ELEKTROTEHNIŠKI VESTNIK 88(3): 85–97, 2021
ORIGINAL SCIENTIFIC PAPER

Multi-Objective Optimization with Multi-View Attention for a Knowledge-Graph Enhanced-Recommendation System

Yingjie Tian[1,3,4], **Kunlong Bai**[2,3,4], **Dalian Liu**[5,6†]

[1]*School of Economics and Management, University of Chinese Academy of Sciences,*
No. 80 of Zhongguancun East Street, Haidian District, Beijing 100190, China
[2]*School of Computer Science & Technology, University of Chinese Academy of Sciences,*
Beijing 100049, China
[3]*Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences,*
No. 80 of Zhongguancun East Street, Haidian District, Beijing 100190,China
[4]*Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences,*
No. 80 of Zhongguancun East Street, Haidian District, Beijing 100190, China
[5]*Department of Basic Course Teaching, Beijing Union University, Beijing 100101, China*
[6]*Institute of Fundamental and Interdisciplinary Sciences, Beijing Union University, Beijing 100101, China*

[†] *E-mail: tyj@ucas.ac.cn*

Abstract. The recommendation system (RS) has become an essential component of various e-commerce platforms. Its purpose is to predict the next interaction item based on the user's information and his/her interaction history. As the existing methods have not flexibly used the external knowledge, the potential knowledge-level correlation between products can not be fully utilized. Therefore, the results recommended for users are limited to simple models and can not be reasonably extended. We propose a novel model which takes the multi-task learning with a multi-view attention for recommendation system (RS) and knowledge graph embedding (KGE) tasks simultaneously, motivated by the following. First, RS and KGE both involve embedding learning problems, one of which is at the item level and the other at the knowledge level. Second, these two tasks can help each other to improve. In other words, RS can incorporate the external knowledge from the knowledge graph, and KGE can be enhanced by learning a contextual information from RS. To improve the interactive process between these two tasks, we propose a novel multi-task learning scheme, which ingeniously employs a multi-view attention learned from various views on interacting these tasks more effectively and learning the embedding representation more comprehensively. The experiments conducted on several standard datasets demonstrate the effectiveness of the proposed method and an improvement in RS and KGE.

Keywords: neural networks, multi-task learning, multi-view attention, recommendation system (RS), knowledge graph (KG), knowledge graph embedding (KGE)

Večkriterijska optimizacija grafikona znanja v sistemih za priporočila

Sistem za priporočila (RS) je postal bistveni sestavni del različnih platform za e-poslovanje. Njegov namen je predvideti naslednji korak interakcije na osnovi uporabnikovih informacij. Ker obstoječe metode ne uporabljajo dodatnega znanja in informacij, potencialnih korelacij med predmeti ni mogoče v celoti upoštevati. V prispevku predlagamo nov model, ki upošteva večopravilno učenje za simultano vključevanje nalog v sistem za priporočila. Učinkovitost predlaganih metod smo eksperimentalno preverili na standardnih naborih podatkov.

Received 26 January 2021
Accepted 23 March 2021

1 INTRODUCTION

RS plays a vital role in improving users' online service experience. It analyzes the user's behaviour, finds the their personalized needs, and personalizes some products to the corresponding users, helping them find the products they want but are difficult to find. Although a well-performing RS can significantly reduce the users' time and energy in looking for things of interest, sometimes there are some unexpected recommendation items that cause confusion to users. To building an effective RS, there are many powerful neural network-based recommendation algorithms [1, 2, 3, 4, 5, 6]. By encoding historical interaction records as hidden vectors, these methods can capture dynamic users' preferences over time and predict the probability of the next product. However, most algorithms have data sparsity problems, such as a lack of a detailed information about the network of products or the social information about users, and it is difficult for these algorithms to give an apparent recommendation reason.

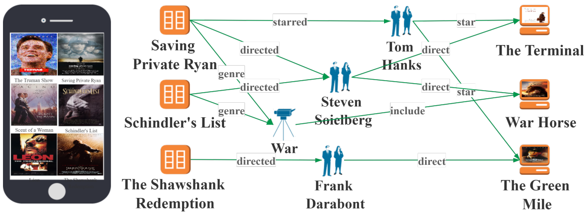


Figure 1. Diagram of KG-enhanced movie RS. KG provides a wealth of links among facts and entities, improving the accuracy, diversity, and interpretability of recommended result.

