Hierarchical Aggregation Based Knowledge Graph Embedding for Multi-task Recommendation

Yani Wang[1], Ji Zhang[2](✉), Xiangmin Zhou[3], and Yang Zhang[4]

[1] Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[2] The University of Southern Queensland, Toowoomba 4350, Australia
Ji.Zhang@usq.edu.au
[3] RMIT University, Melbourne 3001, Australia
[4] Zhejiang Lab, Hangzhou 311121, China

Abstract. Recently, knowledge graph has been used for alleviating the problems such as sparsity faced by the recommendation. Multi-task learning, which is an important emerged frontier research direction, helps complement the available information of different tasks and improves recommendation performance effectively. However, the existing multi-task methods ignore high-order information between entities. At the same time, the existing multi-hop neighbour aggregation methods suffer from the problem of over-smoothing. Also, the existing knowledge graph embedding methods in multi-task recommendation ignore the attribute triples in knowledge graph and recommendation tends to neglect the learning of user attributes. To mitigate these problems, we propose a multi-task recommendation model, called AHMKR. We use hierarchical aggregation and high-order propagation to alleviate the over-smoothing problem and obtain a better entity representation that integrates high-order information for multi-task recommendation. We leverage the text information of attribute triples, to improve the performance of knowledge graph in expanding the features of recommendation items. For users, we conduct fine-grained user learning based on the user attributes to capture user preferences in a more accurate matter. The experiments on the real-world datasets demonstrate the good performance of AHMKR.

Keywords: Recommender systems · Knowledge graph · Multi-task learning · Graph neural network

1 Introduction

Currently, many different kinds of side information are applied to alleviate the sparsity and cold start problems and help provide better recommendation, such as item category [2], social network, and knowledge graph. The use of knowledge graph in recommendation has attracted great research attention. Knowledge graph is a kind of structured knowledge base, which organizes scattered knowledge effectually. Its main goal is to describe various concepts existing in the real world and their relationships.

Supported by Natural Science Foundation of China (No. 62172372) and Zhejiang Provincial Natural Science Foundation (No. LZ21F030001).

B. Li et al. (Eds.): APWeb-WAIM 2022, LNCS 13423, pp. 174–181, 2023.
https://doi.org/10.1007/978-3-031-25201-3_13

The knowledge graph consists of entities, relations and attributes. Entities are objects in the actual world. Relations mean the relationships between objects or describe attributes of objects [5]. Knowledge graph is usually encoded as triples. There are two types of knowledge graph triples [7]. One is relation triples meaning triples which contain relationships between entities, e.g., ("Say It, Mo Yan", Author, Mo Yan). And the other one is attribute triples meaning triples which contain entities and its attributes, e.g., ("Say It, Mo Yan", Date of Publication, "July, 2007"). The existing recommendation models try many ways to improve the recommendation performance by combining knowledge graphs. Among these, the multi-task learning approach learns recommendation and KG-related tasks jointly and enables the knowledge graph to improve recommendation effectually. KG-related tasks can help recommendation get rid of local minimum, prevent recommendation from overfitting, and increase the generalization ability of recommendation. It is a relatively new and cutting-edge research direction.

However, the existing multi-task recommendation methods only model direct relations between entities, which ignore high-order information [6]. And the items associated with the entities are not enough to capture the complex and diversified preferences of users. At the same time, the existing multi-hop neighbour aggregation methods have the problem of over-smoothing. Obviously, the ignorance of relation is one of the causes [15]. Also, the existing knowledge graph embedding methods in multi-task recommendation neglect the attribute triples. Only structural information is unable to differentiate the meanings of relations and entities in different attribute triples [7]. It leads to unsatisfied recommendation effect affected by the sparsity and incompleteness of the knowledge graph. Moreover, the existing multi-task recommendation methods tend to ignore the fine-grained learning of user attributes. User attributes as part of user-side information are greatly vital. This results in the failure of fusing users' stable preferences contained in user attributes.

