new information aggregation approaches in ablation experiments.

Organization. The rest of the paper is structured as follows: Section 2 discusses related works, and Sect. 3 formalizes the task. We present our method in Sect. 4, followed by experiments in Sect. 5. Section 6 concludes the paper.

2 Related Work

Recent research of KG-based recommender systems can be divided into three categories [3]: 1) path-based methods, 2) embedding-based methods and 3) unified-based methods. Unified-based methods combine the idea of path-based methods and embedding-based methods to realize recommendation, and unified-based methods avoid the disadvantage of the first two methods: the path needs to be set manually and the high-order semantic information of the graph cannot be obtained. Representative model of unified-based recommendation include RippleNet, KGCN, KGAT, CKAN [12] and LKGR [1], etc.

RippleNet is a memory-network model that propagates usersâĂŹ potential preference in the KG and explores their hierarchical interests. But note that the importance of relations is weakly characterized in RippleNet. In addition, the size of ripple set may go unpredictably with the increase of the size of KG, which incurs heavy computation and storage overhead.

KGCN captures local neighborhood information well and considers neighbor node weights for recommendations. It also pays attention to the importance of relationship. However, it isolates user from item's attribute graph, and ignores user modeling. In addition, with the further research on the application of graph-based neural network in recommendation, it shows that the way of node aggregation in KGCN does not improve the recommendation effect, and results in a waste of computation and storage costs.

Afterwards, KGAT introduces a user-item interaction matrix in the KG and weights the graph relationships to recommend them, while CKAN holds that users are represented by interacting items, and items are represented by those items that are interacted by the users. Distinctively, LKGR employs information propagation strategies in the hyperbolic space to encode heterogeneous information from historical interaction and KG, but fails to take into account the preference of each user for different relationships.

3 Task Formulation

In recommendation, users are defined as $\mathcal{U} = \{u_1, u_2 \dots u_m\}$, and items as $\mathcal{I} = \{i_1, i_2 \dots i_n\}$. We represent interaction data as a user-item bipartite graph $\mathcal{G}_1$, which is defined as $\{(u, y_u, i) \mid u \in \mathcal{U}, i \in \mathcal{I})\}$, When the user interacts with the item, $y_{ui} = 1$, otherwise $y_{ui} = 0$. Besides, we have side information to items, organized in the form of a knowledge graph $\mathcal{G}_2$, which is a directed graph composed of subject-property-object triple facts. Formally, it is $\{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where each triple describes that there is a relationship r from head entity h to tail entity t. For example, (*Jacky Chen, ActorOf, Police Story*) states the fact that *Jacky Chen* is an actor of the movie *Police Story*.

We adopt the notion of *CKG* [11], which encodes user behaviors and item knowledge as a unified relational graph. In CKG, user behavior is expressed by a triple (u, Interact, i), where $y_{ui} = 1$ is represented as an additional relation *Interact* between user u and item i. Then, based on the *item-entity* alignment set, the user-item graph can be seamlessly integrated with the KG, producing a unified graph $\mathcal{G} = \{(h, r, t) \mid h, t \in \mathcal{E}', r \in \mathcal{R}'\}$, where $\mathcal{E}' = \mathcal{E} \cup \mathcal{U}$ and $\mathcal{R}' = \mathcal{R} \cup \{\text{Interact}\}$.

In summary, the task input is a collaborative knowledge graph $\mathcal{G}$ that combines the user-item bipartite graph $\mathcal{G}_1$ and knowledge graph $\mathcal{G}_2$, and the output is a function that predicts the probability $\hat{y}_{ui}$ that user u would adopt item i.

4 Methodology

We now present the simplified knowledge-aware attention network (SKAN). The core of this paper lies in the design of the knowledge-aware propagation. To facilitate the personalized propagation, nodes in SKAN obtains the neighbor information via linear aggregation. Without the feature transformation and nonlinear activation, during propagation the model can well concentrate on user preferences, rather than the KG semantics. Figure 1 depicts the overall framework.

