Knowledge Graph-Enhanced Recommendation with Box Embeddings

Qiuyu Liang[1], **Weihua Wang**[1,2,3]*, **Lei Lv**[1], **Feilong Bao**[1,2,3]
[1] College of Computer Science, Inner Mongolia University, Hohhot, China
[2] National and Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian, Hohhot, China
[3] Inner Mongolia Key Laboratory of Mongolian Information Processing Technology, Hohhot, China
wangwh@imu.edu.cn

Abstract

Knowledge graphs are used to alleviate the problems of data sparsity and cold starts in recommendation systems. However, most existing approaches ignore the hierarchical structure of the knowledge graph. In this paper, we propose a box embedding method for knowledge graph-enhanced recommendation system. Specifically, the box embedding represents not only the interaction between the user and the item, but also the head entity, the tail entity and the relation between them in the knowledge graph. Then the interaction between the item and the corresponding entity is calculated by the multi-task attention unit. Experimental results show that our method provides a large improvement over previous models in terms of Area Under Curve (AUC) and accuracy in publicly available recommendation datasets with three different domains.

1 Introduction

Recommendation systems have been proposed to solve the information explosion problem and to improve the user experience in various applications. Early recommendation systems were mainly based on collaborative filtering algorithms. These algorithms build user-item interaction matrices and perform similarity calculations to obtain final recommendations. However, collaborative filtering-based approaches encounter data sparsity and cold-start problems in practical applications.

To address these problems, some researchers have proposed recommendation methods that use Knowledge Graphs (KGs) as auxiliary information to improve the performance of recommendation tasks. Since KGs contain rich semantic relationships between entities corresponding to user-related items, the recommended items can take advantage of interactions with entities in the KG. Previous knowledge graph enhanced recommendation methods mainly used multiple modules to learn the representation of knowledge graph and recommendation task independently. For example, CKE (Zhang et al., 2016) used the TransR to encode the knowledge graph, while extracted the text and image features of the recommendation task through another self-encoding structure. Recently, several researchers have proposed methods for learning entities and items using multi-task learning. The core idea of such methods trained feature interaction layers jointly to connect the recommendation task and the knowledge graph-related task. For example, MKR (Wang et al., 2019) used cross units for training the knowledge graph embedding task and recommendation task. But the units were computed by matrix multiplication. This lead to the insufficient learning about the feature. In addition, few existing methods can model the semantic hierarchical structure relationship between recommendation data and the knowledge graph.

To address above shortcomings, we proposed **BoxMAKR**, a box embedding for knowledge graph-enhanced recommendation. Specifically, our model utilizes the attention mechanism to learn higher-order interaction features between items and entities more efficiently, and then embeds semantic hierarchical relationships into entities, users and items. We conduct experiments on three recommendation datasets, and the results show that BoxMAKR proposed in this paper achieves a large improvement compared with advanced models such as MKR.

* Corresponding author.

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 1172-1182, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

2 Related Work

Knowledge graph enhanced recommendation systems. The recommendation systems can provide users with interesting content based on their historical information. Early recommendation algorithms include collaborative filtering (Hu et al., 2008; Wang et al., 2020; Liu and Wang, 2022; Jiang et al., 2022), factorization machines (Rendle, 2010), and matrix factorization (Zhang et al., 2006). These algorithms basically used the historical interaction data to calculate the similarity between each user and item. Recently, some methods proposed more efficient mechanisms to learn the representation of users and items. However, these methods still suffered from data sparsity and cold start problems. For example, DNNRec (Kiran et al., 2020) integrated side information about users and items into a deep neural network to alleviate the cold start problem. DHMR (Khan et al., 2021) proposed higher-order non-linear latent feature to learn interactions from reviews and metadata information. But learning the lower-order feature interactions of users and items were separately.

Besides user reviews information, knowledge graphs also can be considered as an additional information source. This kind of task is called knowledge graph enhanced recommendation system. They are usually contain two subtasks: knowledge graph embedding and the recommendation task. To integrate the two tasks more effectively, more and more researchers focused on multi-task learning. For instance, MKR (Wang et al., 2019) designed an interaction unit to share the features between items and entities and trained jointly. Ripp-MKR (Wang et al., 2021) used the core idea of RippNet (Wang et al., 2018) to combine the knowledge graph with historical user interaction to represent user features based on MKR. TransMKR (Hu et al., 2022) improved the knowledge graph embedding module of MKR with TransR, which enhanced the representation capability of knowledge graph embedding.

Geometric embedding. The semantic hierarchical structure relationship of data explicitly modeled in the embedding space is very useful for solving natural language processing tasks. However, modelling the structure of word in the Euclidean space requires large embedding dimension and additional constraints. These constrain increased the computational complexity of the model. Therefore, some researchers proposed non-Euclidean embedding methods. For example, POE (Lai and Hockenmaier, 2017) was proposed to learn the sequential mapping of objects based on the semantic hierarchical structure instead of the traditional distance relationship. Its core idea is to model the objects using the partial order in the semantic hierarchical structure. Reference (Nickel and Kiela, 2017; Maximillian and Kiela, 2018; Liang et al., 2024) modeled objects' semantic hierarchical structure by similarity between the concepts in hyperbolic space. Box embeddings (Onoe et al., 2021; Patel and Sankar, 2020; Vilnis et al., 2018) used hyper-rectangles(box) to represent semantic hierarchical structure between objects as explicit conditional probabilities.

