Similarity-Based Heterogeneous Graph Attention Network for Knowledge-Enhanced Recommendation

Fan Zhang, Rui Li, Ke Xu, and Hongguang Xu(✉)

Harbin Institute of Technology, Shenzhen, China
{19s152091,19s152089,18b95055}@stu.hit.edu.cn, xhg@hit.edu.cn

Abstract. The introduction of knowledge graphs (KG) has improved the accuracy and interpretability of recommendations. However, the performance of KG-based recommender system is still limited due to the lack of valid modeling of user/item similarity and effective constraints on user/item embeddings learning. In addition, common sampling and propagation methods for homogeneous graphs do not apply to KGs due to their heterogeneity. In this work, we propose *Similarity-based Heterogeneous Graph Attention Network* (SHGAT), which learns both the collaborative similarity and knowledge similarity of items by pre-training item embeddings with user-item interaction data and knowledge propagation in the KG. Meanwhile, users are represented by the items they have interacted with, thus establishing similarity between users and strengthening the learning of item embeddings. Besides, we design an importance sampling and aggregation method based on attention mechanism for heterogeneous graphs. We apply the proposed model on two real-world datasets, and the empirical results demonstrate that SHGAT significantly outperforms several compelling state-of-the-art baselines.

Keywords: Recommender systems · Knowledge graph · Similarity modeling · Heterogeneous graph attention network · Importance sampling

1 Introduction

Recommender system has become an indispensable part in many Internet applications to alleviate information overload. Traditional recommendation methods based on collaborative filtering (CF) learn user/item similarity from their co-occurrence matrix to infer users' preferences [6,7,10], and they have been widely used in the recommendation field due to simplicity and low cost. However, these methods usually suffer from sparsity and cold-start problems in default of side information. Recently, knowledge graph (KG) has been proved to be an effective side information to alleviate the above two problems [4,13,15,16]. KG is a kind of heterogeneous graph, where entities act as nodes and relationships between entities act as edges. Items and their attributes can be mapped into the KG

H. Qiu et al. (Eds.): KSEM 2021, LNAI 12816, pp. 488–499, 2021.
https://doi.org/10.1007/978-3-030-82147-0_40

to understand the mutual relations between items [20]. Besides, various KGs have been proposed, such as Freebase [2], DBpedia [8] and YAGO [11], which makes it more convenient to build KGs for recommendation. The main challenge of KG-based recommendation is to learn efficient user/item embeddings from their interactions and KG's comprehensive auxiliary data [19]. However, most of the existing methods in this field lack sufficient discussion on user/item similarity modeling, which is the heart of collaborative filtering, and also the key to the performance of recommender system. In addition, many efforts have been devoted to applying graph neural networks (GNNs) to recommendation due to their superior performance in graph data learning. GNN is able to capture the high-order connectivity in graph data through iterative propagation, so as to integrate additional knowledge into user/item representation. For example, KGCN [15] uniformly samples a fixed size of neighbors for each entity as their receptive field, then utilizes user-specific relation-aware GNN to aggregate the information of entities in the neighborhood. KGAT [16] integrates users and items into one KG, and adopts the GAT mechanism [12] to fully exploit the relationship between entities. However, most of these models are more suitable for dealing with homogeneous graphs rather than heterogeneous graphs such as KGs. Therefore, more reasonable and effective model structures should be further investigated.

In this paper, we propose SHGAT for KG-based recommendation, with the aim to solve the above-mentioned shortcomings of existing methods. Our main contributions are summarized as follows:

- We adopt a two-stage item embedding learning method to capture both collaborative similarity and knowledge similarity of items. Meanwhile, users are represented by the items they have interacted with instead of random initialization, thereby spreading the item similarity to the user representations and making it easier to learn more effective item embeddings.
- We propose an importance sampling and aggregation strategy for heterogeneous graphs, in order to distinguish the differentiated interests of different users in a fine-grained way and aggregate significant neighbors of entities in a unified description space.
- We conduct extensive experiments on two public benchmarks, and the experimental results show that SHGAT significantly beats state-of-the-art methods in click-through rate prediction and top-K recommendation.