To mitigate these issues, we propose a multi-task recommendation model, AHMKR. Our model alternately trains recommendation and knowledge graph embedding task. Also, model shares features between items and entities. We take node-level and relation-level attention to aggregate and propagate neighbour information, so our model can alleviate over smoothing and obtain a better entity representation integrating high-order information. We process the text information of attribute triples, the performance of the knowledge graph is improved which is helpful for expanding the features of the recommendation items. In user-side, the user attributes are assigned different weights to obtain fine-grained user embedding and capture users' stable preferences. The experiments on the real-world datasets demonstrate the good performance of AHMKR.

2 Related Work

Our model can be counted as a kind of special embedding based recommendation. Embedding methods have gained many development. Most of the existing embedding based methods apply varieties of side information to improve the item representation. Meanwhile, the information also help learn user representation exactly. Some models learn user preferences straightly through bringing in users to construct user-item graph. Papers [1,10,12] apply multi-task learning approach to train recommendation and KG-related tasks which are the works closest to our method. The aggregation and propagation in our method utilizes the idea of Graph Neural Network (GNN). GNN is applied for recommendation increasingly. In recent research, the recommender systems based on GNN build models from two perspectives, one is social recommendation [3,4], another one is knowledge-graph-aware recommendation [9,11]. Our model can be counted as the latter.

3 Our Approach

In this part, We will introduce the proposed model AHMKR which is shown in the Fig. 1. The model contains four modules: user feature processing unit, recommendation, cross-compress unit and hierarchical aggregation based knowledge graph embedding.

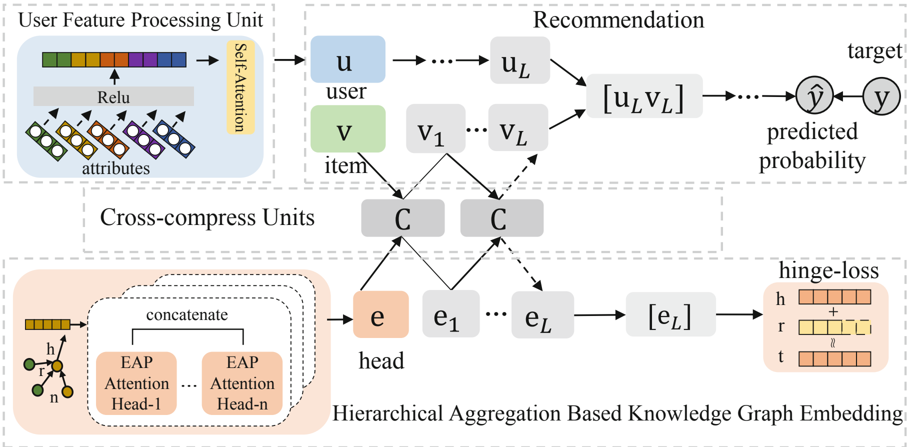

Fig. 1. Illustration of the AHMKR model.

3.1 User Feature Processing Unit

In the user feature processing unit, we process user attributes to get more accurate user feature vectors that can reflect users' stable preferences.

We apply multi-layer perceptron to extract features. Then we integrate the user attribute information and get the user feature vectors. However, we find the lack of expression about correlation and importance in attributes will make us left off information when mining user attribute features. So we take self-attention which is naturally suited to capture the internal relevance of data.

3.2 Recommendation

In the recommendation sub-module, we process the user vectors and item vectors to get the prediction result reflecting the user preferences for items. The input of recommendation is the feature vector $\mathbf{u}$ of user u which is the output result of the user feature processing unit and the feature vector $\mathbf{v}$ of item v which is the original feature of item.

For user u, we use L-layers multi-layer perceptron to process the feature. For item v, we acquire the feature through the processing of cross-compress units:

$$\mathbf{v}_L = E_{e\sim S(v)}\left[Cross^L[\mathbf{v}]\right] \tag{1}$$

Here, the associated entity set of item v is expressed by $S(v)$.

After processing the features of user u and item v, we predict the probability of user u participating in item v through applying the prediction function:

$$\hat{y}_{\mathrm{uv}} = \sigma f_{\mathrm{RS}}\left(\mathbf{u}_L, \mathbf{v}_L\right) \tag{2}$$

3.3 Hierarchical Aggregation Based Knowledge Graph Embedding

In this sub-module, we extract text information in attribute triples for better recommendation. As different meanings of relation and entity in different triples can't be distinguished by only using structural information, and it is easily affected by sparsity and incompleteness of knowledge graph. We utilize Long Short-Term Memory (LSTM) to encode attribute values of attribute triples into vectors.

