Graph Neural Network Combined Knowledge Graph for Recommendation System

Dong Nguyen Tien and Hai Pham Van(✉)

Hanoi University of Science and Technology, Hanoi, Vietnam
dongnguyentien1996@gmail.com, haipv@soict.hust.edu.vn

Abstract. With a view to increase recommendation systems accuracy and practical applicability, using traditional methods which are namely interaction model between users and items, collaborative filtering and matrix factorization cannot achieve the supposed results. In fact, the properties between users or items always remains as social and knowledge relations. In this paper, we have proposed a new graph deep learning model associated with knowledge graph with the aim of modeling the latent feature of user and item. We exploit the relations of items based on knowledge graph as well as the relationships between users in social. Our model supplies the principle of organizing interactions as a graph, combines information from social network and all kind of relations in the heterogeneous knowledge graph. The model is evaluated on real world datasets to demonstrate this method's effectiveness.

Keywords: Recommendation system · Graph neural network · Knowledge graph · Social recommendation

1 Introduction

The traditional method of the RS is collaborative filtering [1], based on the behavior of users and modeling their interactions by analyzing matrix factorization [2] or neural networks. In the recent years, neural network technology for graph data have made lots of remarkable developments [3] which are called Graph Neural Networks (GNN) [4]. In terms of creating features, recent studies like [6, 7] not only use individual features but also link them together to form knowledge graph (KG). KG is a directional heterogeneous graph where the nodes correspond to the items and the edges correspond to the relationships. Combining KG benefits the results in three ways: (1) The rich semantic relatedness among items in a KG can help explore their latent connections and improve the precision of results; (2) Different types of relation in a KG are useful to logically extend user interests and increase the variety of proposed items; (3) KG connects a user's historically-liked and recommended items. In Fig. 1, the graph includes social relations, interaction graph between users and items, and knowledge graph of items. As a result, with its advantages, incorporating GNNs with the KG provides an unprecedented opportunity to enhance the results.

S. Chellappan et al. (Eds.): CSoNet 2020, LNCS 12575, pp. 59–70, 2020.
https://doi.org/10.1007/978-3-030-66046-8_6

This paper has presented a new model to assemble multiple aspects of data which is shown in Fig. 1, simultaneously addressing the mentioned challenges. Contributions include: Propose a model combining a KG called **KconvGraphRec**; provide an approach to building model from three base graphs and combine them to make predictions, experiment and demonstrate the effectiveness on real world datasets.

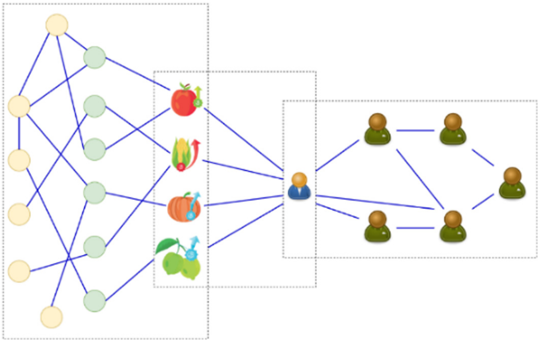

Fig. 1. Graph data in recommendation system. Includes interaction graph between user - item (middle), user social relationship graph (right corner) and items knowledge graph (left corner)

2 Related Work

Recently, the application of graph neural networks in recommendation systems, using data from social relations and knowledge graph to make representation vector for users and items. The exploitation of data related to the user's social relationship is great interest recently [8, 9]. One common assumption about these models is that a user's preference is like or influenced by the people around him/her (nearest neighbours), which can be proven by social correlation theories [10]. TrustMF [11] modeled mutual influence between users, and mapped into two low-dimensional spaces: truster space and trustee space, by factorizing social trust networks. SoDimRec [12] applies community detection algorithms to divide users into clusters, then exploits the social relationships and weakly dependent connections. A comprehensive overview of the social relations recommendation system can be found in the survey [13].

Although neural network had flourished but the use in social recommendation systems was unusual by far. NeuMF [1] presented a neural collaborative filtering model to learn non-linear interactions between users and items. For graph data, there have been currently studies of applying neural networks to graph data and have proven their effectiveness compared to conventional data types [14]. GraphRec [15] - state of the art model for applying social relations data of users and the interaction between users and items.