The remaining of this article is organized as follows: Sect. 2 formulates the KG-based recommendation problem. Section 3 presents the design of the SHGAT model. Section 4 demonstrates the experiments and discusses the results. Finally, Sect. 5 concludes this paper.

2 Problem Formulation

The KG-based recommendation problem is formulated as follows. In a typical recommendation scenario, we have a set of M users $\mathcal{U} = \{u_1, u_2, ..., u_M\}$ and a

set of N items $\mathcal{V} = \{v_1, v_2, ..., v_N\}$. The user-item interaction matrix $\mathrm{Y} \in \mathbb{R}^{M \times N}$ is defined according to users' implicit feedback, where $y_{uv} = 1$ indicates that user u has interacted with item v, otherwise $y_{uv} = 0$. In addition, we also have a knowledge graph $\mathcal{G} = \{(h, r, t)|h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where h, r, t denote the head, relation and tail in a knowledge triple, $\mathcal{E}$ and $\mathcal{R}$ are the sets of entities and relations in the KG, respectively. For example, the triple (*Titanic, film.film.director, James Cameron*) states the fact that the director of the film "Titanic" is James Cameron. Additional knowledge of an item can be obtained if it can be aligned with the corresponding entity in the KG through entity linking.

Given the user-item interaction matrix Y and knowledge graph $\mathcal{G}$, our goal is to learn a prediction function $\hat{y}_{uv} = \mathcal{F}(u, v|\Theta, \mathrm{Y}, \mathcal{G})$, where $\hat{y}_{uv}$ denotes the probability that user u will engage with item v, and Θ denotes the model parameters of function $\mathcal{F}$.

The notations we will use throughout the article are summarized in Table 1.

Table 1. Key notations used in this paper

Notations	Descriptions
$\mathcal{U} = \{u_1, u_2, ..., u_M\}$	Set of M users
$\mathcal{V} = \{v_1, v_2, ..., v_N\}$	Set of N items
u/v	Target user/item for CTR prediction
$\mathcal{S}_u$	Set of items interacted by user u
$\mathbf{e}_v$	Embedding of item v
$\mathbf{u}$	Representation of user u
$\mathbf{e}'$	Embedding of entity e
$\mathbf{W}_r$	Parameter matrix corresponding to relation r
$\mathcal{N}(\cdot)$	Set of neighbors in KG
π_e^u	Attention score of user u to entity e
$\mathcal{G}_{uv}^h$	The h-order KG subgraph of item v with respect to user u
$\mathbf{z}_{\mathcal{N}'(z)}^{(h)}$	The h-order neighbor representation of entity z
$\mathbf{z}^{(h)}$	The h-order knowledge representation of entity z
$\mathbf{v}^u = \mathbf{v}^{(H)}$	Overall knowledge representation of item v with respect to user u

3 Methodology

In this section, we introduce the SHGAT in detail. The workflow is shown in Fig. 1. The upper part models the similarity of items through Item2Vec and further spread it to user representations. The lower part calculates the user-specific attention scores for importance sampling and aggregation to obtain the knowledge representations of items. Finally, inner product is used to calculate the interaction probability between the user and the item.