To alleviate the over-smoothing problem and obtain a better entity representation, we conduct embedding aggregation propagation (EAP). Our embedding aggregation propagation is composed of node-level aggregation and relation-level aggregation.

In the node-level aggregation, we aggregate the neighbour nodes of head entities in specific relation. $\mathbf{h}_{\mathcal{N}\left(h_{r_i}\right)}$ means the neighbour embedding of specific relation.

$$\mathbf{h}_{\mathcal{N}\left(h_{r_i}\right)} = \sum_{t\in\mathcal{N}\left(h_{r_i}\right)} \pi^{r_i}(h,r,t)\mathbf{t} \tag{3}$$

We use softmax function on the significance of entity t to h in specific relation r_i:

$$\pi^{r_i}(h,r,t) = \frac{\exp\left(LeakyRelu\left((\mathbf{Wh})^T\mathbf{W}(\mathbf{r}+\mathbf{t})\right)\right)}{\sum_{t'\in\mathcal{N}\left(h_{r_i}\right)}\exp\left(LeakyRelu\left((\mathbf{Wh})^T\mathbf{W}(\mathbf{r}+\mathbf{t}')\right)\right)} \tag{4}$$

We aggregate the embedding of the corresponding head entity h with the neighbour embedding of specific relation, and the result is expressed as $\mathbf{h}'_{\mathcal{N}\left(\mathbf{h}_{\mathbf{r_i}}\right)}$.

$$\mathbf{h}'_{\mathcal{N}\left(h_{r_i}\right)} = f\left(\mathbf{h}, \mathbf{h}_{\mathcal{N}\left(h_{r_i}\right)}\right) \tag{5}$$

After node-level aggregation, We treat the enhanced entity embedding that incorporates neighbour nodes under particular relation as virtual entity, so that each type of relation is contained merely once in the allocation to alleviate the over-smoothing problem. And according to the relation between the entity and the virtual entity, we assign different attention weights to carry out relation-level aggregation.

We acquire different weights between entity and virtual nodes and normalize α_{r_i} to get normalization coefficient between the entity h and the neighbour of specific relation.

$$\alpha_{r_i} = \frac{\exp\left((\mathbf{h})^T \operatorname{Diag}(\mathbf{r}) \mathbf{h}'_{\mathcal{N}\left(h_{r_i}\right)}\right)}{\sum_{t=1}^{T} \exp\left((\mathbf{h})^T \operatorname{Diag}(\mathbf{r}) \mathbf{h}'_{\mathcal{N}\left(h_{r_t}\right)}\right)} \tag{6}$$

We get the final head entity embedding $\mathbf{H}$ after the relation-level aggregation.

$$\mathbf{H} = \sum_{i} \alpha_{r_i} \mathbf{h}'_{\mathcal{N}\left(h_{r_i}\right)} \tag{7}$$

To obtain the high-order connectivity information in the knowledge graph, we utilize embedding propagation to gather the deeper information of the neighbours so that we can capture the complex and diverse preferences of users.

$$\mathbf{H}^{(l)} = f\left(\mathbf{H}^{(l-1)}, \mathbf{h}'^{(l-1)}_{\mathcal{N}_h}\right) \tag{8}$$

After the aggregation and propagation, we use the cross-compress unit to share features of items and entities:

$$\mathbf{e}_L = E_{v \sim \mathcal{S}(h)} \left[Cross^L[\mathbf{e}] \right] \tag{9}$$

Here, the associated item set of entity h is expressed by $\mathcal{S}(h)$ and $\mathbf{e}$ means the feature of entity h.

3.4 Cross-compress Unit

The cross-compress unit shares the latent features of items and entities. It contains two operations, cross operation and compress operation.

In the cross operation, unit constructs a cross feature matrix. This is a fusion of item features and entity features. After the cross operation, unit conducts compress operation which projects the cross feature matrix into vectors of items and entities in their latent representation spaces to get the next layer feature vectors.