In other studies, some recommendation algorithms introduce the external knowledge into the recommendation process [7, 8, 9, 10, 11]. The external information knowledge should surely be rich and flexible and can represent contextual details in various fields. After an in-depth study, we find that structured knowledge graphs show a great potential in providing information about recommended items and offer promising solutions to improve the RS's accuracy and interpretability [12, 13, 14, 15]. KG is a heterogeneous structure that stores the world knowledge in a form of a machine-readable graph, in which nodes represent entities and edges define relationships between entities. We can import the recommended item attributes into KG as an auxiliary information. In general, researchers extract these attributes from open KG databases and mark them as an item knowledge to reduce data sparsity. KG is a trendy term in many practical applications [16, 17, 18, 19, 20] because it is very convenient and useful to construct a background knowledge in a graphic form. To illustrate the significance of incorporating the KG into recommendation modelling, we use a toy example, which is extracted from a real movie dataset. In Figure 1, the movies, actors, and directors are entities and relations are edges. It shows that a user has watched some movies on his/her mobile phone, among which the Schindler's List and The Terminal are directed by the same director (i.e., Steven Spielberg); The Schindler's List and the War Horse belong to the same genre (i.e., war). Therefore, the multiple-relationships of the movie knowledge can be effectively summarized through a movie-knowledge graph. Through the entity representation based on the attribute correlation, the potential relationship between the target user and the candidate item can be discovered accurately to improve the recommendation performance. In order to fully obtain a compact representation of a KG information, researchers use knowledge embedding methods to map entities and relationships into low-dimensional vectors formed knowledge graph embedding (KGE) [21, 22, 23]. KGE transforms entities and relationships into a continuous vector space, while retaining the KG's original structure. The learned entity and relationship embedding vectors are then used in various tasks, such as RS.

Based on the above motivations, we propose a novel multi-task learning framework to provide an accurate and interpretable recommendation. The basic idea includes two aspects: 1) using the KG facts as an auxiliary information to enhance the modelling of the user-item interaction; 2) learning the representation of entities and relationships in a low-dimensional continuous vector space on the basis of an improved user-item modelling. The framework is composed of a RS module and a KGE module. The KGE module learns a general and compact representation for entities and relationships, which is easy to use and integrate for a subsequent use. The RS module combines the existing recommendation model with the learned knowledge fact vectors to better deal with the cold start problems and give interpretable recommendations. In KGE module, we exploit KG to summarize the available facts from the product association and provide convincing recommendations based on the facts of RS module. For example, we can learn about users' preferences for actors through relevant entities and relationships in the movie-knowledge facts. That is to say, if some users favour Tom Hanks, we can recommend other movies that they have not watched before, such as The Green Mile, in which Tom Hanks is one of the actors. This discovery motivates us to design a multi-task learning paradigm leveraging the ready-made knowledge facts. The existing multi-task learning schemes mostly divide the neural network layers into a task-specific and a task sharing layer [24, 25]. The shared layer is shared among all tasks, while the task-specific layer is independent from other tasks. To model an interaction between different tasks in a more fine-grained way, we aggregate the information of task-specific layers through an attention mechanism and then learn more comprehensive item representations in shared layers, termed a multi-view attentive unit. Specifically, a shared unit uses a multi-view attention mechanism to enhance the item representational learning by combining item-view information from a RS module and a knowledge-view information from a KGE module if they refer to the same thing. Each neuron from different modules combines different inputs (user-item interaction data and knowledge facts) and applies a nonlinear activation function in the multi-view attentive unit. By using a multi-view attention scheme, a item-level and knowledge-level attention information can be shared and transferred between different tasks effectively.

In summary, our contributions in this paper are as follows:

1) By building a new RS model, we extend the traditional collaborative filtering to learning over a heterogenous knowledge-based embedding, making it possible to capture the users' preferences comprehensively.
2) KG is incorporated into our model through a multi-task paradigm, which takes KGE learning as an auxiliary task of the model.
3) By proposing a multi-view attention mechanism to link different tasks, a critical information of the task-specific layers is integrated into the shared layers, enabling the model to learn the item-level and knowledge-level representation interactively.
4) By providing a deep insights on the rationale of our model design mechanism, and three real-world RS datasets are linked to the facts in Microsoft Satori * for experiments. Our model's superiority over the state-of-the-art models is demonstrated.