3 Problem Formula

In the recommendation task, there exists a user set $U = \{u_1, u_2, ..., u_m\}$, an item set $V = \{v_1, v_2, ..., v_n\}$, and a user-item interaction matrix $\mathcal{Y} \in \mathbb{R}^{m \times n}$, where $y_{uv} \in \{0, 1\}$. This user-item matrix represents whether an interaction exists between user u_m and item v_n, which is defined by users' feedbacks. In the matrix $\mathcal{Y}$, $y_{uv} = 1$ represents the user u engaged with item v, otherwise $y_{uv} = 0$. These feedbacks include implicit users' click, watching, browsing. Furthermore, a knowledge graph $\mathcal{G}$ can be composed of many triples (h, r, t). Where h denotes the head entity, t denotes the tail entity, and r denotes the relationship between them. In the knowledge graph each triple is associated with the items$v$. Also, the item $v \in V$ may be related to one or more entities in the knowledge graph.

Thus, the knowledge graph enhanced recommendation task can be formulated as follows: given a user set U, an item set V, a user-item interaction matrix $\mathcal{Y}$ and a knowledge graph $\mathcal{G}$, the model will predict the $\hat{y}_{uv}$ which means the user u is interested in item v.

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 1172-1182, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers

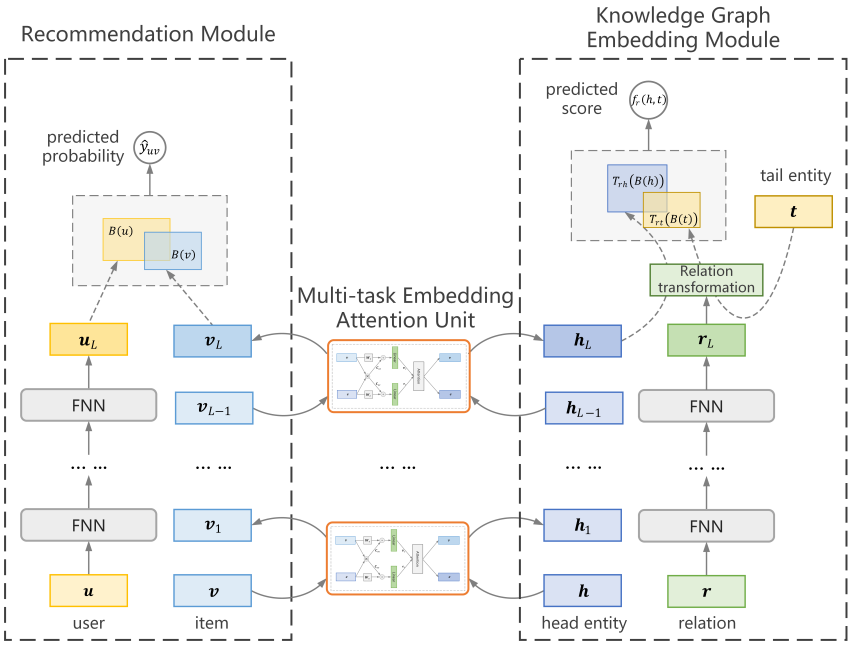

Figure 1: The architecture of BoxMAKR model.

4 Method

4.1 Model Overview

BoxMAKR consists of three parts: multi-task embedding attention units, recommendation module, and knowledge graph embedding module. The architecture of BoxMAKR is shown in Figure 1.

- The multi-task embedding attention units are implemented based on the attention mechanism for modeling the higher-order interaction features between items in the recommendation module and entities in the knowledge graph embedding module.
- The inputs of the recommendation module are users and items. Firstly, user and item embeddings are learned by multi-layer feedforward neural networks and multi-task embedding attention units. Then the user and item are learned by box embeddings. Finally, the prediction probabilities of users for items are output using condition probability of box embeddings.
- The knowledge graph embedding module has a similar network structure as the recommendation module. The relation and entity embeddings learned by multi-layer feedforward neural networks and multi-task embedding attention units. Then the head and tail entities are learned by box embeddings. Finally, the prediction scores of triples are output using condition probability of box embeddings.

4.2 Box Embeddings for Semantic Hierarchical Structure

Box embeddings represent each object as a hyper-rectangle $B(x)$ in the d dimensional embedding space, each hyper-rectangle is called a box. The box is represented in each dimension by a pair of vectors $\langle \boldsymbol{x}_m, \boldsymbol{x}_M \rangle$. In each dimension $i \in \{1, 2, ..., d\}$ there is $x_{m,i} < x_{M,i}$, where $x_{m,i}$ and $x_{M,i}$ represent the vector values of the bottom-left and upper-right positions of the box, respectively, i.e., the minimum and maximum values of the box in the embedding space. The box embeddings are shown in Figure 2. If the box embeddings space $\Omega_B \subseteq \mathbb{R}^d$ is normalized to volume 1, then the box can