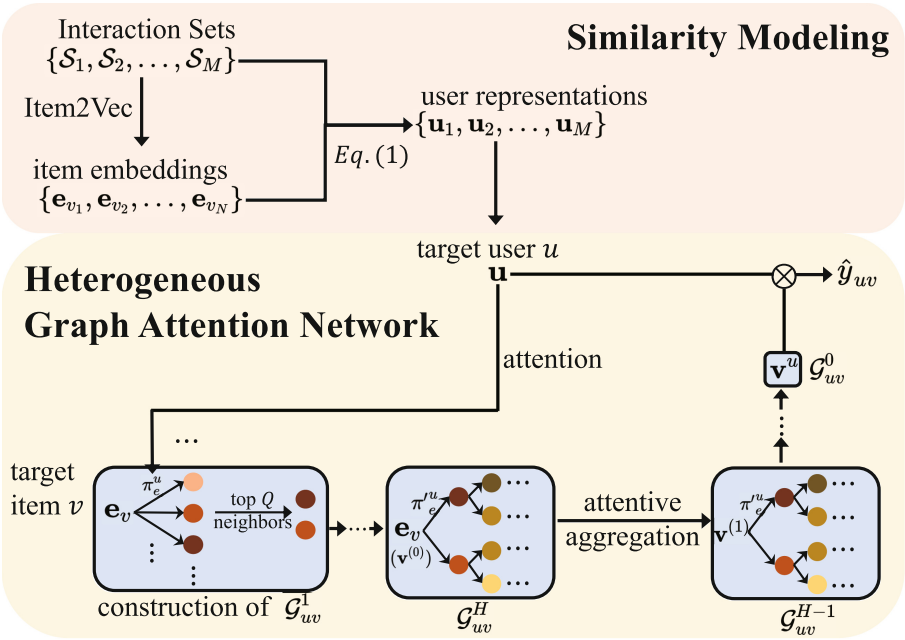


Fig. 1. The workflow of the proposed SHGAT

3.1 Similarity Modeling

Computing item similarities is the heart of Item-CF, and it is also a key building block in modern recommender systems [1]. Many studies focus on learning the low-dimensional embeddings of users and items simultaneously, but do not explicitly impose similarity constraints on these embeddings, which greatly limits the performance of recommender system. We model the user/item similarity based on three assumptions to get more effective user/item representation, namely: (1) Items preferred by the same user within a suitable time window are similar. (2) Items that share similar neighborhoods in the KG are similar. (3) Users with similar item preferences are similar.

Specifically, we denote the positive items that user u has interacted with as an interaction set: $\mathcal{S}_u = \{v|y_{uv} = 1\}$, and we can infer from the first assumption that similar items have similar interaction contexts whether they appear in the same interaction set or not, which is consistent with the idea in Word2Vec [9]. Word2Vec is a well-known algorithm to compute word vectors in Natural Language Processing, and Item2Vec [1] extends it to recommender system. Based on this, we take all the interaction sets $\{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_M\}$ as training data to use Item2Vec to get pre-trained item embeddings of dimension d: $\{\mathbf{e}_{v_1}, \mathbf{e}_{v_2}, ..., \mathbf{e}_{v_N}\}$, which will be further optimized with the training of SHGAT. In fact, some researches get entity embeddings through *knowledge graph embedding* (KGE), and then optimize them according to specific recommendation goals [16,20].

However, KGE focuses more on modeling strict semantic relationships (e.g., TransE [3] assumes that *head+relation=tail*), hence, the divergence of the optimization objectives is not conducive to the learning of user/item embeddings.

The initial item embeddings essentially contain the collaborative similarity learned from the historical user-item interactions, then we further model the knowledge similarity of items through knowledge propagation based on the second assumption. In practice, we use the proposed heterogeneous graph attention network to aggregate neighborhood information with bias when calculating the knowledge representations of entities. For the same user, similar neighborhoods produce similar aggregation results. Thus, the knowledge similarity in local proximity structure is successfully captured and stored in the knowledge representation of each entity. The knowledge representation of the item and its own embedding will jointly affect the prediction result $\hat{y}_{uv}$.

In order to model user similarity, we draw on the third assumption, which is the key to User-CF. In specific, we treat users as special items by averaging the item embeddings in their corresponding interaction sets in the training phase:

$$\mathbf{u} = \frac{1}{|\mathcal{S}_u|} \sum_{v \in \mathcal{S}_u} \mathbf{e}_v. \tag{1}$$

Thus, users and items share the same representation space, user similarity can be obtained from items and can be maintained all the time. Moreover, because the user embedding matrix is discarded, the amount of model parameters can be reduced greatly, and the item embeddings can be optimized more sufficiently. Such a practice is simple and effective, especially for the recommendation scenarios with large number of users whose embeddings are difficult to learn.

