A Neural User Preference Modeling Framework for Recommendation Based on Knowledge Graph

Guiming Zhu[1,2], Chenzhong Bin[2(✉)], Tianlong Gu[2], Liang Chang[2], Yanpeng Sun[2], Wei Chen[2], and Zhonghao Jia[2]

[1] School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
jackming555@gmail.com
[2] Guangxi Key Lab of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China
binchenzhong@163.com, w_chen369@163.com, {cctlgu,changl}@guet.edu.cn, yanpeng_sun@yeah.net, 1090994959@qq.com

Abstract. To address the data sparsity and cold start problems in the traditional recommender systems, lots of researchers aim at incorporating knowledge graphs (KG) into recommender systems to enhance the recommendation performance. However, existing efforts mainly rely on hand-engineered features from KG (e.g., meta paths), which requires domain knowledge. What's more, as relations are usually excluded from meta paths, they hardly specify the holistic semantics of paths. To address the limitations of existing methods, we propose an end-to-end neural user preference modeling framework (UPM) to incorporate features of entity and relation of KG into the representations of users and items, so as to learn user latent interests precisely. Specifically, UPM first propagate user's interests along links between entities in KG iteratively to learn user's potential preferences for the item. Furthermore, these preference features are dynamically during the preference propagation process. That is to say, the importance of these preference features to characterize user is different. Therefore, an attention network is used in UPM to calculate the influence of preference features at different propagating stages, then the final preference vector of the user is calculated from the preference features and the corresponding weights. Lastly, the final prediction probability of user-item interaction is obtained by inner product operation between the embedding of item and user. To evaluate our framework, extensive experiments on two real-world datasets demonstrate significant performance improvements over state-of-the-art methods.

Keywords: Recommender systems · Knowledge graph · User modeling · Preference propagation

A. C. Nayak and A. Sharma (Eds.): PRICAI 2019, LNAI 11670, pp. 176–189, 2019.
https://doi.org/10.1007/978-3-030-29908-8_14

1 Introduction

With the rapid development of the Internet, user's personalized needs have been constantly improving. How to help users get the information they need and how to address the information overload are the research hotspots in recommender systems field. Traditional collaborative filtering based recommender systems only use historical interactive information (explicit or implicit feedback) of user and item as input. This brings two problems: First, the interactive information between users and items is usually very sparse. Second, since the systems do not have historical interactive information, it cannot represent user accurately by historical interests and preferences of user, nor can it push personalized information to users. This situation is called cold start problem.

A common way to address the problems of data sparsity and cold start is to introduce some additional auxiliary information as a complementary of the recommendation algorithm. Recently, KG, which is a type of directed heterogeneous graph, has attracted a lot of researcher's attention due to large quantity of entities and concepts and rich semantic relations [1]. KG contains various types of information related to entities in the form of triplet which is expressed as (h, r, t), where h, r and t are head entity, relation and tail entity respectively, e.g. (Saving Private Ryan, directed, Spielberg). The form of triplet can seamlessly integrate user-item interactive data and improve the sparsity of interactive data.

At present, the methods of introducing KG into recommender systems can be divided into two categories: feature-based method and path-based method. The feature-based approaches unify features of users and items as input of recommendation algorithms [2]. However, these methods are not specifically designed for KG, so it cannot utilize all the information of KG effectively. For example, feature-based methods fail to learn multi-hop relational knowledge. To address this weakness, path-based approaches regard KG as a heterogeneous information network, and constructs meta path-based features between items [3]. A meta path is a specific path linking two entities. For example, there is a path (Tom Hanks → The Terminal → Stephen Spielberg → Schindler List) linking Tom Hanks and Schindler List, so this path can be used as a way to mining the potential relation between actors and movies. However, these methods heavily rely on handcrafted features to encode the semantics of path, which further relies on domain knowledge. Furthermore, this approach cannot be applied in where entities do not belong to the same domain (e.g. news recommendation) [4], and the meta paths cannot be predefined.