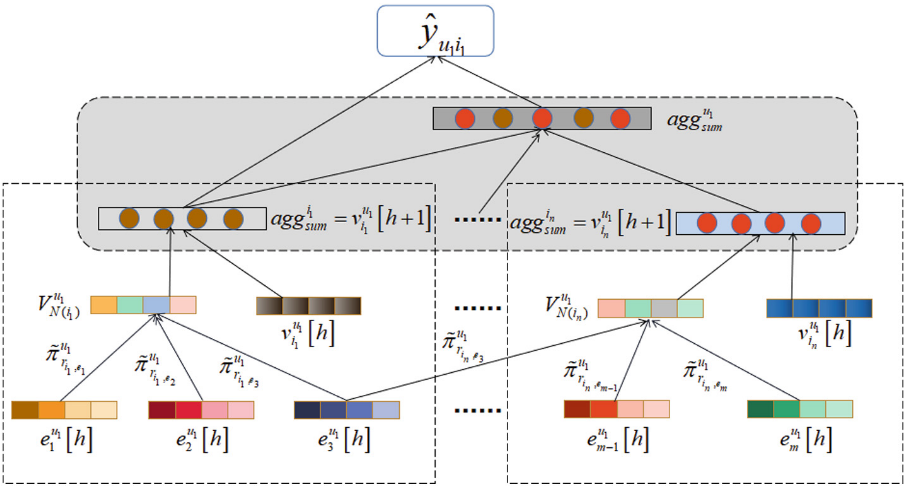

Fig. 1. The overall framework of SKAN.

4.1 Personalized Knowledge Aggregation

We propagate user preferences over the unified graph $\mathcal{G}$ through node aggregation. The high-order propagation is typically achieved through stacking multiple GNN layers. We then introduce the single GNN layer first. For every node except the users in $\mathcal{G}$, SKAN performs personalized aggregation of the information from node neighbors. Since the distribution of node degree is generally long-tail, for different entities in the graph, the number of neighbors varies largely. To address this degree imbalance, we follow [15] to sample fixed length of neighbors for every nodes. The sampled set of nodes are denoted by $\mathcal{N}$.

SKAN perform weighted aggregation with user-specific attentions:

$$\boldsymbol{e}_{\mathcal{S}(e)} = \sum_{e' \in \mathcal{N}(e)} \pi^u_{r_{e,e'}} \boldsymbol{e'}, \tag{1}$$

where $\pi^u_{r_{e,e'}}$ is the personalized attention score. We use function $f(\boldsymbol{u}, \boldsymbol{r})$ to calculate the attention, e.g.,

$$\pi^u_{r_{e,e'}} = \frac{\exp\left(f(\boldsymbol{u}, \boldsymbol{r}_{e,e'})\right)}{\sum_{e' \in \mathcal{N}(e)} \exp(f(\boldsymbol{u}, \boldsymbol{r}_{e,e'}))}, \tag{2}$$

We further aggregate the neighbor representation with the entity representation, where we remove the feature transformation and the nonlinear activation.

$$agg^i_{\text{sum}} = \boldsymbol{e} + \boldsymbol{e}_{\mathcal{S}(e)}. \tag{3}$$

Note that Eq. (3) is slightly different from that in [4]. In LightGCN [4], the entity representation itself is not aggregated. We argue that this is necessary for KGCF. The reason is that LightGCN is working on the user-item bipartite graph, where the neighbor information of the item are users only. Items can be well defined by the aggregations of interacted users in the bipartite graph. In comparison, the KG has other entities to distract the node information. Therefore, the entity representation is necessary to balance the possible distraction, especially when the entity is an item.