3.2 Heterogeneous Graph Attention Network

To deal with large-scale graphs, existing works adopt sampling methods to construct subgraphs of KG. Most of them follow the sampling strategy proposed in GraphSage [5], i.e., uniformly sampling a fixed size of neighbors of an entity. This way of subgraph construction may cause problems such as ignoring important entities and introducing noise. For example, a user pours more attention into the director and the genre of a movie, but if the sampled subgraph does not contain these two entities except some unimportant ones, it becomes difficult for the model to infer this user's preference in the movie reasonably and accurately. Besides, as KG is a typical heterogeneous graph, most algorithms designed for homogeneous graphs cannot be directly applied to KG [17]. To address the above problems, we transform the entity embeddings into a unified description space, and design a user-specific attention calculation method to sample subgraphs and perform knowledge propagation.

Specifically, consider a candidate pair of user u and item v, we use $\mathcal{N}(v)$ to denote the set of entities directly connected to the target item v, and then perform the entity embedding transformation:

$$\mathbf{e} = \mathbf{W}_r \mathbf{e}', \tag{2}$$

where $\mathbf{e}' \in \mathbb{R}^d$ is the embedding of eneity $e \in \mathcal{N}(v)$, and $\mathbf{W}_r \in \mathbb{R}^{d \times d}$ is a learnable parameter matrix of the relation r between e and v. As a result, the related entitiy embeddings are projected to a unified description space. For example, the movie "Titanic" can be described as "directed by James Cameron", "released in 1997" and so on. In order to calculate the user's preference for this movie, we calculate u's attention to the above different knowledge descriptions by inner product and then normalize the result:

$$\pi_e^u = \text{SoftMax}(\mathbf{u}^\top \mathbf{e}). \tag{3}$$

Then we rank the entities according to the attention scores and keep the top Q most important entities as set $\mathcal{N}'(v)$, then renormalize their attention scores to ${\pi'}_e^u$, where Q is a configurable hyperparameter. Now we have completed the construction of the one-hop sub-graph of v, denoted as $\mathcal{G}_{uv}^1$. Then we replace v with each sampled entity and repeat the above operation until we obtain the H-hop sub-graph $\mathcal{G}_{uv}^H$, where H is also a hyperparameter indicating the maximum depth of item v's receptive field. However, the flexibility of attention scores makes the learning process prone to overfitting, so we follow the method proposed in [14] to perform label smoothness regularization to get better attention scores.

In order to obtain an overall knowledge-enhanced representation of item v, we iteratively aggregate the neighborhoods for entities in the subgraph. Specifically, for each entity z in sub-graph $\mathcal{G}_{uv}^{H-1}$, we calculate the linear combination of its neighborhood as:

$$\mathbf{z}_{\mathcal{N}'(z)}^{(0)} = \sum_{e \in \mathcal{N}'(z)} {\pi'}_e^u \mathbf{e}^{(0)}, \tag{4}$$

where $\mathbf{e}^{(0)} = \mathbf{e}$, then we update the representation of z by combining its own representation $\mathbf{z}^{(0)}$ and neighborhood representation:

$$\mathbf{z}^{(1)} = \sigma(\mathbf{z}_{\mathcal{N}'(z)}^{(0)} + \mathbf{z}^{(0)}), \tag{5}$$

where σ is the nonlinear activation function. Note that for the target item v, $\mathbf{z}^{(0)} = \mathbf{v}^{(0)}$ is its own embedding $\mathbf{e}_v$ without projection. The above process is repeated until the sub-graph is reduced to $\mathcal{G}_{uv}^0$, which only contains the target item v and its overall knowledge representation $\mathbf{v}^u = \mathbf{v}^{(H)}$. With $\mathbf{v}^u$ and the target user's representation $\mathbf{u}$, we use a prediction function $f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ (i.e., inner product) to calculate the interaction probability between u and v:

$$\hat{y}_{uv} = \mathbf{u}^\top \mathbf{v}^u. \tag{6}$$

3.3 Model Training

To ensure effective model training, we sample the same number of negative samples as the positive samples for each user. Finally, we have the following loss function for SHGAT:

$$\mathcal{L} = \sum_{u,v} J(y_{uv}, \hat{y}_{uv}) + \lambda R(\mathcal{A}_{uv}) + \gamma \left\| \Theta \right\|_2^2, \tag{7}$$

where J is the cross-entropy loss function, $\mathcal{A}_{uv}$ is the attention scores in the user-specific sub-graph $\mathcal{G}_{uv}^{H}$, $R(\cdot)$ is the label smoothness regularization that can be seen as a constraint on the attention scores and the definition of $R(\cdot)$ can be found in [14], $\gamma \left\|\Theta\right\|_2^2$ is the L2-regularizer, λ and γ are balancing hyperparameters.