To address the problems mentioned above, we propose a novel neural user preference modeling framework (abbr. UPM), which takes user-item interaction as input data and predicts the probability of a user interact with a particular item. Specifically, for each user, each item he has interacted with is regarded as a seed item in the KG, and extends the user's interests iteratively along the links in the KG. In this process, the preference features at different stages of the user with respect to the candidate item can be learned, and the influence of the preference features are different to characterize user, thus, we propose get the weights of different preference features through an attention

mechanism. After get the weights, UPM takes the sum of different preference features weighted by the corresponding weight, and the final preference vector of a user is generated. Finally, the probability of user-item interaction (e.g. a clicking or browsing action) is calculated by inner product of embedding of user and item. The experimental results on real-world datasets show that the proposed framework outperforms all of the baseline methods in click through rate (CTR) task.

The major contributions of this paper are as follows:

- We propose innovatively combines feature-based methods, path-based methods and attention mechanism in KG-aware recommendation.
- In order to introduce KG into recommender systems, an end-to-end user preference modeling framework (UPM) is proposed to mine the potential preference of user automatically by a user preference propagating process in the KG.
- To distinguish the importance of preference features at different propagating stages to characterize user, we propose calculate a weight for each preference features by an attention network, and make the preference features contribute to the preference vector of user according to the importance weights.
- Compared with the baseline methods, the proposed model performs best on two real-world datasets, indicating the superiority of our model.

2 Related Work

In this section, we mainly introduce the related work of introducing KG into recommender systems, i.e. feature-based and path-based methods. And the attention mechanism used in KG-aware recommendation.

2.1 Introducing KG into Recommender Systems

Feature-Based Methods. In the news recommendation scenario, Wang et al. [4] proposed to fuse the word vectors of news headlines, the entity vectors of KG and the entity context vectors, to generate the vector representation of news. Huang et al. [5] used TransE [6] to generate vector representations of entities and item, and then updates user's vector representations through memory networks based on user preferences for specific entities. Compared with other existing methods, feature-based methods have better performance. However, these methods ignore the semantics of the relations between entities represented by paths, so it cannot fully obtain the rich semantics of KG. On the other hand, since the links between users and items are realized by an implicit way, the regularization term of KG feature learning cannot fully discover the links between users and items.

Path-Based Methods. In the path-based approaches, some previous studies [7, 8] referred to the link patterns between KG entities as meta paths, and used meta paths to improve the performance of recommendations. Meta paths are defined as a sequence of

entity types, e.g. a meta path (user → movie → director → movie) obtain user-item related attributes contained in KG. Yu et al. [3] proposed HeteMF to factorize the user-item rating matrix and constrain the distance between latent vectors of similar entities by a graph regularization method.

Meanwhile, there are some other works aim at using meta paths to model user-user or user-item relations. Luo et al. [9] proposed Hete-CF to model user-item, user-user and item-item relations based on the similarity of meta paths. Shi et al. [10] proposed SemRec model and introduced the concept of weighted meta path, which aims at describing the path semantics by distinguishing the nuances between link attribute values. Wang et al. [11] design a matrix factorization method by regularizing the user-user relation using the calculated similarity based on meta paths.

However, the above methods heavily depend on the quality and quantity of meta paths, what's more, the sequence dependencies of entities and relations in meta paths are neglected, which limits the quality of the generated recommendations.

2.2 Attention Mechanism in Recommendation

Attention mechanism shows the effectiveness in various machine learning tasks, such as machine translation [12], text categorization [13] et al. Recently, more and more researchers have applied attention mechanism to recommendation tasks. For example, Pei et al. [14] used the attention network to capture the joint effects of user-item interaction and measure the relevance between users and item. Chen et al. [15] proposed item-level and component-level attention mechanisms to model implicit feedback in multimedia recommendation.

Compared with the simple path-based and feature-based approaches, UPM combines merits of path-based and feature-based approaches to model user's preferences through rich semantic information contained in the KG, and obtain embedding of users by an attention network. Compared with the existing methods, UPM can automatically learning the semantic relations of entities and the sequence dependencies of entities and relations in the path.

3 Neural User Preference Modeling Framework

In this section, we present the proposed UPM framework in detail.

3.1 Notations and Definition

Table 1 summarizes all the notations used in this paper. The user-item interaction matrix $Y = \{y_{uv} | u \in U, v \in V\}$, if the interaction between u and v is observed $y_{uv} = 1$, otherwise $y_{uv} = 0$. A KG G consists of a large number of triplets (h, r, t), where h, r and t are the head entity, relation and tail entity of G respectively.