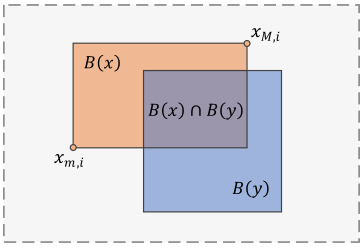

Figure 2: The box embeddings.

be interpreted as a probability distribution of binary random variables. In this case, the marginal probabilities of the objects are represented as the volume of the corresponding box:

$$V\left[B\left(x\right)\right] = \prod_{i=1}^{d}\left(x_{M,i} - x_{m,i}\right) \tag{1}$$

After that, in order to calculated the relation between different boxes, we introduce the joint and conditional probability. The joint probability of two boxes can be defined as the volume of intersection, since intersection process between two boxes will yield another box. The joint probability can be defined as:

$$V[B(x) \cap B(y)] = \prod_{i=1}^{d} max(min(x_{M,i}, y_{M,i}) - max(x_{m,i}, y_{m,i}), 0) \tag{2}$$

The conditional probabilities between two boxes can be further calculated with marginal probabilities and joint probabilities:

$$P\left(x|y\right) = \frac{V\left[B\left(x\right) \cap B\left(y\right)\right]}{V\left[B\left(y\right)\right]} \tag{3}$$

Optimization. The basic box embeddings model suffers from gradient loss during gradient descent-based optimization, which causes the model to fail to learn the embedding representation of objects. The Gumbel distribution is commonly used in probability theory. It models the distribution of the maximum or minimum values of the sample. According to Dasgupta et al. (Dasgupta et al., 2020), they demonstrated the the minimum and maximum values of the box can be represented as MaxGumbel distribution and the MinGumbel distribution with variance β, respectively, where β is the location parameters.

For the two boxes $B\left(x\right)$ and $B\left(y\right)$ represented in Ω_H using Gumbel box embeddings, the intersection box $B\left(z\right)$ can also be represented by the Gumbel distribution, where the two vector parameters z_m and z_M of $B\left(z\right)$ are calculated as:

$$\boldsymbol{z}_m = \beta \ln\left(e^{\frac{x_m}{\beta}} + e^{\frac{y_m}{\beta}}\right), \boldsymbol{z}_M = -\beta \ln\left(e^{-\frac{x_M}{\beta}} + e^{-\frac{y_M}{\beta}}\right) \tag{4}$$

The volume $V\left[B\left(x\right)\right]$ of the box $B(x)$ based on the Gumbel distribution representation can be approximated as a softplus function. The approximated box volume is calculated as follows.

$$V\left[B\left(x\right)\right] == \prod_{i=1}^{d} softplus\left(\frac{x_{M,i} - x_{m,i}}{\beta} - 2\gamma\right) \tag{5}$$

4.3 Multi-task Embedding Attention Unit

Items in the recommendation system usually correspond to many entities. The knowledge graph enhanced recommendation method will establish interaction relations between the items and the corresponding entities. The interaction relations will be established by interaction units. The multi-task embedding attention unit introduces the attention mechanism in the interaction process between items and entities to effectively share the high order interaction feature. The structure of this unit is shown in Figure 3.

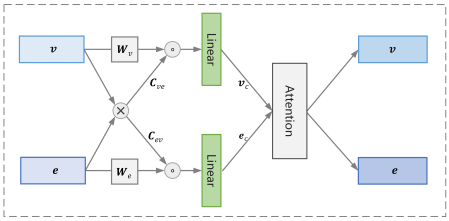

Figure 3: Multi-task embedding attention unit.

The semantic hierarchical structure of word can be modeled by conditional probabilities of box embeddings. For example, there is an edge from "Dog" to "Mammal" in the WordNet. We can convert this edge into the conditional probability of a binary random variable $P\left(Mammal|Dog\right) = 1$, then get the box embeddings of each word and their relationship, as shown in Figure 4.

This unit input item embeddings $\boldsymbol{v} \in \mathbb{R}^d$ and entity embeddings $\boldsymbol{e} \in \mathbb{R}^d$, performs a matrix product operation to construct the feature matrix $\boldsymbol{C}_{ve}$ and $\boldsymbol{C}_{ev}$ between the items and entities:

$$\boldsymbol{C}_{ve} = \boldsymbol{v}^\top \otimes \boldsymbol{e},\ \boldsymbol{C}_{ev} = \boldsymbol{e}^\top \otimes \boldsymbol{v} \tag{6}$$

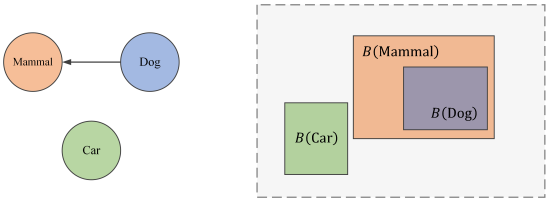

Figure 4: The box embeddings model semantic hierarchical structure.