2 Related Work

In the early RS research, researchers focused on recommending similar users or items to target users, such as collaborative filters (CF) [26], factorization machines [27, 28], and matrix factorization [29]. The standard practice of these methods is to utilize the users' history interaction and extract the representation of users and items to calculate their similarity. With the emergence of the neural network [30, 31], an increasing number of deep-learning methods [32, 33, 34, 35, 36, 37, 38] extend the traditional similarity-based methods, and propose a more effective mechanism to automatically extract the potential features of users and items for recommendation. However, they still have problems such as the data sparsity and cold-start. Researchers try to use the content-based methods to deal with these problems by adding various auxiliary information (side information) [39, 7, 40], such as a context review, product attributes, user social network and KG.

Among the side information, KG shows an excellent potentials in recommendations with its well-defined structure and sufficient resources. According to the given mapping relationship between entities and items, this type of methods transfers the structural knowledge of entities from knowledge facts to user-item interaction modelling. The KGE use in RS is mainly due to the successful application of several public KG datasets (such as freebase, DBpedia, Yago) in semantic [41], information extraction [42] and other tasks [43, 44]. Embedding-based methods usually pre-process KG with KGE algorithms and incorporate the learned entity embeddings into a RS framework. Collaborative Knowledge base Embedding (CKE) [9] combines CF with the structural knowledge, textual knowledge and visual knowledge in a common framework. However, in a general research and industrial environments, it is not easy to have the structural data, visual data and textual data at the same time. Moreover, CKE itself does not use the relationship information between entities in KG. The knowledge-aware network (DKN) [46] treats the entity, word and context embeddings as different channels and then designs a convolutional neural network (CNN) to combine them for news recommendation. Also,it does not use the relationship information, and the entity information in KG is directly used after preprocessing in advance. This brute-force transfer would damage the model recommendation performance. Our model uses the relationship information in KG, so that the KG vector representation is closer to the corresponding recommendation data. RippleNet [47] is a memory-network-like model that propagates users' potential KG preferences and explores their hierarchical interests. It regards the user historical interest as a KG seed, and then iteratively expands the user's interest along the KG link to discover their potential interest in candidate products. However, as the number of hops increases, the number of paths calculated by the model increases sharply, which leads to an excessive amount of calculations. MKR [48] goes a step further cross-domain recommendation (CKE) and proves that incorporating knowledge information enhances RS performance by introducing multiple objectives. It adopts a multi-task learning framework and treats the RS and KGE learning as two independent but related tasks. However, its KGE processing is not perfect, which leads to the inability to solve the common multi-head issue in KG. A multi-head phenomenon is that some relationships may correspond to multi-head entities or tail entities, leading to a severe many-to-one and many-to-many problem, making the learned triple-embedding unable to be effectively utilized.

The significant differences between M^2RK and the existing algorithm are as follows: (1) M^2RK integrates the product knowledge graph through a multi-task learning paradigm, which uses KGE learning as an auxiliary task to ensure that the model obtains a dynamic representation of users in the relevant field; (2) M^2RK proposes a multi-view attention mechanism to link different tasks. This mechanism integrates the critical information of a task-specific layer into a shared layer. It enables the model to interactively learn the representation of the item level and the knowledge level.

3 Methodology

In this section, we introduce the details of M^2RK, including the related algorithms involved in the model. We first illustrate the problem and then analyze the M^2RK pipeline followed by a detailed description of each component. We lastly discuss the learning algorithm.