The relevant definitions are as follows:

Table 1. Notations and descriptions

Notations	Descriptions
$U = \{u_1, u_2, \ldots, u_m\}$	User set
$V = \{v_1, v_2, \ldots, v_n\}$	Item set
$\mathrm{Y} \in \mathrm{R}^{m \times n}$	User-item interaction matrix
$\epsilon = \{e_1, e_2, \ldots e_e\}$	Entity set
$R = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, r_r\}$	Relation set
G	KG
ε_u^k	k-hop relevant entities set of user u
S_u^k	k-hop triplets set of user u
$H_u = \{h_1, h_2, \ldots, h_t\}$	Historical interaction record of u
P_i	Relevance probability
O_k^u	k-hop preference features of user u
Att	Attention network
w_k	Weight of k-hop preference features of u
v	Embedding of item v
u	Embedding of user u
y_{uv}	Predicted probability that u interact with v

Definition 1 (KG). Define $\boldsymbol{\varepsilon} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, e_e\}$, $\boldsymbol{R} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, r_r\}$ denote the sets of entities and relations respectively. G = ($\boldsymbol{\varepsilon}$, $\boldsymbol{L}$) is a directed graph with an entity type mapping function $\phi : \boldsymbol{\varepsilon} \rightarrow A$ and a link type mapping function $\psi : L \rightarrow \boldsymbol{R}$. Each entity $\boldsymbol{e} \in \boldsymbol{\varepsilon}$ belongs to an entity type $\phi(\boldsymbol{e}) \in A$, and each link $\boldsymbol{r} \in L$ belongs to a link type (relation) $\psi(r) \in \boldsymbol{R}$ [3].

Definition 2 (Relevant Entity). Given user-item interaction matrix Y and the k-hop relevant entities set of user u is defined as follows:

$$\varepsilon_u^k = \left\{t | (h, r, t) \in \mathrm{G} \,\&\, h \in \varepsilon_u^{k-1}\right\}, k = 1, 2, \ldots, K \tag{1}$$

Where $\varepsilon_u^0 = H_u = \{v | y_{uv} = 1\}$, i.e. the historical interaction record of user u [4].

Relevant entities can be regarded as the natural extensions of a user's interest in the KG. Given the definition of the relevant entity, the k-hop triplets set of user u is defined as follows:

Definition 3 (Set of Triplets). The k-hop triplets set of user u is defined as the set of triplets from ε_u^{k-1} [4]:

$$S_u^k = \left\{(h, r, t) | (h, r, t) \in \mathrm{G} \,\&\, h \in \varepsilon_u^{k-1}\right\}, k = 1, 2, \ldots, K \tag{2}$$

With the increase of hop number k, the set of triplets may become very large, which will greatly increase the computational overhead. In order to address the problems, we

proposes the following restrictions: (1) In a specific recommendation scenario (such as movie recommendation), the relations in the KG can be limited to movie-related attributes. (2) In practice, the total number of hop K is generally not very large, because entities locating far away from user history interaction items may be irrelevant to user latent preference. In this paper, $K = 2$ or 3.

3.2 Architecture of Framework

The framework of UPM is illustrated in Fig. 1. UPM takes a user u and an item v as input of the framework, and outputs the probability that the user u will interact (click, browse, etc.) with the item v. Specifically, for the input user u, his historical interaction record H_u is treated as seeds in the KG, then extended along links to form multiple triplet sets S_u^k ($k = 1, 2, \ldots, K$). A triplet set S_u^k is the set of knowledge triplets that are k-hops away from the seed set H_u. And the user's preference features (the dark blue, olive and yellow blocks) at different hops are obtained through extended interests of user iteratively along the links in triplet sets S_u^k. Then the preference features of user and embedding (the light blue block) of item are input into the attention network simultaneously, and the final preference vector (the pink block) of user is calculated. The probability y_{uv} of user-item interaction can be obtained by inner product between the embedding of item v and user u.