Knowledge graph is being abundantly studied lately [8, 16]. However, the fact that KG is directly applied to the RS is still insignificant. KGAT [8] proposes the attention mechanism for the KG, the end-to-end framework in order to model high-order structural connections in graph neural networks. KGCN [17] uses the idea of convolution graph network to predict binary subclass. This method works directly on the original graph and determines convolutional for group of nodes. In order to process different sized neighborhoods and maintain the sharing information, the researchers proposed sample a fixed-sized neighborhood group in KGCN. DKN [18] focused on solving by embedding the KG, capturing the ratings from the users to the item with their implicit relationships. DKEN explores the relationship between users and entities that interact with items in the KG. Lately, HAGERec [19], which is further improved from KGCN, can aggregate more hops and change the model to evaluate the impact of users in KG.

3 The Proposed Model

3.1 Definitions and Notations

Let $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_m\}$ are the set of users and items respectively, where n is number of users and m is number of items. We assume $R \in R^{n\times m}$ is interaction matrix users-items. If u_i rate item v_j, r_{ij} is a rating score, otherwise $r_{ij} = 0$. Let $N(i)$ is set of users have relations with user u_i in social graph, C(i) is set of items which rated by user u_i and B(j) is set of users who have interacted item v_j. Social graph $T \in R^{n\times n}$ with $T_{ij} = 1$ if u_i has a relation to user u_j and 0 otherwise. Knowledge graph $\mathfrak{G}$, form with triple entity – relation – entity (h_g, r_g, t_g) where $h_g \in \mathcal{E}, t_g \in \mathcal{E}, r_g \in \mathfrak{R}$ are head, tail and relation respectively. $\varepsilon, \mathfrak{R}$ are the number of entities và relations in this knowledge graph. Then users-items **R**, social relations graph **T** và knowledge graph $\mathfrak{G}$, we aim to predict the missing rating value r of set user-item in **R.** We use user embedding vector u_i is $p_i \in R^d$, item embedding vector v_j is $q_j \in R^d$ where d is number dimension of embedding vector.

3.2 The Proposed Model

In this paper, the data input included: rating data users to items, social data and knowledge graph for items (entity). These datasets will be modeled concurrently and processed with the output set as the result of predictive rating of user – item.

The proposed model includes 3 parts as follows: user modeling, item modeling and rating modeling. Firstly, user modeling which to learn user latent vector. Because the data includes two different of graph: social graph and interaction graph, two aggregations are introduced to respectively process these two different graphs. The second component is item modeling, which learn item latent vector.

3.3 User Modeling

User modeling aims to learn user latent vector, denoted as $h_i \in R^d$ for user u_i. First is aggregator from item space $h_i^I \in R^d$ of interaction graph. The second aggregate from social space $h_i^S \in R^d$. After that, they are combined to the final user latent vector h_i

Item Aggregation
Interactions between user and item contain rating score from 1 to 5. The purpose of item aggregation is to learn item-space latent vector, which has a function as (Fig. 2):