In order to explain the learning process of SHGAT more clearly, we summarize it into Algorithm 1.

Algorithm 1: Learning algorithm for SHGAT

Input: Interaction matrix Y, knowledge graph $\mathcal{G}$
Output: Prediction function $\mathcal{F}(u, v|\Theta, \mathrm{Y}, \mathcal{G})$

```
Construct sample set from Y, and split it into training set, validation set and test set;
Get initial item embeddings {e_{v_1}, e_{v_2}, ..., e_{v_N}} through Word2Vec with training set;
Construct interaction sets {S_1, S_2, ..., S_M} from training set;
Initialize all parameters;
while SHGAT not converge do
    for (u, v) in training set do
        Calculate user representation u on Eq.(1);
        for h = 1, ..., H do
            Do spatial transformation for related entities on Eq.(2);
            Calculate attention score π_e^u for each entity on Eq.(3);
            Filter the entities to get G_{uv}^h;
        for h = H, ..., 1 do
            Attentively aggregate and update entity representation on Eq.(4)-(5);
        Calculate predicted probability on Eq.(6);
        Update parameters by gradient descent;
return F;
```

4 Experiments

4.1 Datasets

We conduct our experiments against two realistic datasets, i.e., Movielens-20M and Last.FM. The codes and data are publicly available for reproducibility and further study.[1]

– **MovieLens-20M**[2] is a widely used dataset in movie recommendations, which contains nearly 20 million explicit ratings (ranging from 1 to 5) on the MovieLens website.

[1] https://github.com/GhostShipZ/SHGAT.
[2] https://grouplens.org/datasets/movielens/.

– **Last.FM**[3] is a music listening dataset collected from Last.fm online music systems and the tracks are viewed as the items.

We follow the procedures of [15] to process these two datasets, which are both linked to the sub-KGs extracted from the Microsoft KG Satori[4]. The basic statistics of the two datasets are presented in Table 2.

Table 2. Statistics and hyper-parameter settings (Q: neighbor sampling size, d: dimension of embeddings, H: depth of receptive field, λ: weight of label smoothness regularization, γ: weight of L2 regularization, η: learning rate.)

	MovieLens-20M	Last.FM
# users	138159	1872
# items	16954	3846
# interactions	13501622	42346
# entities	102569	9366
# relations	32	60
# KG triples	499474	15518
(Q, d, H, batch size)	(4, 128, 2, 8192)	(8, 16, 1, 256)
(λ, γ, η)	(0.1, 1×10^{-7}, 3×10^{-3})	(0.1, 1.5×10^{-4}, 2×10^{-3})

4.2 Baselines

– **BPRMF** [10] is a CF-based method that takes Matrix Factorization as the underlying predictor and minimizes the pairwise ranking loss for implicit feedback.
– **CKE** [20] combines CF with structural, textual, and visual knowledge for recommendation. We implement CKE as CF with a structural knowledge module in this paper.
– **RippleNet** [13] is a state-of-the-art model that propagates users' potential preferences in the KG for recommendation.
– **KGCN** [15] is another state-of-the-art model that aggregates neighborhood information with bias to learn the knowledge representations of items and users' personalized interests.
– **CKAN** [18] is also a state-of-the-art model that combines collaborative information with knowledge information together and learns user/item representations through a knowledge-aware attention mechanism.
– **SHGAT**$_{w/o\ c\_sim}$ is a variant of SHGAT that removes the collaborative similarity modeling of items, and item embeddings are initialized randomly.
– **SHGAT**$_{w/o\ u\_sim}$ is a variant of SHGAT that removes the user similarity modeling, and we assign a learnable embedding matrix for the users instead.