where $\boldsymbol{C}_{ve}$, $\boldsymbol{C}_{ev} \in \mathbb{R}^{d\times d}$ can explicitly model all interaction features between items and entities. However, the matrix product calculation made features between item and entity over fused, which result in reducing feature weight of the item in the recommendation task. Thus, the feature representations of items and entities are further enriched by four parameter matrices $\boldsymbol{W}_v$, $\boldsymbol{W}_{ve}$, $\boldsymbol{W}_e$, $\boldsymbol{W}_{ev}$. Then preliminary interative feature $\boldsymbol{v}_c$ and $\boldsymbol{e}_c \in \mathbb{R}^d$ of item embeddings and entity embeddings can by obtained by a linear layer:

$$\boldsymbol{v}_c = l\left(\boldsymbol{W}_v\left(\boldsymbol{v}^\top \boldsymbol{v}\right) \circ \boldsymbol{W}_{ve}\boldsymbol{C}_{ve}\right),\ \boldsymbol{e}_c = l\left(\boldsymbol{W}_e\left(\boldsymbol{e}^\top \boldsymbol{e}\right) \circ \boldsymbol{W}_{ev}\boldsymbol{C}_{ev}\right) \tag{7}$$

Where $\circ$ denotes Hadamard product operation, $\boldsymbol{W}_v$, $\boldsymbol{W}_e \in \mathbb{R}^{d\times d}$ are the parameter matrix of item embeddings $\boldsymbol{v}$ and entity embeddings $\boldsymbol{e}$, $\boldsymbol{W}_{ve}$, $\boldsymbol{W}_{ev} \in \mathbb{R}^{d\times d}$ are the parameter matrix of feature matrix$\boldsymbol{C}_{ve}$ and $\boldsymbol{C}_{ev}$, and $l(\cdot)$ denote linear neural network layer.

To further learn the feature weights of item and entity embeddings, the attention mechanism is to model the higher-order interaction features based on the preliminary interaction feature representations. Since attention calculations are involved between the two tasks, it is necessary to define the query $\boldsymbol{q}$ and the key $\boldsymbol{k}$ for item embeddings and entity embeddings, respectively:

$$\boldsymbol{q}_v = \boldsymbol{W}_q\boldsymbol{v}_c\,, \boldsymbol{k}_v = \boldsymbol{W}_k\boldsymbol{v}_c\,, \boldsymbol{q}_e = \boldsymbol{W}_q\boldsymbol{e}_c\,, \boldsymbol{k}_e = \boldsymbol{W}_k\boldsymbol{e}_c \tag{8}$$

Where $\boldsymbol{W}_q$, $\boldsymbol{W}_k \in \mathbb{R}^{d\times d}$ are the parameters of the query and key. Then the correlation scores $\boldsymbol{s}$ between them are calculated by scaling the dot product. Finally, the correlation scores $\boldsymbol{s}$ are normalized using the softmax function to obtain the attention weight distribution α:

$$\alpha_{ve} = softmax\left(\boldsymbol{s}_{ve}\right) = \frac{exp\left(\boldsymbol{s}_{ve}\right)}{\sum_{v,e} exp\left(\boldsymbol{s}_{ve}\right)}\,, \boldsymbol{s}_{ve} = \frac{\boldsymbol{q}_v{}^\top \boldsymbol{k}_e}{\sqrt{d}} \tag{9}$$

The final item $\boldsymbol{v}$ and entity embedding $\boldsymbol{e}$ are obtained from the following equations:

$$\boldsymbol{v} = \alpha_{ve}\boldsymbol{v}_c, \boldsymbol{e} = \alpha_{ev}\boldsymbol{e}_c \tag{10}$$

where α_{ev} is calculated in the same way with α_{ve}. After learning from multi-task embedding attention units, the item embedding and entity embedding can effectively capture the higher-order interaction features between items and entities. After the interaction, the learned item embedding and entity embedding are more suitable for the process of their individual task.

4.4 Recommendation Module

The recommendation module is used to learn the box embeddings representation of users and items. The module is divided into two parts: the low-level encoding and the box embeddings. Its inputs are the initial embeddings $\boldsymbol{u}$ of users u and $\boldsymbol{v}$ of the items v. The user embeddings $\boldsymbol{u}$ are first encoded by the feedforward neural network(FNN) with layer number of L . The encoded user embeddings are represented as $\boldsymbol{u}_L$. The item embeddings $\boldsymbol{v}$ are encoded with the corresponding entity embeddings $\boldsymbol{e}$ in the knowledge graph by L-layer multi-task embedding attention unit (MEAU), and the encoded item embeddings are represented as $\boldsymbol{v}_L$.