3.5 Optimization

We introduce the following loss function to train our model.

$$\mathcal{L} = \sum_{u \in U, v \in V} \mathcal{J}\left(\hat{y}_{uv}, y_{uv}\right) + \lambda_1 \sum_{(h,r,t) \in T} \sum_{(h',r,t') \in T'} \left[\gamma + d(h+r,t) - d\left(h'+r,t'\right)\right]_+ + \lambda_2 \|W\|_2^2 \tag{10}$$

Here, the first term is the loss of recommendation, the second term is the loss of knowledge graph embedding, $\lambda_2 \|W\|_2^2$ is the regularization term for preventing overfitting.

4 Experiments

4.1 Datasets

In our experiments, we use MovieLens dataset and Book-Crossing dataset. MovieLens-1M has 6036 users, 3152 items and 796132 interactions. The KG for it has 215332 triples. Book-Crossing contains 19676 users, 19875 items and 163724 interactions in the Book-Crossing community. The KG for it has 69754 triples.

4.2 Experiments Setup

The ratio of training, validation and test set in AHMKR is 6:2:2. We process each experiment three times to obtain the average result. We set $\lambda_2 = 10^{-7}$, and f_{RS} is the inner product. Aggregating two-order information is beneficial enough to get satisfied performance and keep relative low computation, so we set $l = 2$. We evaluate AHMKR in two scenarios. One is the CTR prediction. Another one is the top-K recommendation.

Table 1. Results of AUC and ACC in CTR prediction scenario.

Model	Movielens-1M		Book-crossing	
	AUC	ACC	AUC	ACC
MKR	0.917 (–1.93%)	0.843 (–1.86%)	0.734 (–2.00%)	0.704 (–2.36%)
Wide&Deep	0.898 (–3.96%)	0.820 (–4.54%)	0.712 (–4.94%)	0.624 (–13.45%)
DKN	0.655 (–29.95%)	0.589 (–31.43%)	0.622 (–16.96%)	0.598 (–17.06%)
CKE	0.801 (–14.33%)	0.742 (–13.62%)	0.671 (–10.41%)	0.633 (–12.21%)
PER	0.710 (–24.06%)	0.664 (–22.70%)	0.623 (–16.82%)	0.588 (–18.45%)
AHMKR	**0.935**	**0.859**	**0.749**	**0.721**
AHMKR-U	0.927 (–0.86%)	0.852 (–0.81%)	0.741 (–1.06%)	0.714 (–0.97%)
AHMKR-E(a)	0.925 (–1.07%)	0.851 (–0.93%)	0.739 (–1.34%)	0.711 (–1.39%)
AHMKR-E(h)	0.929 (–0.64%)	0.854 (–0.58%)	0.744 (–0.67%)	0.716 (–0.69%)

4.3 Performance

Overall Comparison. Table 1 is the AUC and ACC of various models in CTR prediction scenario. Figure 2 is the result of Recall@K and Precision@K in top-K scenario. We can observe that PER [13] gets unsatisfactory results as rational design of meta-paths is difficult. DKN [8] has poor performance as it is better proper for recommendation about long text. CKE [14] performs better than DKN as it utilizes the structured content in the knowledge graph. Wide&Deep merely connects the attributes, so the effect is mediocre. MKR shares features between user-item interaction and knowledge graph so that MKR [10] performs well. Our method outperforms all baselines, because we take hierarchical aggregation propagation to obtain high-order information and process the text information of attribute triples to expand the features of the recommendation items. Also, we extract user attributes to capture users' more accurate preferences.

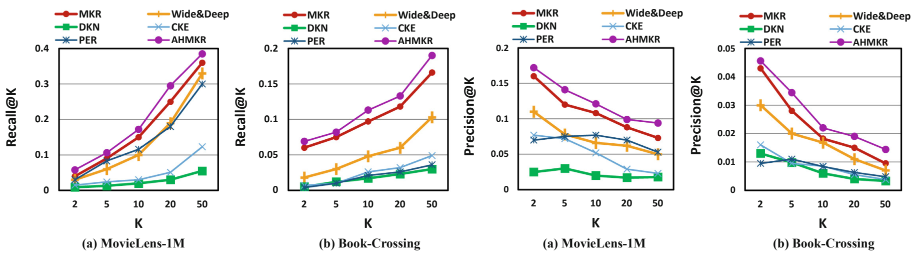

Fig. 2. The results of Recall@K and Precision@K in top-K recommendation scenario.