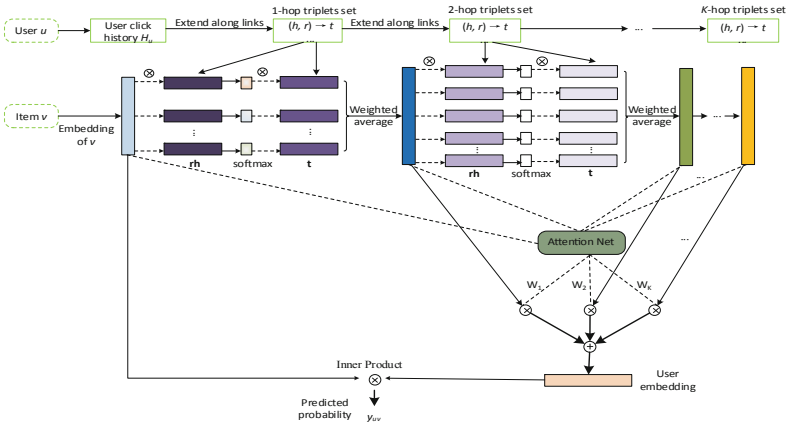


Fig. 1. The framework architecture. The light blue part is the embedding of item v, the dark blue part, the olive part and the yellow part are 1-hop, 2-hop and K-hop preference features of user respectively. (Color figure online)

3.3 1-Hop Preference Feature of User

The traditional collaborative filtering methods firstly learns the latent representation of users and items, then calculates the predicted probability through the inner product. In order to model user-item interaction more accurately, we proposes a neural user preference modeling framework to represent potential preferences of users.

As shown in Fig. 1, each item v has an associated embedding $\mathbf{v} \in \mathrm{R}^d$, d is dimension of the embedding. Each item embedding is generated by the attributes of this item. Given $\mathbf{v}$ and 1-hop triplets set S_u^1 of user u, by calculating similarity between item v, head entity h_i and entity relations r_i in S_u^1, each triplet in S_u^1 is assigned a relevance probability $\boldsymbol{P_i}$:

$$\boldsymbol{P_i} = \mathrm{softmax}\left(\mathbf{v^T r_i h_i}\right) = \frac{\exp(\mathbf{v^T r_i h_i})}{\sum_{(h,r,t)\in \boldsymbol{S_u^1}} \exp(\mathbf{v^T r h})} \tag{3}$$

Where $\mathbf{r_i} \in \mathrm{R}^{d\times d}$, $\mathbf{h_i} \in \mathrm{R}^d$ are the vector representation of r_i and h_i respectively, softmax function ensures that the sum of all calculated relevance probabilities is 1. $\boldsymbol{P_i}$ can be regarded as the similarity between item v and head entity h_i on entities relations r_i. It should be noted that the vector representation $\mathbf{r_i}$ of r_i must be taken into account when calculating the above relevance probability $\boldsymbol{P_i}$, because the similarity between item v and head entity h_i may be different on different entities relations. For example, "Saving Private Ryan" and "Schindler's List" are highly similar when considering director and genre, but they are completely different from the actor attribute.

After obtaining the relevance probability $\boldsymbol{P_i}$ of each triplet (h_i, r_i, t_i) in 1-hop triplets set S_u^1, all tail entity t_i of triplets in S_u^1 are weighted by the corresponding relevance probability $\boldsymbol{P_i}$, and the 1-hop preference feature $\boldsymbol{O_u^1}$ of user u is given by:

$$\boldsymbol{O_u^1} = \sum\nolimits_{(h_i,r_i,t_i)\in \boldsymbol{S_u^1}} \boldsymbol{P_i}\mathbf{t_i} \tag{4}$$

Where $\mathbf{t_i} \in \mathrm{R}^d$ is the vector representation of tail entity t_i.

3.4 Preference Propagation

There are rich semantic relations between entities, more complete user preferences can be obtained by the extension of entities and relations. Through the operation in Eqs. (3) and (4), the interest preferences of user u can be propagated from his historical interaction record H_u along the links in the 1-hop triplets set S_u^1 to his 1-hop relevant entities set ε_u^1. This process is called preference propagation.