$$\boldsymbol{u}_L = FNN\left(\cdots FNN\left(\boldsymbol{u}\right)\right) \tag{11}$$

$$\boldsymbol{v}_L = \mathbb{E}_{v\sim e}\left(MEAU\left(\cdots MEAU\left(\boldsymbol{v},\boldsymbol{e}\right)[\boldsymbol{v}]\right)\right) \tag{12}$$

where $\mathbb{E}_{v\sim e}$ represents the mapping process between item v and entity e. After the low-level encoding, the user and item need to be modeled as box embeddings. For the box, its geometric transformation is mainly translation and scale. Therefore, we further convert the user and item embedding into two vectors. They are the center vector $\boldsymbol{c} \in \mathbb{R}^d$ and the bias vector $\boldsymbol{o} \in \mathbb{R}^d$. The center vector controls the position of the box, while the bias vector controlls the size of the box. Therefore, the user box $B(u)$ and the item box $B(v)$ are represented as:

$$B\left(u\right) = [(\boldsymbol{u}_m, \boldsymbol{u}_M)] = [\sigma\left(\boldsymbol{c}_u - softplus\left(\boldsymbol{o}_u\right)\right), \sigma\left(\boldsymbol{c}_u + softplus\left(\boldsymbol{o}_u\right)\right)] \tag{13}$$

$$B\left(v\right) = [(\boldsymbol{v}_m, \boldsymbol{v}_M)] = [\sigma\left(\boldsymbol{c}_v - softplus\left(\boldsymbol{o}_v\right)\right), \sigma\left(\boldsymbol{c}_v + softplus\left(\boldsymbol{o}_v\right)\right)] \tag{14}$$

Finally, based on the conditional probability of the box, the probability of user u is interested in the item v is predicted as $\hat{y}_{uv}$:

$$\hat{y}_{uv} = P\left(\ B\left(v\right)|B\left(u\right)\right) = \frac{V\left[B\left(v\right) \cap B\left(u\right)\right]}{V\left(B\left(u\right)\right)} \tag{15}$$

4.5 Knowledge Graph Embedding Module

The knowledge graph embedding module is used to learn the box embeddings representation of entities and relations in the knowledge graph embedding task. Its network structure is similar to the recommendation module. Its inputs are the initial feature embeddings $\boldsymbol{h}$ of head entities h, $\boldsymbol{r}$ of relations r, and $\boldsymbol{t}$ of tail entities t in the knowledge graph triples (h, r, t). The relation embedding $\boldsymbol{r}$ is first encoded using an L-layer FNN, and the encoded relation embeddings are represented as $\boldsymbol{r}_L$. The head entity embeddings $\boldsymbol{h}$ are encoded with the corresponding item embeddings $\boldsymbol{v}$ in the recommendation by the L-layer MEAU, and the encoded head entity embeddings are represented as $\boldsymbol{h}_L$.

$$\boldsymbol{r}_L = FNN\left(\cdots FNN\left(\boldsymbol{r}\right)\right) \tag{16}$$

$$\boldsymbol{h}_L = \mathbb{E}_{h\sim v}\left(MEAU\left(\cdots MEAU\left(\boldsymbol{h},\boldsymbol{v}\right)[\boldsymbol{h}]\right)\right) \tag{17}$$

where $\mathbb{E}_{h\sim v}$ represents the mapping process between the head entity h and the item v. After completing the lower-level encoding of relations and entities, the box embeddings of relations and entities need to be modeled. Where the box representation of entities is similar to the box representation of users and items in the recommendation module. But in the knowledge graph, multiple types of relationships often exist between entities, the embedding of the same entity under different relationships should be different. Therefore, we further represent the relation r between entities as two transformation vectors, which are the center transformation vector $\boldsymbol{c}^r \in \mathbb{R}^d$ and the bias transformation vector $\boldsymbol{o}^r \in \mathbb{R}^d$ controlling the position and size of the entity box, respectively, and the transformation vectors of different relations can transform the position and size of entity box in different ways. For the head entity box $B(h)$ and the tail entity box $B(t)$, their center vectors and bias vectors under the relation r are represented as follows:

$$\boldsymbol{c}_h\left(r\right) = \boldsymbol{c}_h + \boldsymbol{c}_h^r\ ,\ \boldsymbol{o}_h\left(r\right) = \boldsymbol{o}_h \circ \boldsymbol{o}_h^r \tag{18}$$

$$\boldsymbol{c}_t\left(r\right) = \boldsymbol{c}_t + \boldsymbol{c}_t^r\ ,\ \boldsymbol{o}_t\left(r\right) = \boldsymbol{o}_t \circ \boldsymbol{o}_t^r \tag{19}$$

where $\circ$ is the Hadamard product, $\boldsymbol{c}_h\left(r\right),\ \boldsymbol{o}_h\left(r\right)$ and $\boldsymbol{c}_t\left(r\right),\ \boldsymbol{o}_t\left(r\right)$ represent the center vector and bias vector of the head entity box $B(h)$ and the tail entity box $B(t)$ after the transformation of the relation r, respectively. Based on the above center vectors and bias vectors, the box embeddings of $B(h)$ and $B(t)$ under the relation r are given as $T_{rh}\left(B\left(h\right)\right)$ and $T_{rt}\left(B\left(t\right)\right)$.

$$T_{rh}\left(B\left(h\right)\right) = [\sigma\left(\boldsymbol{c}_h\left(r\right) - softplus\left(\boldsymbol{o}_h\left(r\right)\right)\right), \sigma\left(\boldsymbol{c}_h\left(r\right) + softplus\left(\boldsymbol{o}_h\left(r\right)\right)\right)\ ] \tag{20}$$

$$T_{rt}\left(B\left(t\right)\right) = [\sigma\left(\boldsymbol{c}_t\left(r\right) - softplus\left(\boldsymbol{o}_t\left(r\right)\right)\right), \sigma\left(\boldsymbol{c}_t\left(r\right) + softplus\left(\boldsymbol{o}_t\left(r\right)\right)\right)\ ] \tag{21}$$