4.2 User Aggregation

Many KG-based recommender systems pay too much attention to the modeling of the item-side and ignore the modeling of the user-side. Similar to the item-side, the user representation in SKAN is also composed of two parts of information, one is the user's own information, and the other is the historical item information that the user has interacted with. Different from the item side, when aggregating user neighbor nodes, only one layer of aggregation is done. Since users have different preference for interactive goods, we introduce the user's attention mechanism for items as follows:

$$\lambda_{u,e} = g(\boldsymbol{u}, \boldsymbol{e}), \tag{4}$$

Users' preference for items is normalized to

$$\tilde{\lambda}_{u,e} = \frac{\exp\left(\lambda_{u,e}\right)}{\sum_{e \in \mathcal{N}(u)} \exp\left(\lambda_{u,e}\right)}, \tag{5}$$

where u is the representation of user, e is the representation of item that the user has interacted with, $\mathcal{N}(u)$ denotes the set of items connected to u. After sampling and aggregating, the items interacted by users are represented as

$$U_{\mathcal{S}(u)} = \sum_{e \in \mathcal{N}(u)} \tilde{\lambda}_{u,e} \boldsymbol{e}, \tag{6}$$

The final representation and calculation method of user is as follows:

$$agg^{u}_{\text{sum}} = \boldsymbol{u} + U_{\mathcal{S}(u)}. \tag{7}$$

4.3 Prediction Layer

From the above, we can see that in single-layer SKAN, both entity representation and user representation are jointly represented by itself and its 1-hop neighbors. When the single-layer SKAN is extended to multi-layer, the model can deeply mine the high-order information of knowledge spectrum, so as to obtain more accurate entity representation and user representation. Specifically, the initial representation of each entity (order 0 representation) is propagated to its adjacent entities to obtain the first-order entity representation. Then we can repeat this process, that is, further propagate and aggregate the 1-hop representation to obtain the 2-hops representation, so recursively. Therefore, after obtaining the vector representation of the item and user, we input them into the function F to predict the probability together:

$$\hat{y}_{uv} = F\left(\mathbf{u}, \mathbf{v}\right), \tag{8}$$

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

$$\mathcal{L} = \sum_{u \in \mathcal{U}} \left(\sum_{v: y_{uv}=1} \mathcal{J}\left(y_{uv}, \hat{y}_{uv}\right) - \sum_{i=1}^{T^u} \mathbb{E}_{v_i \sim P(v_i)} \mathcal{J}\left(y_{uv_i}, \hat{y}_{uv_i}\right) \right) + \lambda \|\mathcal{F}\|_2^2. \tag{9}$$

where $\mathcal{J}$, P and T^u are the representations of cross-entropy loss, a negative sampling distribution and the number of negative samples for user u, respectively. The last term is the L2-regularizer.

5 Experiments

In this section, we evaluate the proposed model under three real world scenarios, with the aim of answering the following research questions:

– **RQ1**: How does SKAN perform compared with state-of-the-art methods?
– **RQ2**: How do different components affect SKAN?

5.1 Datasets

In order to verify the effectiveness of our proposed method, we conducted comparative experiments on three public datasets, i.e., Movie-Lens20M, Book-Crossing and Dianping-Food. The statistics of the datasets are presented in Table 1.

Table 1. Statistics of the three datasets: Movie-Lens20M, Book-Crossing, and Dianping-Food. The inter-avg means the average interactions per user, the link-avg means the average links per item.

Dataset	Movie-Lens20M	Book-Crossing	Dianping-Food
# users	138,159	17,860	2,298,698
# items	16,954	14,967	1,362
# interactions	13,501,622	139,746	23,416,418
# inter-avg	98	8	10
# entities	102,569	77,903	28,115
# relations	32	25	7
# KG triples	499,474	151,500	160,519
# link-avg	29	10	118

- **Movie-Lens20M**[1] is a widely used benchmark dataset in movie recommendations, which consists of approximately 20 million explicit ratings (ranging from 1 to 5) on the MovieLens website.
- **Book-Crossing**[2] is collected from the book-crossing community, which consists of trenchant ratings (ranging from 0 to 10) from different readers about various books.
- **Dianping-Food**[3] is provided by Dianping.com, which contains over 10 million interactions (including clicking, buying, and adding to favorites) between approximately 2 million users and 1 thousand restaurants. The corresponding KG contains 28, 115 entities, 160, 519 edges and 7 relation-types.