[3] https://grouplens.org/datasets/hetrec-2011/.
[4] https://searchengineland.com/library/bing/bing-satori.

– **SHGAT**$_{w/o\ hete}$ is another variant of SHGAT, it removes the importance sampling module and entity embedding transformation module, instead, we adopt the sampling and aggregation method proposed in KGCN.

4.3 Experimental Settings

In SHGAT, the activation function σ in Eq. (5) is set as *tanh* for the last aggregation and *ReLU* for the others. Other hyper-parameters are determined by optimizing $F1$ score on a validation set and the final settings of hyper-parameters are provided in Table 2. For the other baselines, we set the hyper-parameters according to their original papers. We split each dataset into training, validation and test set at a ratio of 6:2:2. Each experiment is repeated 3 times, and the average performance is reported. We evaluate these models in two experiment scenarios: (1) For click-through rate (CTR) prediction, we use the metrics *AUC* and $F1$ score. (2) For top-K recommendation, we use *Recall@K*. In terms of optimization, Adam algorithm is adpted to optimize all trainable parameters.

We use Skip-gram to pre-train item embeddings, where hierarchical softmax is enabled. The window size is set to 30 for both movie and music dataset. Since $|\mathcal{S}_u|$ and $|\mathcal{N}(e)|$ both have wide ranges, in order to ensure the effective use of video memory, we perform a non-replacement uniform sampling on them in advance, while retaining most of the information. The vacancies are padded with a special entity, which has no real impact. Specifically, for MovieLens-20M, the quantiles are set to 99 for interaction sets and 90 for entity neighbors, for Last.FM, the numbers are 99, 95. In addition, as the timestamps of user behavior are not available in the music dataset, we have to weaken the first assumption by removing the condition of time window.

4.4 Results

Table 3 summarizes the results of CTR prediction on the two datasets, and the results of top-K recommendation are presented in Fig. 2. The major findings from the experimental results are as follows:

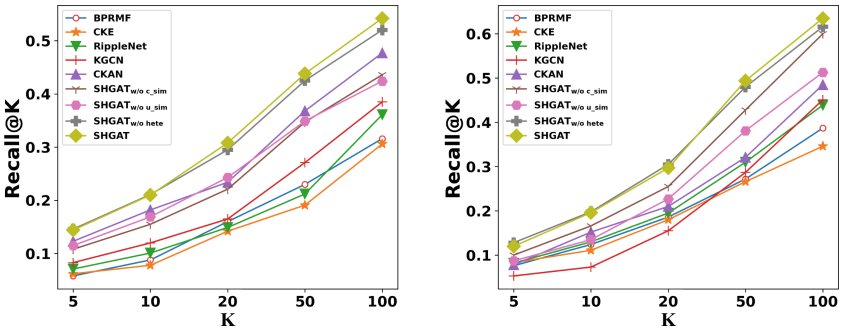
Compared with these state-of-the-art baselines, SHGAT achieves strongly competitive performance on both two datasets, which proves the superiority of our purposed methods. Especially for the movie dataset, SHGAT outperforms other baselines by a large margin *w.r.t.* *AUC*, $F1$, and the excellent performance *w.r.t.* *Recall* shows that SHGAT can not only predict users' preferences for items well, but also with very high accuracy.

The KG-aware model CKE is inferior to the CF-based model BPRMF, which indicates that it is difficult for recommendation to directly benefit from the KGE task. The remaining models are all based on knowledge propagation in the KG, and their overall performance is better than that of CKE and BPRMF, which proves that knowledge propagation is an effective way to utilize KG to enhance recommendation.

The experimental results of the two ablation versions,
SHGAT$_{w/o\ c_sim}$ and SHGAT$_{w/o\ u_sim}$, *w.r.t.* all three metrics in both datasets

Table 3. The results of *AUC* and *F*1 in CTR prediction.