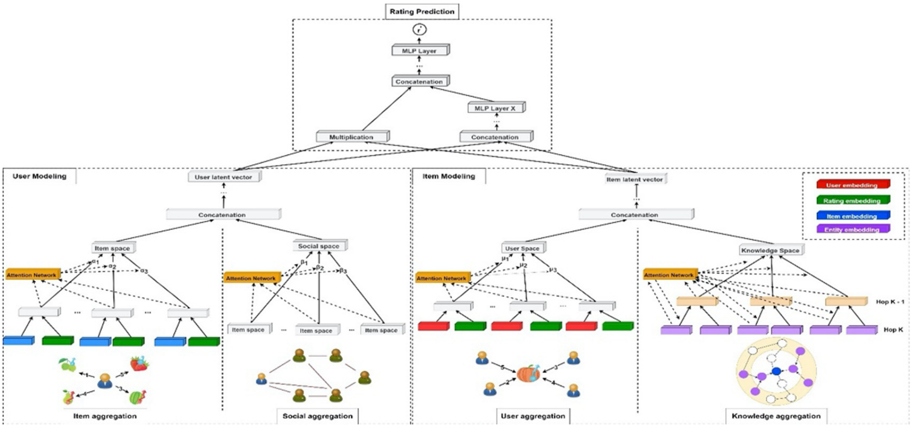

Fig. 2. Detailed architecture of the proposed model KconvGraphRec

$$h_i^I = \sigma(W.Aggre_{items}(\{x_{ia}, \forall a \in C(i)\}) + b) \tag{1}$$

With C(i) set of items which are interacted by *user* u_i, x_{ia} is representation vector to denote opinion-aware interaction from *user* u_i to *item* v_a and $Aggre_{items}$ is aggregation function. W, b, σ are hyper parameter of neural network. To modeling opinions, we have opinion embedding vector $e_r \in R^d$ of rating $r \in \{1, 2, 3, 4, 5\}$ respectively. For each rating, x_{ia} combined vector item embedding q_a vector opinion embedding e_r via Multi-Layer Perceptron (MLP), denoted as g_v:

$$x_{ia} = g_v([q_a \oplus e_r]) \tag{2}$$

where $\oplus$ is concatenation operation. Next, we will introduce of $Aggre_{items}$:

$$h_i^I = \sigma\left(W.\left\{\sum\nolimits_{\alpha \in C(i)} \alpha_{ia} x_{ia}\right\} + b\right) \tag{3}$$

Where α_{ia} is coefficient attention, output of 2 layers neural network and obtained by normalizing using softmax function:

$$\alpha_{ia}^* = W_2^T.\sigma(W_1.[x_{ia} \oplus p_i] + b_1) + b_2 \tag{4}$$

$$\alpha_{ia} = \frac{\exp(\alpha_{ia}^*)}{\sum_{\alpha \in C(i)} \exp(\alpha_{ia}^*)} \tag{5}$$

Social Aggregation

The social correlation theories [14] have demonstrated the impact of the relationship between people on the interests of each individual. The relationship depends on how much of general interaction according users. In other words, constructing the user vector from social space needs to consider heterogeneous relationships in society. Then, the social space user latent vector as the follow:

$$h_i^S = \sigma\left(W.Aggre_{neighbors}\left(\left\{h_i^o, o \in N(i)\right\}\right) + b\right) \tag{6}$$

$$h_i^S = \sigma\left(W.\left\{\sum\nolimits_{o \in N(i)} \beta_i h_i^o\right\} + b\right) \tag{7}$$

With β_i is the attention score, built via 2 layers neural network from item-space vector with user embedding vector p_i.

$$h_i^S = \sigma\left(W.\left\{\sum\nolimits_{o \in N(i)} \beta_{io} h_i^o\right\} + b\right) \tag{8}$$

$$\beta_{io}^* = W_2^T.\sigma\left(W_1.\left[h_i^o \oplus p_i\right] + b_1\right) + b_2 \tag{9}$$

$$\beta_{io} = \frac{\exp\left(\beta_{io}^*\right)}{\sum_{o \in N(i)} \exp\left(\beta_{io}^*\right)} \tag{10}$$

Learning User Latent Vector

In order to learn better user latent vector, social space and item space need to be considered together, since the social graph and interaction graph supply 2 aspect of user. We combine these two latent factors via standard MLP. Formally, with l is the number of hidden layers, the user latent vector is defined as:

$$c_1 = \left[h_i^I \oplus h_i^S\right] \tag{11}$$

$$c_2 = \sigma(W_2.c_1 + b_2) \tag{12}$$

$$h_i = \sigma(W_l.c_{l-1} + b_l) \tag{13}$$

3.4 Item Modeling

Item latent vector, denoted as z_j, is the combination of two components: user aggregation and knowledge aggregation. We are not only construct from the interactions of all users for item v_j, but also utilizing the information from KG of items.