5.2 Baselines

In order to demonstrate the effectiveness of SKAN, we compare it with the state-of-the-art methods, including CF-based method (BPRMF), embedding-based methods, path-based method (PER) unified-based methods (RippleNet, KGCN, KGNN-LS, KGAT, CKAN, LKGR):

- **BPRMF** [7] is a Bayesian personalized ranking model based on matrix factorization (MF), which realizes personalized recommendation by learning individual user preference.
- **CKE** [14] is a representative embedding-based method, which leverages KG embeddings of entities derived from TransR as ID embeddings of items under the MF framework. Where, KG relations are only used as the constraints in TransR to regularize the representations of endpoints.

[1] https://grouplens.org/datasets/movielens/.
[2] https://searchengineland.com/library/bing/bing-satori.
[3] https://github.com/hwwang55/KGNN-LS/raw/master/data/restaurant/Dianping-Food.zip.

– **PER** [13] treats the KG as HIN and extracts meta-path based features to represent the connectivity between users and items. In this paper, we use all item-attribute-item features for PER (e.g., âĂIJmovie-director-movie”).
– **RippleNet** [8] is an end-to-end framework that uses KG to realize recommendation system. It combines the embedding-based and path-based method into the recommendation system based on KG for the first time. Through the method of preference propagation in KG, users' potential hierarchical interests are continuously and automatically found.
– **KGCN** [10] is a state-of-the-art unified-based method which extends spatial GCN approaches to the KG domain. By aggregating high-order neighbor information, both structure information and semantic information of the KG can be learned to capture usersâĂŹ potential long-distance interests.
– **KGNN-LS** [16] is based on KGCN, which transforms KG into user-specific graphs, and then considers user preference on KG relations and label smoothness in the information aggregation phase, so as to generate personalized item representations.
– **KGAT** [11] is a propagation-based recommender model. It applies a unified relation-aware attentive aggregation mechanism in UKG to generate user and item representations.
– **CKAN** [12] is based on KGNN-LS, which utilizes different aggregation schemes on the user-item graph and KG respectively, to encode knowledge association and collaborative signals.
– **LKGR** [1] is a state-of-the-art hyperbolic GNN method with Lorentz model, which employs different information propagation strategies in the hyperbolic space to encode heterogeneous information from interactions and KG.

5.3 Experimental Settings

We implemented SKAN in `TensorFlow`, and we randomly divided the dataset into training set, evaluation set and test set, and their division ratio is $6:2:2$. In our experiments, click-through rate (CTR) prediction was the recommendation scenario, and we used AUC and F1 to evaluate the effectiveness.

The hyper-parameter settings for SKAN are as follows, where neighbor sampling size, dimension of embeddings, depth of receptive field, L2 regularizer weight, learning rate, training times are denoted as K, d, H, λ, η and n respectively: for Movie-Lens20M, $K = 8$, $d = 32$, $H = 2$, $\lambda = 1e-7$, $\eta = 2e-3$, $n = 20$; for Book-Crossing, $K = 8$, $d = 64$, $H = 3$, $\lambda = 2e-5$, $\eta = 2e-4$, $n = 10$; for Dianping-Food, $K = 4$, $d = 8$, $H = 2$, $\lambda = 1e-7$, $\eta = 2e-2$, $n = 10$. The best settings for hyper-parameters in all comparison methods are reached by either empirical study or following their original papers.

5.4 Performance Comparison (RQ1)

We first report the performance of all the methods, and then analyze the experimental results. The experimental results are reported in Table 2, where %Imp. denotes the relative improvement of the best performing method (bolded) over the strongest baselines (underlined). We find that:

Table 2. The results of AUC and F1 in CTR prediction.