Model	Last.FM		MovieLens-20M	
	AUC	*F*1	*AUC*	*F*1
BPRMF	0.752(–11.6%)	0.698(–9.7%)	0.962(–2.5%)	0.917(–3.2%)
CKE	0.745(–12.5%)	0.675(–12.7%)	0.931(–5.7%)	0.875(–7.6%)
RippleNet	0.777(–8.7%)	0.704(–8.9%)	0.976(–1.1%)	0.929(–1.9%)
KGCN	0.798(–6.2%)	0.715(–7.5%)	0.977(–1.0%)	0.931(–1.7%)
CKAN	0.845(–0.7%)	0.770(–0.4%)	0.972(–1.5%)	0.923(–2.5%)
SHGAT$_{w/o\ c_sim}$	0.839(–1.4%)	0.765(–1.0%)	0.984(–0.3%)	0.940(–0.7%)
SHGAT$_{w/o\ u_sim}$	0.819(–3.8%)	0.745(–3.6%)	0.978(–0.9%)	0.935(–1.3%)
SHGAT$_{w/o\ hete}$	0.847(–0.5%)	0.762(–1.4%)	0.987(0.0%)	0.947(0.0%)
SHGAT	**0.851**	**0.773**	**0.987**	**0.947**



(a) Last.FM (b) MovieLens-20M

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

prove the importance of similarity modeling for recommendation. What is more, we find out that fewer iterations are required for the model to achieve optimal performance if the similarity modeling is included. For example, in an experiment, the epochs corresponding to the best performance of the full SHGAT are 6, 13 for the movie and music datasets respectively, and the number is 12, 44 for SHGAT$_{w/o\ c_sim}$ and 34, 31 for SHGAT$_{w/o\ u_sim}$. It shows that similarity modeling makes it easier for the model to learn more effective item embeddings.

The comparison of full SHGAT and SHGAT$_{w/o\ hete}$ in CTR and top-K tasks shows that the adoption of importance sampling and heterogeneous aggregation can bring some benefits, but not as much as similarity modeling does. We believe that the following two reasons can explain this: (1) Due to the sparsity of the extracted sub-KGs of the two datasets, most of the entities have only a few neighbors. Thus, the filtering ability of importance sampling cannot be fully utilized. (2) The learned entity embeddings and attention scores are not effective

enough due to the lack of explicit semantic constraints, which poses a challenge to the optimization of SHGAT.

5 Conclusions

In this paper, we propose SHGAT, a similarity-based heterogeneous graph attention network for knowledge-enhanced recommendation. Firstly, similarity modeling, a key component that has been rarely explored recently, is used to learn more effective user/item representations. Secondly, the heterogeneous graph attention network, which is composed of importance sampling and heterogeneous aggregation, is used to better distinguish users' personalized interests and propagate knowledge in a more reasonable way. Extensive experimental results demonstrate the superiority of SHGAT over the state-of-the-art baselines on two public benchmark datasets.

Acknowledgments. This work was supported by Pengcheng Laboratory under Project "The Verification Platform of Multi-tier Coverage Communication Network for Oceans (PCL2018KP002)".

References

1. Barkan, O., Koenigstein, N.: Item2vec: neural item embedding for collaborative filtering. In: 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6. IEEE (2016)
2. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250 (2008)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS), pp. 1–9 (2013)
4. Guo, Q., et al.: A survey on knowledge graph-based recommender systems. IEEE Trans. Knowl. Data Eng. (2020). https://doi.org/10.1109/TKDE.2020.3028705
5. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. arXiv preprint arXiv:1706.02216 (2017)
6. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
7. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
8. Lehmann, J., et al.: Dbpedia-a large-scale, multilingual knowledge base extracted from Wikipedia. Semant. Web **6**(2), 167–195 (2015)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
10. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)

11. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web, pp. 697–706 (2007)
12. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
13. 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)
14. 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)
15. 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)
16. 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)
17. Wang, X., Bo, D., Shi, C., Fan, S., Ye, Y., Yu, P.S.: A survey on heterogeneous graph embedding: Methods, techniques, applications and sources. arXiv preprint arXiv:2011.14867 (2020)
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, pp. 219–228 (2020)
19. Wu, S., Zhang, W., Sun, F., Cui, B.: Graph neural networks in recommender systems: A survey. arXiv preprint arXiv:2011.02260 (2020)
20. 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)