User Aggregation

Similar with User modeling, each *item* v_j, we synthesis all interaction of *users* who rated with *item* v_j, denoted as B(j). Even on the same item, users might express different opinions. These opinions from different users can capture the characteristics of the same item in different ways, which help to learn better item latent. For an interaction user u_t to *item* v_j with rating r, the function f_{jt} which is obtained from the user embedding p_t and opinion embedding e_r via a MLP, denoted as g by following:

$$f_{jt} = g_u\left(\left[p_t \oplus e_r\right]\right) \tag{14}$$

Then, attention mechanism to differentiate the importance weight μ_{jt}, represent the influence of different user for different item, it depends on rating score:

$$z_j^U = \sigma\left(W.\left\{\sum\nolimits_{t \in B(j)} \mu_{jt}.f_{jt}\right\} + b\right) \tag{15}$$

$$\mu_{jt}^{*} = W_2^T.\sigma\left(W_1.\left[f_{jt} \oplus q_j\right] + b_1\right) + b_2 \tag{16}$$

$$\mu_{jt} = \frac{\exp\left(\mu_{jt}^{*}\right)}{\sum_{t \in B(j)} \exp\left(\mu_{jt}^{*}\right)} \tag{17}$$

Knowledge Aggregation

In KG, item (entity) has many relations in triple (head, relation, tail). The key idea is to aggregate and incorporate neighborhood information when calculating the representation of a given entity. This design has advantages: (1) Through the neighborhood, the local proximity structure is captured and stored in each entity. (2) Neighbors are weighted dependent on the relation and specific user, which characterizes both the semantic information and users 'personalized interests. (3) Attention mechanism leveraging weight have well-established node classification. To resolve the size of an entity's neighbors varies and maybe large, we sample a fixed-size neighborhood.

For each pair user u_i and item (entity) v_j. Having $N_g(v)$ is set of entities which have relationship with item v_j and so we denote r_{e_i,e_j} is the relation score of entity e_i and e_j. We have a function to calculate score between user and relation in KG:

$$\pi_r^u = g(u \oplus v) \tag{18}$$

Where $u \in R^d$ và $r \in R^d$are the representations of user and item v and d is the dimension vector. Weight π_r^u represents the importance of relation r to user u. To alleviate the limitation of mean-based aggregator, we utilize MLP to build attention weigh to express the specificity of each user for specific relation in knowledge graph:

$$\pi_r^u = W_2^T.\sigma(W_1.[u \oplus v] + b_1) + b_2 \tag{19}$$

To characterize the topological proximity structure of item v, we compute the linear combination of v's neighborhood.

$$v_{N_g(v)}^u = \sum\nolimits_{e \in N_g(n)} \tilde{\pi}_{r_{v,e}}^u \boldsymbol{e} \tag{20}$$

$$\tilde{\pi}_{r_{v,e}}^u = \frac{\exp\left(\pi_{r_{v,e}}^u\right)}{\sum_{e \in N_g(n)} \exp\left(\pi_{r_{v,e}}^u\right)} \tag{21}$$

Where $\tilde{\pi}_{r_{v,e}}^u$ is the normalized user-relation score, $\boldsymbol{e}$ is entity embedding vector and $\tilde{\pi}_{r_{v,e}}^u$ is the attention score. We uniformly sample a fixed size set of neighbors. The neighborhood area of entity v and $v_{S_g(v)}^u$, where $S(v) = \left\{e | e \in N_g(v) \;\&\; \left|S_g(v)\right| = K\right\}$, K is constant. Finally, we aggregate the entity representation v and its neighborhood $v_{S_g(v)}^u$ into single vector: $R^d \times R^d \to R^d$ (Fig. 3).

$$z_j^K = \sigma\left(W.\left(v + v_{S_g(v)}^u\right) + b\right) \tag{22}$$

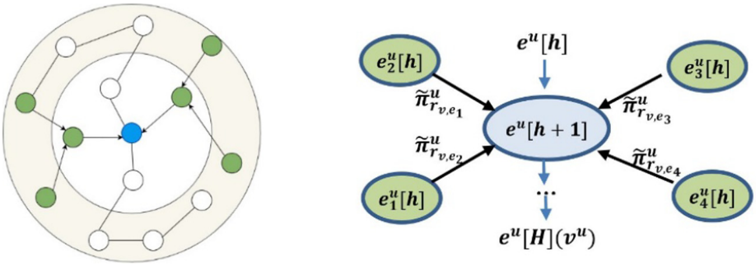

Fig. 3. The neighborhood (green entities) in 2 hops of item (blue entity) in KG (left corner). The formation to aggregate the information set of neighboring nodes about item (right corner) (Color figure online)