Ablation Study. Our model is a framework composed of sub-modules. For different modules, we conduct ablation study to assess whether they are beneficial to the goal. AHMKR-U is a variant that doesn't process attribute triples or conduct hierarchical aggregation propagation. AHMKR-E(a) is a variant that doesn't process the user attributes or conduct hierarchical aggregation propagation. AHMKR-E(h) is another variant which doesn't process the user attributes or process attribute triples. Table 1 shows the performance of the variants. We can find that the user feature extraction which captures stable preferences is helpful for better recommendation according to the performance of AHMKR-U. AHMKR-E(a) proves the importance of the text information in attribute triples which expands the features of recommendation items. The variant AHMKR-E(h) has a relatively good performance as hierarchical aggregation propagation obtains better entity representation for multi-task recommendation. Above all, the satisfying performance is achieved when we combine all the sub-modules.

Table 2. Results of AUC in movie CTR prediction with different ratios of training set r.

Model	20%	40%	60%	80%	100%
MKR	0.874	0.882	0.897	0.908	0.917
Wide & Deep	0.802	0.815	0.840	0.876	0.898
DKN	0.582	0.601	0.620	0.638	0.655
CKE	0.692	0.716	0.754	0.775	0.801
PER	0.607	0.638	0.662	0.688	0.710
AHMKR	**0.887**	**0.895**	**0.911**	**0.921**	**0.935**

Results in Sparse Scenarios. We verify the performance of models in sparse scenarios by adjusting the proportion of the training set. We test the proportions of the training set as 100%, 80%, 60%, 40%, 20% respectively. Table 2 is the performance in sparse scenarios. It can be found that with the raising of the sparsity, the overall performance of all models declines. However, our approach outperforms other baselines consistently. More importantly, our method still performs well when the training set is 20%, indicating that AHMKR can sustain satisfying performance even when the data is badly sparse.

5 Conclusions

In this work, we propose a multi-task recommendation model. We take hierarchical aggregation propagation to alleviate over smoothing and obtain entity representation integrating high-order information. Also, we extract the text information of attribute triples which is helpful for expanding the features of the items. In user-side, we process fine-grained attribute embedding to capture users' stable preferences. Experiments on the real-world datasets demonstrate that AHMKR has accomplished better performance than other models.

References

1. Cao, Y., Wang, X., He, X., Hu, Z., Chua, T.S.: Unifying knowledge graph learning and recommendation: towards a better understanding of user preferences. In: The World Wide Web Conference (2019)
2. Du, J., Zheng, L., He, J., Rong, J., Wang, H., Zhang, Y.: An interactive network for end-to-end review helpfulness modeling. Data Sci. Eng. **5**, 261–279 (2020)
3. Fan, W., et al.: Graph neural networks for social recommendation. In: The World Wide Web Conference (2019)
4. Fan, W., et al.: A graph neural network framework for social recommendations. IEEE Trans. Knowl. Data Eng. **34**, 2033–2047 (2020)
5. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A survey on knowledge graphs: representation, acquisition and applications. IEEE Trans. Neural Netw. Learn. Syst. **33**, 494–514 (2020)
6. Liu, W., Cai, H., Cheng, X., Xie, S., Yu, Y., Zhang, H.: Learning high-order structural and attribute information by knowledge graph attention networks for enhancing knowledge graph embedding. ArXiv abs/1910.03891 (2019)
7. Sun, Z., Hu, W., Li, C.: Cross-lingual entity alignment via joint attribute-preserving embedding. In: SEMWEB (2017)
8. 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 (2018)
9. Wang, H., et al.: Knowledge graph convolutional networks for recommender systems with label smoothness regularization. ArXiv abs/1905.04413 (2019)
10. 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 (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 (2019)
12. Xin, X., He, X., Zhang, Y., Zhang, Y., Jose, J.M.: Relational collaborative filtering: Modeling multiple item relations for recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (2019)
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 (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 (2016)
15. Zhao, X., Jia, Y., Li, A., Jiang, R., Chen, K., Wang, Y.: Target relational attention-oriented knowledge graph reasoning. Neurocomputing **461**, 577–586 (2021)