The preference propagation process is repeated by replacing the embedding $\mathbf{v}$ of item v in (3) with the 1-hop preference features $\boldsymbol{O_u^1}$ of user u. As shown in Fig. 1, $\boldsymbol{O_u^1}$ as u's historical preference is propagated along the links in 2-hop triplets set S_u^1 to his 2-hop relevant entities ε_u^2, repeating the operation in Eqs. (3) and (4) to obtain u's 2-hop preference features $\boldsymbol{O_u^2}$, which is iteratively performed on user u's k-hop triplets $S_u^k (k = 1, 2, \ldots, K)$. Therefore, a user's preference is propagated from his historical interaction record H_u to K-hop relevant entities ε_u^K. Thus, the preference features of user u at different hops can be obtained: $\boldsymbol{O_u^1}, \boldsymbol{O_u^2}, \ldots \boldsymbol{O_u^K}$. The final preference vector of user u with respect to item v can be obtained by simply combining the preference features of user u at different hops:

$$\mathbf{u} = \boldsymbol{O}_u^1 + \boldsymbol{O}_u^2 + \ldots + \boldsymbol{O}_u^K \tag{5}$$

In theory, with the increase of hop number k, the preference feature $\boldsymbol{O}_u^k$ of user u in the last hop contains all the information of the previous preference features, but they may be weakened in $\boldsymbol{O}_u^K$, so the preference features at all hops must be superimposed.

3.5 Attention-Based User Preference Extraction

The above method does not take into account that the weights of user preference features $\boldsymbol{O}_u^1, \boldsymbol{O}_u^2, \ldots \boldsymbol{O}_u^K$ at different hops to user's final preference vector are different. As shown in Fig. 1, to model the different effects of user preference features $\boldsymbol{O}_u^1, \boldsymbol{O}_u^2, \ldots \boldsymbol{O}_u^K$ on the final preference vector of user u, we proposes calculate the weight w_k of k-hop preference features of user u by an attention network Att, $\boldsymbol{w_k}$ formulated by:

$$\boldsymbol{w_k} = softmax\left(Att\left(\mathbf{v}, \boldsymbol{O}_u^k\right)\right) = \frac{exp\left(Att\left(\mathbf{v}, \boldsymbol{O}_u^k\right)\right)}{\sum_{k=1}^{K} exp\left(Att\left(\mathbf{v}, \boldsymbol{O}_u^k\right)\right)} \; k = 1, 2, \ldots, K \tag{6}$$

Attention network Att takes user preference features $\boldsymbol{O}_u^1, \boldsymbol{O}_u^2, \ldots \boldsymbol{O}_u^K$ at different hops and embedding of item v as input, and outputs the corresponding weights $\boldsymbol{w_k}$ of $\boldsymbol{O}_u^1, \boldsymbol{O}_u^2, \ldots \boldsymbol{O}_u^K$. The weight $\boldsymbol{w_k}$ can be regarded as the important scores of user preference features at different hops,$\boldsymbol{w_k}$ adaptively select the informative preference features with different importance, and make the informative preference features contribute more to characterize preference vector of user u. Then we sum up the user preference features $\boldsymbol{O}_u^1, \boldsymbol{O}_u^2, \ldots \boldsymbol{O}_u^K$ at different hops according to the weight $\boldsymbol{w_k}$ provided by Att to get the final preference vector of user u:

$$\mathbf{u} = \sum\nolimits_{k=1}^{K} \boldsymbol{w_k} \boldsymbol{O}_u^k \tag{7}$$

Finally, given the embedding of user u and item v, the probability of the user interact with the item is calculated by inner product:

$$y_{uv} = \sigma\left(\mathbf{u}^{\mathbf{T}}\mathbf{v}\right) \tag{8}$$

Where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function.

3.6 Model Optimization

Given G and implicit feedback matrix Y, the objective of model optimization is to maximize the posterior probability of model parameter Θ:

$$\max p(\Theta | G, \mathrm{Y}) \tag{9}$$

Θ includes the vector representations of all entities, relations. So it's equivalent to maximizing:

$$p(\Theta|G,\mathrm{Y}) = \frac{p(\Theta,G,\mathrm{Y})}{p(G,\mathrm{Y})} \propto p(\Theta)\cdot p(G|\Theta)\cdot p(\mathrm{Y}|\Theta,G) \tag{10}$$