Learning Item Latent Vector

To learning item latent vector, user-space and knowledge-space item latent vector are needed to be combined. We combine these by two latent factors via a standard MLP. With l is number of hidden layers, item latent vector is denoted as:

$$c_1 = \left[z_j^U \oplus z_j^K\right] \tag{23}$$

$$z_j = \sigma(W_l.c_{l-1} + b_l) \tag{24}$$

3.5 Rating Prediction

In this section, we combine item latent vector and user latent vector for rating prediction. In this work, we apply method which is proposed in NeuMF [1]. We utilize Generalized Matrix Factorization (GMF) and Multi-layer perceptron (MLP). Then, we combine these models together to superimpose their desirable characteristics before feeding into neural network standard to calculate final rating prediction. GMF use the operator element wise of user latent vector and item latent vector while MLP concatenate latent vector of user and item as input and feed into neural network.

Finally, concatenation g_{gmf} and g_{mlp} to feed into NeuMF to predict final score:

$$g_1 = \left[g_{gmf} \oplus g_{mlp}\right] \tag{25}$$

$$g_{k-1} = \sigma(W_{k-1}.g_{k-1} + b_{k-1})$$

$$r_{ij}^{'} = W^T.g_{k-1} \tag{26}$$

Where k is the number of hidden layers and $r_{ij}^{'}$ is prediction score of user u_i to item v_j.

3.6 Training Model

To building model hyper parameters, we construct the loss function to optimize. Since the target is to predict rating score from user to item, so the loss function is:

$$Loss = \frac{1}{2|\mathcal{O}|} \sum\nolimits_{i,j \in \mathcal{O}} \left(r'_{ij} - r_{ij} \right) \tag{27}$$

Where, $|\mathcal{O}|$ is the number of observed ratings, r_{ij} is the ground truth of user u_i to item v_j. To optimize the objective function, we adopt the RMSprop [20]. There are 4 vectors embedding in model, included item (entity) embedding q_j, user embedding p_i, opinion embedding e_r and relation embedding r. They are randomly initialized and jointly learned during the training state. By embedding high-dimensional sparse features into low-dimensional latent space, the model can be easy to train and reduce time for training.

4 Experiment

4.1 Experimental Settings

Dataset

In our experiments, we use the datasets which are downloaded from public. There are Ciao, Epinions, MovieLens 1M[1], LastFM[2]

- Dataset Ciao, Epinions can take from popular social networking website Ciao (http://www.ciao.co.uk) and Epinions (www.epinions.com). Both has the data of rating users to items and social networking relations data.
- MovieLens 1M consists of approximately more than 1 million explicit ratings (ranging from 1 to 5) on the MovieLens website.
- Last.FM contains musician listening information from a set of 2 thousand users from Last.fm online music system.

All 4 datasets do not have enough data as expected, including: interactions data between users and items, social network relations among users and KG of items. Under that challenge, we propose to build a dataset for the following cases:

- For Ciao and Epinions, we will construct KG for items by the way that KB4Rec [22] proposed. Specifically, we consider the triple (head, relation, tail) have directly related to the entities associated with the items regardless of head or tail.
- For MovieLen 1M and Last.FM, since there are no data on social relationships, we use the social connections of Epinions dataset, and normalize user ID in Epinions to match in MovieLen 1M and Last.FM datasets (Table 1).

[1] https://grouplens.org/datasets/movielens.

[2] https://grouplens.org/datasets/hetree-2011.