Finally, based on the geometric properties of the box, the score $f_r\left(h,t\right)$ of the triple (h, r, t) is predicted:

$$f_r\left(h,t\right) = \frac{V\left[T_{rh}\left(B\left(h\right)\right) \cap T_{rt}\left(B\left(t\right)\right)\right]}{V\left[T_{rt}\left(B\left(t\right)\right)\right]} \tag{22}$$

4.6 Learning

The BoxMAKR model is jointly trained by the recommendation task and the knowledge graph embedding task. The learning objective $\mathcal{L}$ includes three parts: the recommendation task loss function $\mathcal{L}_{REC}$, the knowledge graph embedding task loss function $\mathcal{L}_{KGC}$, and the regular term $\mathcal{L}_2$. The equation is as follows:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{REC} + \mathcal{L}_{KGC} + \mathcal{L}_2 \\ &= \sum_{u \in U, v \in V} g\left(\hat{y}_{uv}, y_{uv}\right) - \lambda_1 \left(\sum_{(h,r,t) \in \mathcal{G}} f_r\left(h,t\right) - \sum_{(h',r,t') \notin S} f_r\left(h',t'\right) \right) + \lambda_2 \left\| \boldsymbol{W} \right\|_2^2 \end{aligned} \tag{23}$$

where $g\left(\cdot\right)$ is the cross-entropy loss function, the learning objective of the recommendation task is to make the prediction probability $\hat{y}_{uv}$ close to the real probability y_{uv}. In Equation 23, (h', r, t') is the negative sample of the triples. The learning objective of the knowledge graph embedding task is to improve the prediction score $f_r\left(h,t\right)$, while reducing the prediction score $f_r\left(h',t'\right)$ of the negative samples. Finally, the L_2 regular term is to avoid model overfitting. $lambda_1$ and $lambda_2$ are the hyperparameters used to balance the knowledge graph embedding task and the regular term.

Datasets	Users	Items	Interactions	KG triples	Sparsity
MovieLens 1M	6036	2347	753772	20195	94.68%
Book-Crossing	17860	14910	139746	19793	99.95%
Last.FM	1872	3846	42346	15518	99.41%

Table 1: Basic statistics for the three datasets.

5 Experiments

5.1 Datasets

In our experiments, we used three publicly available datasets containing explicit user feedback. MovieLens-1M is a movie recommendation benchmark dataset which described the users' preferences for movies. It includes 6036 users for 2347 movies with 753,772 ratings. Its data sparsity is 94.68%. Book-Crossing is crawled from the Book-Crossing community, which described users' grades for book. In total, it includes 139,746 ratings from 17,860 users on 14,910 book items. Its data sparsity is 99.95%. Last.FM is from the Last.FM online music system, which described the listen records of users and music. It includes a total number of 42,346 listening counts, which produced by 1,872 users about 3,846 songs. Its data sparsity is 99.41%.

We divided the users feedback into positive and negative. The positive feedback of users to items is denoted as "1", while negative feedback is denoted as "0". In MovieLens-1M, user ratings of 4 and 5 were selected as positive feedback. However, the other two datasets have a large data sparsity. We use all user interactions on items in the dataset as positive feedback. After building the positive feedback, we randomly sampled the same number of negative samples for each user with items that are not interaction. Then, we further used the Microsoft Satori service to build the knowledge graph on the above datasets. The statistics of the three datasets are shown in Table 1.

5.2 Baselines

To verify the validity of our model, we compare it with six classical baseline models. KTUP (Cao et al., 2019) jointly learned the recommendation task and the knowledge graph complementation task. The core idea of this method is introducing preference induction to model users' preference features for items. The bridge between the two tasks is the alignment of the recommended items with the corresponding entities. The user preferences are related to the relations between the entities. RippleNet (Wang et al., 2018) introduced the concept of preference propagation to optimize user features by sampling from the knowledge graph in an aggregated way. Eventually, user preferences from historical interaction information are propagated over paths in knowledge graph to realize the joint training of knowledge graph and

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 1172-1182, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers

Datasets	Dim	Layer	Lr1	Lr2	Lr3
MovieLens-1M	32	4	3e-3	1e-3	0.5
Book-Crossing	16	3	5e-4	2e-5	0.1
Last.FM	16	2	3e-3	2e-4	0.1

Table 2: Hyperparameters settings for the three datasets.

recommendation. MKR (Wang et al., 2019) is an unified framework for knowledge graph enhanced recommendation with multi-task learning. The framework includes recommendation and knowledge graph embedding module, which are connected by cross&compress unit. During training, they used alternative learning strategy to train recommendation and knowledge graph embedding task. Ripp-MKR (Wang et al., 2021) imporved MKR model with RippleNet. In their work, Ripp-MKR combined the knowledge graph with the user history interaction information to represent the user's features. TransMKR (Hu et al., 2022) improved the knowledge graph embedding module of MKR by using the TransR algorithm. TransR can enhance the representation capability of embedding knowledge graph.