Taking the negative logarithm of (10) and have the following loss function:

$$\begin{aligned} \min L &= -\log(p(\Theta)\cdot p(G|\Theta)\cdot p(\mathrm{Y}|\Theta,G)) \\ &= \sum_{(u,v)\in \mathrm{Y}} -\left(y_{uv}\log\sigma\left(\mathrm{u}^{\mathrm{T}}\mathrm{v}\right) + (1-y_{uv})\log\left(1-\sigma\left(\mathrm{u}^{\mathrm{T}}\mathrm{v}\right)\right)\right) \\ &+ \frac{\lambda_2}{2}\sum_{r\in R}||\mathrm{I}_r - \mathrm{E}^{\mathrm{T}}\mathrm{R}\mathrm{E}||_2^2 + \frac{\lambda_1}{2}\left(||\mathrm{V}||_2^2 + ||\mathrm{E}||_2^2 + \sum_{r\in R}||\mathrm{R}||_2^2\right) \end{aligned} \tag{11}$$

Where **V**, **R** and **E** are the embedding matrices for all items, relation and entities, respectively, I_r is the slice of the indicator tensor I in the KG. The stochastic gradient descent (SGD) algorithm is used to iteratively optimize the loss function. In order to make the calculation more efficient in each training process, positive (negative) records of the smallest batch are sampled randomly from Y and positive (negative) triplets are sampled from G. The gradient of loss L relative to model parameter Θ is calculated, and all parameters are updated by back propagation algorithm.

4 Experiments and Analysis

In this section, the framework is evaluated by compared with the baseline methods on MovieLens-1M and Book-Crossing datasets.

4.1 Datasets and Preprocessing

The proposed framework is evaluated on two real-world datasets from different domains: MovieLens-1M and Book-Crossing. MovieLens-1M contains about 1 million user ratings (ranging from 1 to 5) on movie websites. Book-Crossing contains 1,149,780 explicit ratings (ranging from 0 to10) of books. In this experiment, we use the pre-processed data in [4]. Because MovieLens-1M and Book-Crossing are explicit feedback data, we transform them into implicit feedback data. Similar to [4], the ID embedding of users and items are used as the original input of framework in this experiment. The data statistics are shown in Table 2.

Table 2. The statistics of datasets

Datasets		MovieLens-1 M	Book-Crossing
User-item interaction	#Users	6,036	17,860
	#Items	2,445	14,967
	#Ratings	753,772	139,746
	#Data Density	5.108%	0.0523%
KG	#Entities	182,011	77,903
	#Links	12	25
	#The first 4-hops triplets	1,440,815	241,163

4.2 Baselines

We use the following methods to compare with the framework proposed in this paper:

- CKE [1] unifies collaborative filtering with structured knowledge, text knowledge and pictures information etc. in a framework for recommendation.
- DKN [4] treats word vectors, entity vectors and entity context vectors as multiple channels to fuse in the framework of CNN for click rate prediction.
- SHINE [16] designed a deep self-encoder to combine semantic network, social network and user profile network for celebrity recommendation.
- LibFM [2] is a widely used feature-based factorization framework for click-through rate prediction. In this experiment, user ID, item ID and corresponding entity embedding learned through TransR are used as input of LibFM.
- Wide&Deep [17] is a general deep framework for recommendation, which combines linear and non-linear channels. The embedding of users, items and entities are used as input for Wide&Deep.

4.3 Experiment Setup

In the experiments, $d = 16$ denotes the dimension of the embedding of items and KG, and $\eta = 0.008$ denotes the learning rate. Specific hyper-parameter settings are shown in Table 3. For fairness, all baseline methods have the same dimension settings as Table 3, while other baseline hyper-parameters are based on grid search. The ratio of training, evaluation and test set is 6:2:2. Each experiment was repeated 5 times and the average results is reported. Accuracy and area under curve (AUC) were used to evaluate the performance of click through rate (CTR) prediction.

Table 3. Hyper-parameter settings for the two datasets

Datasets	Hyper-parameter settings
MovieLens-1M	$d = 16$, $T = 2$, $\lambda_1 = 10^{-7}$, $\lambda_2 = 0.01$, $\eta = 0.008$
Book crossing	$d = 4$, $T = 3$, $\lambda_1 = 10^{-5}$, $\lambda_2 = 0.01$, $\eta = 0.001$

4.4 Performance Comparison

The results of all methods in click through rate prediction are shown in Table 4.