Table 1. Statistics of the datasets

Dataset	Ciao 1hop	Ciao 2hop	Epi 1hop	Epi 2hop	Movie 2hop
#Users	7,375	7,375	49,289	49,289	138,159
#Items	106,797	106,797	261,649	261,649	16,954
#Rating	283,319	283,319	764,352	764,352	1,501,622
#Social connection	111,781	111,781	487,184	487,184	487,184
#Entities	128,572	190,961	205,868	315,548	102,569

Evaluation Metric

In order to evaluate the quality of the algorithms, two popular metrics are adopted namely Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [23]. Smaller values of MAE and RMSE indicate better predictive accuracy. Note that small improvement in RMSE or MAE terms can have a significant impact on the quality of the top-few recommendations [24].

Baseline

To evaluate the performance, we compared our **KconvGraphRec** with four groups of methods including traditional RS, social recommender systems and deep neural network-based RS. For each group, we select representative baselines and below we will detail them:

- NeuMF [1]: This method is a state-of-the-art matrix factorization model with neural network architecture. The original implementation is for recommendation ranking task and we adjust its loss to the squared loss for rating prediction.
- GCMC+GN [22]: This model is a state-of-the-art recommender system with graph neural network architecture
- KGCN [17]: The model proposes KG convolutional network method to aggregate the neighborhood data of the entity. Thereby building the vector representation of the entity that carries the full information of the KG.
- GraphRec [15]: is model state of the art using graph deep learning for both rating information graphs and social graphs to predict rating between users and items.
- HAGERec [19]: utilizes a bi-directional information propagation strategy to fully exploit the semantic information and high-order connectivity. It can learn the central entity's embedding from its local proximity structure.
- Without attention: Proposed model without attention in knowledge aggregation.

4.2 Experiment Results

Table 2. Performance comparison of different recommender systems

Dataset	Metric	Algorithms					
		KGCN	NeuMF	GCMC+GN	GraphRec	Without attention	**KconvGraphRec**
Ciao 1hop	MAE	–	0.8062	0.7526	0.7504	0.7330	**0.7218**
	RMSE	–	1.0617	0.9931	1.0917	1.0216	**0.9914**
Ciao 2hop	MAE	0.8124	0.8062	0.7526	0.8015	0.7215	**0.7179**
	RMSE	1.1187	1.0617	0.9931	1.0928	0.9852	**0.9725**
Epi 1hop	MAE	–	0.9072	0.8590	0.8285	0.8202	**0.8092**
	RMSE	–	1.1476	1.0711	1.1298	1.1183	**1.0146**
Epi 2 hop	MAE	0.8554	0.9072	0.8590	0.8287	0.8137	**0.8057**
	RMSE	1.1398	1.1476	1.0711	1.1357	1.0949	**1.0104**
Movie 2 hop	MAE	0.7591	–	–	0.7280	0.7271	**0.7152**
	RMSE	1.0012	–	–	0.9856	0.9783	**0.9624**

As a result, in Table 2, we have a few evaluations as follows:

- The proposed model outperforms four state of the art methods NeuMF, GCMC+GN, GraphRec and KGCN. That shows the effectiveness of the proposed model for the problem of the recommendation system.
- The model has proven the correctness when combining social network, knowledge graph and interaction between the users and items to synthesize many aspects into the corresponding representation vector to improve the result.

Table 3. Performance comparison with KGCN and HAGERec in AUC, ACC metric

Algorithms	Movie Lens 1M		Last.FM	
	AUC	ACC	AUC	ACC
KGCN	0.907	0.833	0.796	0.724
HAGERec	**0.923**	0.847	**0.814**	0.743
KconvGraphRec	0.9102	**0.848**	0.798	**0.747**

In Table 3, we compare the proposed model with KGCN, HAGERec in MovieLens 1M and Last.FM. Because 2 above models use the metric AUC and ACC for binary classification, we add sigmoid function to the output. To avoid imbalance data, we random sampling to generate data to equalize class 0 and 1. The results show that the proposed model continues to outperform KGCN. For HAGERec, the number of epochs is 200 which is much greater than 30 of proposed model (Fig. 4).

Fig. 4. Number of epochs in training model

5 Conclusion

The paper has proposed a neural network application model in graphs with a social relation data and knowledge graphs to solve the challenges in the recommendation system. Additionally, the paper proves that the theoretical basis and experimental results are much better than the recent state-of-the-art models. Experiments have demonstrated the interplay of implicit factors of users and items that contribute to boosting the predictive results of the recommender system and the model performance. We only incorporate social graphs into recommendations while many real scenarios are linked to a lot of other information. Thus, exploring graph neural networks to make proposals with those features will be considered as a suggestion in the future.