5.3 Experiments setup

In the experiments, the hyperparameters of the baseline models are initialized using the data from the original work, and the optimal parameter is selected by fine-tuning in our experiments. The hyperparameter settings of BoxMAKR model on different datasets are shown in the table 2, where dim represents the embedding dimension, $layer$ represents the number of network layers, lr_1 and lr_2 are the learning rates of the recommendation task and the knowledge graph embedding task, respectively. $lambda_1$ is the weight parameter of the recommendation task. The other hyperparameters are fixed in all datasets. The weight λ_2 of the regular term is 1e-6. The scale parameter β of the Gumbel distribution in the box embeddings is 0.01. The training interval t of the knowledge graph embedding task is 3.

The datasets are divided into the training, validation and test set with the ratio of 6:2:2. Each experiment is conducted three times, and the average results are reported to validate the robustness of the model. We compared all methods in two typical task. Click-through rate (CTR) prediction. The trained model is applied to the test set to get click-through rate prediction scores, and the performance is evaluated using two evaluation metrics, AUC and ACC. TOP-K recommendation. The trained model is applied to the test set, and the K recommendation items with the highest predicted click probability are selected for each user. The performance is evaluated using two evaluation metrics, $Recall@K$ and $F1@K$.

Model	MovieLens-1M		Book-Crossing		Last.FM	
	AUC	ACC	AUC	ACC	AUC	ACC
KTUP	0.822	0.761	0.689	0.643	0.735	0.667
RippleNet	0.920	0.842	0.729	0.662	0.768	0.691
MKR	0.917	0.843	0.734	0.704	0.799	0.752
Ripp-MKR	0.922	0.845	0.740	0.712	0.799	**0.756**
TransMKR	0.916	0.838	0.737	0.706	0.797	0.753
BoxMAKR(ours)	**0.926**	**0.855**	**0.749**	**0.712**	**0.804**	0.752

Table 3: The results of AUC and Accuracy in CTR prediction.

6 Results

6.1 Overall results

The experimental results of the CTR prediction and TOP-K recommendation are showed in table 3. The comparison between different models on Top-K in three datasets are shown in Figure 5, Figure 6 and Figure 7, respectively.

From Table 3, we observed that KTUP performs the worst in the three datasets, especially in the dense MovieLens-1M where the performance is much lower than the other models. This demonstrated that

the preference induction introduced by KTUP cannot build effective interactions between users and the knowledge graph which is not suitable for complex recommendation scenarios.

RippleNet has better performance in the MovieLens-1M because the user-item interaction information is dense and preference propagation can more accurately capture user interests. However, RippleNet does not perform as well as MKR in the Book-Crossing and Last.FM when the data is sparse.

In Table 3, Ripp-MKR and TransMKR are used the same framework of MKR. Ripp-MKR combined the advantages of RippleNet and MKR, has better performance on all datasets. It shows that introducing user preferences under a multi-task learning strategy can effectively integrate the knowledge graph embedding and recommendation task. TransMKR has no significant improvement in overall performance compared with MKR. This is because TransMKR only improves the embedding method of the knowledge graph embedding module without adapting the recommendation module and interaction unit. It leads to the recommendation task not being able to effectively obtain the feature representation in the knowledge graph embedding task.

Compared with other baselines, the BoxMAKR improved the AUC scores by 0.9%, 1.5%, and 0.5% on the three datasets, respectively. Meanwhile, the ACC scores improve by 1.2% and 0.8% on the MovieLens-1M and Book-Crossing, respectively. But not on the Last.FM, which is probably because the amount of data is too small to fully train the network model. BoxMAKR outperforms the optimal result on the three datasets.

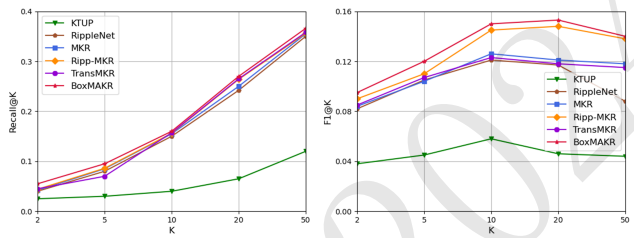

Figure 5: TOP-K recommendation results on the MovieLens-1M.

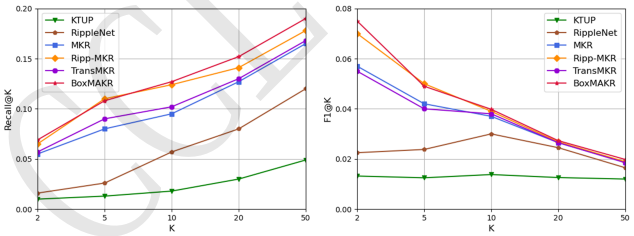

Figure 6: TOP-K recommendation results on the Book-Crossing.

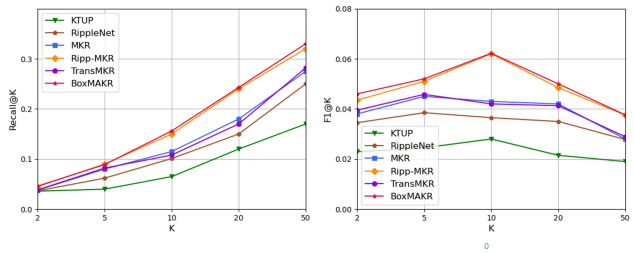

Figure 7: TOP-K recommendation results on the Last.FM.