The proposed framework UPM achieves the best performance on two datasets with all evaluation metrics. CKE performs poorly than LibFM and Wide&Deep, since there is no text and visual information, and structural knowledge cannot characterize users completely. DKN performs worst among all methods in the two datasets, because film titles and book titles are usually short and contains limited information. SHINE performs better than DKN only, because we have no social and user profile networks. As two general recommendation algorithms, LibFM and Wide&Deep performs satisfactorily, which shows that LibFM and Wide&Deep can make full use of semantic information from KG.

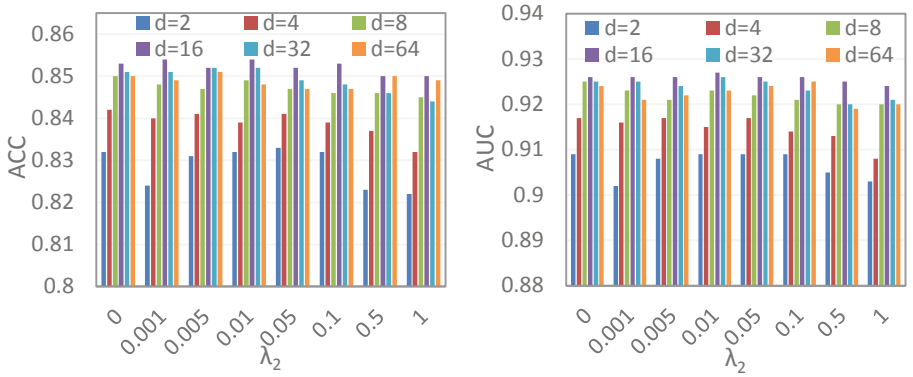
Table 4. The results of AUC and accuracy in click through rate prediction

Framework	MovieLens-1M		Book crossing	
	AUC	ACC	AUC	ACC
CKE	0.796	0.739	0.674	0.635
SHINE	0.778	0.732	0.668	0.631
DKN	0.655	0.589	0.621	0.598
LibFM	0.892	0.812	0.685	0.639
Wide&Deep	0.903	0.822	0.711	0.623
UPM	**0.928**	**0.855**	**0.740**	**0.695**

4.5 The Sensitivity of Hyper-parameters

The effect of dimension of embedding d and training weight of KG term λ_2 on AUC and ACC are shown in Fig. 2, which have similar trends on Book Crossing dataset. d range from 2 to 64, λ_2 range from 0 to 1, while keeping other parameters fixed.

With the increase of d, both AUC and ACC improves and becomes stable, because embedding with larger dimensions can encode more useful information, but when d is greater than 16, both AUC and ACC begin to drops because of possible overfitting. AUC and ACC performed best when $\lambda_2 = 0.01$. This is because when training weight of KG term is very small, it is not enough to provide effective regularization constraints, while a large training weight may mislead the objective function.



(a) ACC score w.r.t λ_2 and d (b) AUC score w.r.t λ_2 and d

Fig. 2. Parameter sensitivity of the proposes framework on MovieLens-1M.

In order to further explore the relationships between the performance of the framework and the maximal hop number K, we vary the maximal hop number K to see how AUC changes in UPM, the results as shown in Table 5.

Table 5. The results of AUC w.r.t. different hop numbers

Hop number K	1	2	3	4
MovieLens-1M	0.927	**0.928**	0.925	0.926
Book crossing	0.739	0.734	**0.740**	0.732

As shown in Table 5, the best performance is achieved when K is 2 or 3. This is because too small of an K can hardly explore inter-entity relatedness and dependency of long distance, while too large of an K brings much more noises than useful signals.

5 Conclusion

To address the challenges of traditional KG-aware recommendation methods, we innovatively combine feature-based methods, path-based methods and attention mechanism in KG-aware recommendation. Specifically, we proposed an end-to-end neural user preference modeling framework (UPM) for recommendation, which introduces KG into recommender systems effectively. UPM mine potential preferences of a user by propagating the user's interests in KG. The attention network is used to adaptively discriminate the importance of the preference features of user at different propagation stages for the final preference vector of user. Experimental results on two real-world datasets shows that the performance of the proposed framework is better

than other baseline methods, which further proves the effectiveness of the proposed method. In the future we will further explore how to represent entity-relation interactions efficiently and how to apply the framework to real-world scenarios.