Dataset	Movie-Lens20M		Book-Crossing		Dianping-Food	
Metric	AUC	F1	AUC	F1	AUC	F1
BPRMF	0.958	0.914	0.658	0.611	0.832	0.764
CKE	0.7321	0.7385	0.7323	0.6363	0.6462	0.6559
PER	0.838	0.792	0.605	0.572	0.766	0.697
RippleNet	0.976	0.927	0.721	0.647	0.863	0.783
KGCN	0.977	0.93	0.684	0.631	0.845	0.774
KGNN-LS	0.975	0.929	0.676	0.631	0.852	0.778
KGAT	0.976	0.928	0.731	0.654	0.846	0.785
CKAN	0.976	0.929	0.753	0.673	0.878	0.802
LKGR	0.979	0.9336	0.7056	0.6536	0.871	0.7932
SKAN(ours)	**0.9839**	**0.9411**	**0.7735**	**0.6998**	**0.9206**	**0.8444**
%Imp.	0.50%	0.80%	2.72%	3.98%	4.85%	5.29%

(1) SKAN consistently yields the best performance on all the datasets. In particular, it achieves significant improvement even over the strongest baselines w.r.t. AUC by 0.5%, 2.72% and 4.85% in Movie-Lens20M, Book-Crossing and Dianping-Food datasets, respectively. We believe that the following reasons have led to this good result: 1)modeling the user makes the embedded representation of the user more accurate; 2)Removing the nonlinear activation function and transformation matrix is indeed helpful to improve the recommendation effect. We will design ablation experiments later to verify the above views.

(2) From the results of three datasets, the ranking of the results of all modelsâĂŹ performance on the three datasets from high to low is Movie-Lens20M, Dianping-Food, Book-Crossing. By looking at the data in Table 1, we believe that the reason for this may be that the average interaction per user and the average links per item are different. On average, Movie-Lens20M and Dianping-Food datasets have richer interactions and links than Book-Crossing datasets. Therefore, for datasets with poor interaction and links, the recommended model does not have enough information to understand the potential embedding.

(3) From the experimental results, it is also easy to see that KG is very obvious to improve the recommendation effect. KG-based methods is more effective than CF-based methods, especially the unified-based methods on the three datasets across two evaluation metrics.

(4) Through the comparison of results of all KG-based methods, The unified-based methods are better than embedding-based methods and the path-based methods on three datasets. This shows the importance of high-order information in KG-based recommender systems. Among all methods, the path-based method performs the worst, which may be because it is difficult to define

the optimal path in reality. The poor performance of the embedding-based method may be due to its static nature, which ignores the information connectivity in the knowledge graph, can not use the multi hop relationship between entities, can not mine the high-order relationship in the graph.

5.5 Ablation Studies (RQ2)

In this section, we examine the contributions of main components in our model to the final performance by comparing SKAN with the following two variants:

(1) SKAN_{fn}: Inspired by lightGCN, we remove the nonlinear activation function and transformation matrix from the aggregation function. The model without removing the nonlinear activation function and transformation matrix is recorded as SKAN_{fn}.
(2) SKAN_{user}: User modeling is an important module in SKAN. The model without user modeling is called SKAN_{user}.

Table 3. Ablation study on feature transformation and nonlinear activation.

Dataset	Movie-Lens20M		Book-Crossing		Dianping-Food	
Metric	AUC	F1	AUC	F1	AUC	F1
SKAN_{fn}	0.9836	**0.9413**	0.7605	0.6923	0.9173	0.8392
SKAN	**0.9839**	0.9411	**0.7735**	**0.6998**	**0.9206**	**0.8444**
%Imp.	0.03%	–0.028%	1.71%	1.08%	0.36%	0.62%

Table 4. Ablation study on user modeling.