Acknowledgement. This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2019.316.

References

1. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: 26th International World Wide Web Conference, WWW 2017, pp. 173–182 (2017)
2. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of 4th ACM Conference on Recommender Systems, RecSys 2010, pp. 135–142 (2010)

3. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, pp. 1–14 (2017)
4. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, NIPS, pp. 1025–1035 (2017)
5. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, NIPS, pp. 3844–3852 (2016)
6. Zhang, Z., Cui, P., Zhu, W.: Deep learning on graphs: a survey. http://arxiv.org/abs/1812.04202 (2018)
7. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: a survey. Knowl.-Based Syst. **151**, 78–94 (2018)
8. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.S.: KGAT: knowledge graph attention network for recommendation. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 950–958 (2019)
9. Cao, Y., Hou, L., Li, J., Liu, Z.: Neural collective entity linking, no. Figure 1.http://arxiv.org/abs/1811.08603 (2018)
10. Yang, B., Lei, Y., Liu, J., Li, W.: Social collaborative filtering by trust. IEEE Trans. Pattern Anal. Mach. Intell. **39**(8), 1633–1647 (2017). https://doi.org/10.1109/TPAMI.2016.2605085
11. Wang, H., et al.: RippleNet: propagating user preferences on the knowledge graph for recommender systems. In: International Conference on Information and Knowledge Management Proceedings, no. 1, pp. 417–426 (2018). https://doi.org/10.1145/3269206.3271739
12. Tang, J., et al.: Recommendation with social dimensions. In: 30th AAAI Conference on Artificial Intelligence, AAAI 2016, pp. 251–257 (2016)
13. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A survey on knowledge graphs: representation, acquisition and applications, pp. 1–26. http://arxiv.org/abs/2002.00388 (2020)
14. Van Pham, H., Nguyen, V.T.: A novel approach using context matching algorithm and knowledge inference for user identification in social networks. In: Proceedings of the 4th International Conference on Machine Learning and Soft Computing, pp. 149–153 (2020)
15. Fan, W., et al.: Graph neural networks for social recommendation. In: Web Conference 2019 – Proceedings of World Wide Web Conference, WWW 2019, pp. 417–426 (2019). https://doi.org/10.1145/3308558.3313488
16. Zhen, F., Zhang, Z., Lu, H.: A review of social recommendation. In: Proceedings of 13th IEEE IEEE Conference on Industrial Electronics and Applications, ICIEA 2018, pp. 537–542 (2018). https://doi.org/10.1109/iciea.2018.8397775
17. Wang, H., Zhao, M., Xie, X., Li, W., Guo, M.: Knowledge graph convolutional networks for recommender systems. In: Web Conference 2019 – Proceedings of World Wide Web Conference, WWW 2019, pp. 3307–3313 (2019). https://doi.org/10.1145/3308558.3313417
18. Wang, H., Zhang, F., Xie, X., Guo, M.: DKN: deep knowledge-aware network for news recommendation. In: Web Conference 2018 – Proceedings of World Wide Web Conference, WWW 2018, pp. 1835–1844 (2018). https://doi.org/10.1145/3178876.3186175
19. Yang, Z., Dong, S.: HAGERec: hierarchical attention graph convolutional network incorporating knowledge graph for explainable recommendation. Knowl.-Based Syst. **204**, 106194 (2020). https://doi.org/10.1016/j.knosys.2020.106194
20. Wang, S., Tang, J., Wang, Y., Liu, H.: Exploring hierarchical structures for recommender systems. IEEE Trans. Knowl. Data Eng. **30**(6), 1022–1035 (2018)
21. Logani, M.K., Solanki, V., Slaga, T.J.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. Carcinogenesis **3**(11), 1303–1306 (1982). https://doi.org/10.1093/carcin/3.11.1303
22. van den Berg, R., Kipf, T.N., Welling, M.: Graph convolutional matrix completion. http://arxiv.org/abs/1706.02263 (2017)