Acknowledgments. This work was partially supported by the National Natural Science Foundation of China (Nos. U1501252, 61572146, U1711263), the Natural Science Foundation of Guangxi Province (No. 2016GXNSFDA380006, AC16380122), the Guangxi Innovation Driven Development Project (No. AA17202024), the Platform Construction Project of Guangxi Information Science Experiment Center (No. PT1601), the Basic Ability Promotion Project for Young and Middle-aged Teachers in Universities of Guangxi (2018KY0203) and the Innovation Project of GUET Graduate Education (Nos. 2019YCXS042).

References

1. Zhang, F., Yuan, N., Lian, D., et al.: Collaborative knowledge base embedding for recommender systems. In: 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–362. ACM, New York (2016)
2. Rendle, S.: Factorization machines with libFM. Trans. Intell. Syst. Technol. **3**(3), 57:1–57:22 (2012)
3. Yu, X., Ren, X., Sun, Y., et al.: Personalized entity recommendation: a heterogeneous information network approach. In: 7th International Conference on Web Search and Data Mining, pp. 283–292. ACM, New York (2014)
4. Wang, H., Zhang, F., Xie, X., et al.: DKN: deep knowledge-aware network for news recommendation. In: 27th International Conference on World Wide Web, pp. 1835–1844. ACM, New York (2018)
5. Huang, J., Zhao, W., Dou, H., et al.: Improving sequential recommendation with knowledge-enhanced memory networks. In: 41th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 505–514. ACM, New York (2018)
6. Bordes, A., Usunier, N., Garcia-D, A., et al.: Translating embeddings for frameworking multi-relational data. In: 26th International Conference on Neural Information Processing Systems, pp. 2787–2795. MIT Press, Cambridge (2013)
7. Sun, Y., Han, J.: Mining heterogeneous information networks: a structural analysis approach. ACM SIGKDD Expl. Newslett. **14**(2), 20–28 (2013)
8. Yu, X., Ren, X., Gu, Q., et al.: Collaborative filtering with entity similarity regularization in heterogeneous information networks. In: 23th International Joint Conference on Artificial Intelligence. Elsevier, Burling (2013)
9. Luo, C., Pang, W., Wang, Z., et al.: Hete-CF: social-based collaborative filtering recommendation using heterogeneous relations. In: 2014 IEEE International Conference on Data Mining. IEEE Computer Society, Washington (2015)
10. Shi, C., Zhang, Z., Luo, P., et al.: Semantic path based personalized recommendation on weighted heterogeneous information networks. In: 24th ACM International on Conference on Information and Knowledge Management, pp. 453–462. ACM, New York (2015)
11. Wang, Y., Xia, Y., Tang, S., et al.: Flickr group recommendation with auxiliary information in heterogeneous information networks. Multimedia Syst. **23**(6), 703–712 (2017)
12. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: 31th Conference on Neural Information Processing Systems, pp. 6000–6010. MIT Press, Cambridge (2017)
13. Yang, Z., Yang, D., Dyer, C., et al.: Hierarchical attention networks for document classification. In: NAACL-HLT 2016, pp. 1480–1489. ACL, Stroudsburg (2016)

14. Pei, W., Yang, J., Sun, Z., et al.: Interacting attention-gated recurrent networks for recommendation. In: 26th ACM Conference on Information and Knowledge Management, pp. 1459–1468. ACM, New York (2017)
15. Chen, J., Zhang, H., He, X., et al.: Attentive collaborative filtering: multimedia recommendation with item- and component-level attention. In: 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 335–344. ACM, New York (2017)
16. Wang, H., Zhang, F., Hou, M., et al.: Shine: Signed heterogeneous information network embedding for sentiment link prediction. In: 11th ACM International Conference on Web Search and Data Mining, pp. 592–600. ACM, New York (2018)
17. Cheng, H., Levent, K., Jeremiah, H., et al.: Wide & deep learning for recommender systems. In: 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10. ACM, New York (2016)