6.2 Ablation

We further demonstrated the effectiveness of box embeddings and multi-task embedding attention unit through ablation experiments.

In Table 4, BoxMAKR(w/o Att) and BoxMAKR(w/o box) are models of BoxMAKR using MKR's interaction unit and prediction network, respectively. Compared with BoxMAKR, BoxMAKR(w/o Att) and BoxMAKR(w/o box) decreased the AUC metrics by 0.5% and 0.7% on average and the ACC metrics by 0.6% and 0.8% on average in the three datasets. The results of the ablation experiments show that introducing attention mechanisms in the interaction units can share the higher-order interaction features between items and entities more effectively. It can also be seen that box embeddings play a very important role in BoxMAKR because box embeddings can provide more semantic information for recommendation tasks and knowledge graph embedding tasks by modeling the semantic hierarchical structural relationships between data.

Model	MovieLens-1M		Book-Crossing		Last.FM	
	AUC	ACC	AUC	ACC	AUC	ACC
BoxMAKR(w/o Att)	0.922	0.846	0.745	0.707	0.799	0.748
BoxMAKR(w/o Box)	0.920	0.842	0.742	0.704	0.796	0.749
BoxMAKR	**0.926**	**0.855**	**0.749**	**0.712**	**0.804**	**0.752**

Table 4: The ablation results of AUC and Accuracy in CTR prediction.

7 Conclusions

This paper proposed a box embedding method for knowledge graph enhanced recommendation system. Based on the MKR model, we use box embeddings to represent users, items and entities. This embeddings allow us to model the semantic hierarchical structure relationships between those objects. Further, the multi-task attention method learned the higher-order interactions between items and corresponding entities. We conduct extensive experiments in three recommendation datasets. The results showed that our method is better than the original model MKR, and have great advantages over other baseline models.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Nos.62066033, 61966025); Inner Mongolia Natural Science Foundation (Nos.2024MS06013, 2022JQ05); Inner Mongolia Autonomous Region Science and Technology Programme Project (Nos.2023YFSW0001, 2022YFDZ0059); Collaborative Innovation Project of Universities and Institutes in Hohhot City. We also thank all the anonymous reviewers for their insightful comments.

References

Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The World Wide Web Conference*, pages 151–161.

Shib Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Li, and Andrew McCallum. 2020. Improving local identifiability in probabilistic box embeddings. *Advances in Neural Information Processing Systems*, 33:182–192.

Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*, pages 263–272. IEEE.

Bojing Hu, Yaqin Ye, Yingqiang Zhong, Jiao Pan, and Maosheng Hu. 2022. TransMKR: Translation-based knowledge graph enhanced multi-task point-of-interest recommendation. *Neurocomputing*, 474:107–114.

Xueyong Jiang, Baisong Liu, Jiangchen Qin, Yunchong Zhang, and Jiangbo Qian. 2022. Fedncf: Federated neural collaborative filtering for privacy-preserving recommender system. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Zahid Younas Khan, Zhendong Niu, Ally S Nyamawe, and Ijaz ul Haq. 2021. A deep hybrid model for recommendation by jointly leveraging ratings, reviews and metadata information. *Engineering Applications of Artificial Intelligence*, 97:104066–104066.

R Kiran, Pradeep Kumar, and Bharat Bhasker. 2020. Dnnrec: A novel deep learning based hybrid recommender system. *Expert Systems with Applications*, 144:113054–113054.

Alice Lai and Julia Hockenmaier. 2017. Learning to predict denotational probabilities for modeling entailment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 721–730.

Qiuyu Liang, Weihua Wang, Feilong Bao, and Guanglai Gao. 2024. L^2GC:lorentzian linear graph convolutional networks for node classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9988–9998, Torino, Italia, May. ELRA and ICCL.

Xinyu Liu and Zengmao Wang. 2022. Cfda: Collaborative filtering with dual autoencoder for recommender system. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

Nickel Maximillian and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pages 3779–3788. PMLR.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *Advances in Neural Information Processing Systems*, 30.

Yasumasa Onoe, Michael Boratko, Andrew McCallum, and Greg Durrett. 2021. Modeling fine-grained entity types with box embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Dhruvesh Patel and Shib Sankar. 2020. Representing joint hierarchies with box embeddings. *Automated Knowledge Base Construction*.

Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 263–272.

Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 417–426.

Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *The World Wide Web Conference*, pages 2000–2010.

Chao Wang, Hengshu Zhu, Chen Zhu, Chuan Qin, and Hui Xiong. 2020. Setrank: A setwise bayesian approach for collaborative ranking from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6127–6136.

Yuequn Wang, Liyan Dong, Yongli Li, and Hao Zhang. 2021. Multitask feature learning approach for knowledge graph enhanced recommendations with ripplenet. *PLoS ONE*, 16.

Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. 2006. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 549–553. SIAM.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 353–362.