Dataset	Movie-Lens20M		Book-Crossing		Dianping-Food	
Metric	AUC	F1	AUC	F1	AUC	F1
SKAN_{user}	0.974	0.9287	0.7049	0.6603	0.8409	0.7703
SKAN	**0.9839**	**0.9411**	**0.7735**	**0.6998**	**0.9206**	**0.8444**
%Imp.	1.02%	1.34%	9.73%	5.98%	9.48%	9.62%

Are Nonlinear Activation Functions and Transformation Matrices Useful? The KG-based recommender systems generally has nonlinear activation function and feature transformation matrix in the aggregation function part, but lightGCN points out that they not only do not improve the recommendation effect, but increase the difficulty of training. Because The premise that the nonlinear activation function and transformation matrix can work is that the nodes have rich feature representation. It can be seen from the results in Table 3 that this conclusion is still applicable to the KG-based recommender systems. Notice that, SKAN does not perform best on Movie-Lens20M in Table 3, since the average interactions per user are so abundant that the nodes have rich feature representation (Table 4).

Is User Modeling Working? Most KG-based recommender systems ignore the modeling of users. SKAN aggregates historical items that users have interacted with to obtain better user representations. From the experimental results, user modeling can improve the recommendation effect.

6 Conclusion

In this work, we propose a recommendation system model to make better use of item attribute information. First, we build a collaborative knowledge graph, then we model entities and users on CKG to get their embedded representations by simplified knowledge-aware attention network which simplifies the knowledge-aware aggregation by removing feature transformation and nonlinear activation. Specifically, we use neighbor sampling and aggregation with attention mechanism for entities and users, in which the attention mechanism takes into account the user's preference for relationships and items. Experiment results show that SKAN outperforms state-of-the-art baselines on three real-world recommendation scenarios. In future, it is an exciting research direction that using KG information to explore the interpretability of recommender systems.

Acknowledgement. This work was in part supported by NSFC under grant No. 61872446, and The Science and Technology Innovation Program of Hunan Province under grant No. 2020RC4046.

References

1. Chen, Y., et al.: Modeling scale-free graphs with hyperbolic geometry for knowledge-aware recommendation. In: Proceedings of the 15th ACM International Conference on Web Search and Data Mining, pp. 94–102 (2022)
2. Cheng, H.T., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10 (2016)
3. Guo, Q., et al.: A survey on knowledge graph-based recommender systems. IEEE Trans. Knowl. Data Eng. **34**, 3549–3568 (2020)
4. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 639–648 (2020)
5. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

6. Wang, Y., Gao, S., Li, W., Jiang, T., Yu, S.: Research and application of personalized recommendation based on knowledge graph. In: Xing, C., Fu, X., Zhang, Y., Zhang, G., Borjigin, C. (eds.) WISA 2021. LNCS, vol. 12999, pp. 383–390. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87571-8_33
7. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
8. 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, pp. 417–426 (2018)
9. Wang, H., Zhang, F., Xie, X., Guo, M.: DKN: deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide Web Conference, pp. 1835–1844 (2018)
10. Wang, H., Zhao, M., Xie, X., Li, W., Guo, M.: Knowledge graph convolutional networks for recommender systems. In: The World Wide Web Conference, pp. 3307–3313 (2019)
11. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.S.: KGAT: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 950–958 (2019)
12. 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, pp. 219–228 (2020)
13. Yu, X., et al.: Personalized entity recommendation: a heterogeneous information network approach. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 283–292 (2014)
14. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–362 (2016)
15. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Adv. Neural Inform. Process. Syst. **30** 1024–1034 (2017)
16. 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, pp. 968–977 (2019)
17. Zhao, X., Zeng, W., Tang, J., Wang, W., Suchanek, F.: An experimental study of state-of-the-art entity alignment approaches. IEEE Trans. Knowl. Data Eng. **34**, 2610–2625 (2020)
18. Chen, Y., Wang, Y., Zhao, X., Zou, J., Rijke, M.D.: Block-aware item similarity models for top-n recommendation. ACM Trans. Inform. Syst. **38**, 1–26 (2020)