*https://searchengineland.com/library/bing/bing-satori

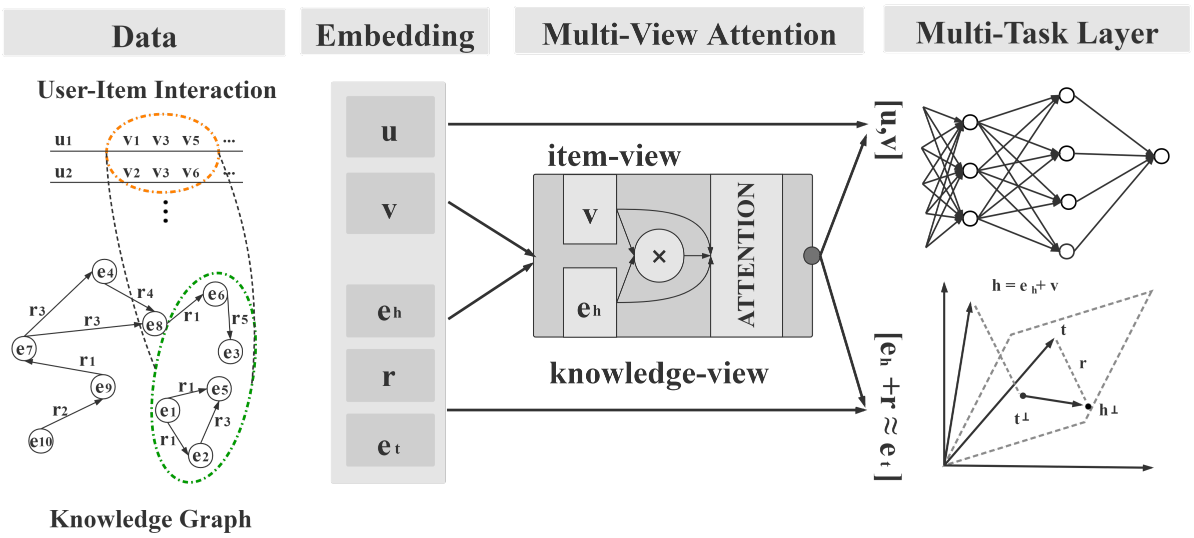


Figure 2. Overall pipeline of the proposed M^2RK. The arrows in the figure indicate the data flow. The user-item entry and the related KG facts are extracted simultaneously and then fed into the main body. Inspired by the multi-task learning, the main body consists of three components: multi-view attention unit, recommendation module (top of the multi-task layer), and knowledge graph embedding module (bottom of the multi-task layer).

Notations. The users' set is denoted as $U = \{u_1, u_2, \cdots, u_n\}$ and the items' set as $V = \{v_1, v_2, \cdots, v_n\}$. If an interaction between user u and item v is observed, then each entry $y_{u,v}$ in user-item feedback matrix $R \in \mathbb{R}^{n \times m}$ is defined as $y_{u,v} = 1$; otherwise it is defined as 0. For the sake of generality, we use the term ”entity” to refer to objects that can be mapped to KG. KG dataset consists of entities (as nodes) and relationships (as different types of edges), which can be represented as $G = (E, R)$. We can use many triplets (head entity, relationship, tail entity) to represent the facts (head entity, tail entity) $\in E$ and relationship $\in R$ in KG.

Assuming that the RS items can be linked with the KG entities, RS item set V is regarded as a subset of the KG entity set E, then having $V \in E$. By linking the RS item with the KG entity, obtained all its related KG triplets.

Problem Defintion. In the multi-task scenario, our research focuses on a joint learning of the embedding vector in KG and RS. Our problem is defined as follows: **Given** users U, items V, user-item interactions and a knowledge graph G, our **aims** is jointly (1) learn the embedding vector among triplets based on KG and (2) learn RS to recommend the item to each user based on their interaction history and KGE. This framework outputs probability $y^{'}_{uv}$ which quantifies the preference of u like v, and all its related KG triplet embedding vectors.

3.1 *Framework Overview*

To explain more clearly, we introduce a schematic M^2RK diagram from the left side of Figure 2. Besides user and item training samples, the relevant KG triplets are also included in M^2RK of the model training stage to learn the corresponding entity embeddings. In particular, the KG-enhanced RS includes two subtasks: 1) learning the latent vector representation in KG based on item association. 2) recommending products for each user u based on the embedding vector generated by the purchase history and KG facts. To fulfil these tasks, we design a multitask learning framework. The framework consists of three modules, a KGE module, RS module and multi-view attentive module. The KGE module derives the impact embedding vectors by translating the head entities and relations with the ground-truth tail entities in KG. Based on the embedding set, we design an item-entity feature sharing module, namely multi-view attentive module in which the item embedding vectors in the RS module and the entity embedding vectors in KGE module fully interact. In the RS module, the item-entity feature vector is used as an additional input to improve the performance of recommendation and explain the recommendation results. The critical M^2RK components are given follow:

Multi-View Attention Module. The product item is assumed to usually correspond to the knowledge entities in many fields, such as books, movies and music, which makes it possible to transfer the knowledge across fields. We believe that the information related to the two tasks is complementary, revealing a connectivity

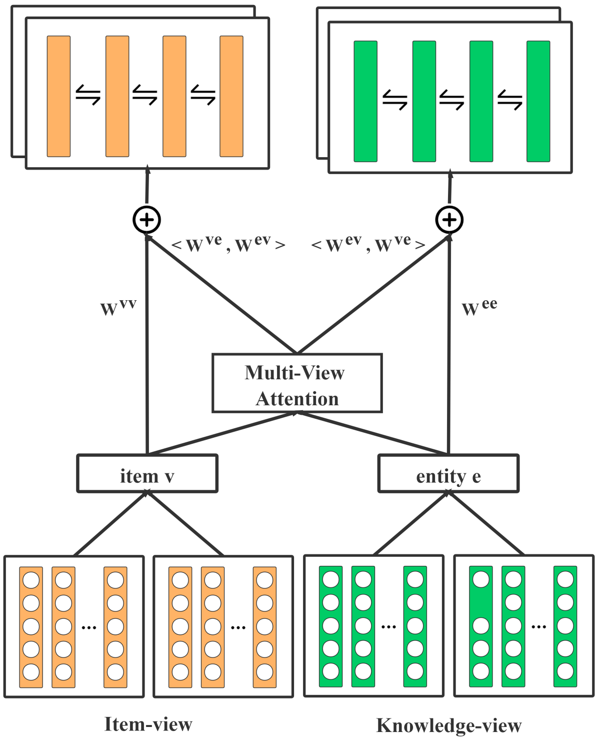


Figure 3. Diagram of a multi-view attentive unit. At the bottom, the item (orange boxes) and entity (green boxes) embedding vectors denote the input of the l layer multi-view attention module. In the middle part, the items, entities and their interactive vectors learn the attention weight and then output the transferred representation from two perspectives after completing a multiple-attention interaction.

between users and products. We propose a multi-view attention mechanism that obtains a critical information from the task-specific layer and the shared layer. Firstly, for item v and its related entities e, we construct a high-level interaction of its latent $v \in \mathbb{R}^d$ and $e \in \mathbb{R}^d$:

$$v_l = w_0 + w^{vv} v_{(l-1)} + < w^{ve}, w^{ev} > v_{(l-1)} e_{(l-1)} \quad (1)$$

$$e_l = w_0 + w^{ee} e_{(l-1)} + < w^{ev}, w^{ve} > v_{(l-1)} e_{(l-1)} \quad (2)$$

where v_l and e_l denote the augmented item and entity representations in the l layer, w_0 is the global bias, $w \in \mathbb{R}^d$ models the weight of the variable, $< \cdot, \cdot >$ is the dot product that models the interaction between the latent feature v and e. The weight of $< w^{ve}, w^{ev} >$ and $< w^{ev}, w^{ve} >$ is calculated based on the correlation between the item and entity. In this layer, both item and entity representation are simultaneously augmented by multi-view attention module of a symmetrical neural architecture (see Figure 3).

Although the attention mechanism has been widely used in deep recommendation models[6, 5, 49, 50], they only apply the attention mechanism to the item side without considering the entity-level embedding, which limits a further improvement of the recommendation performance. As shown in Figure 3, unlike other attention-sharing schemes, besides utilizing the attention from the task-specific layer, we also combine the information from the shared layer. In addition, we also obtain an attention information of one thing from two perspectives of the item and knowledge level, because a item- and knowledge-level information may considering contribution to the representation learning. In subsequent calculations, the multi-view interactive operation is expressed as M (v, e).

RS Module. The RS Module can be divided into three layers: embedding, feature-sharing and prediction layer. In the embedding layer, we first use an embedding look-up method to project each user and item into low-dimensional vectors. Then, the user feature vectors are extracted as a latent condensed feature using the l-layer multi-layer perceptron (MLP), as shown below :

$$u_l = MLP(MLP(\cdots MLP(u))) \quad (3)$$

where $MLP(x) = \sigma(Wx + b)$ is a multi-layer perception composed of hidden layers with weight W, bias b and relu $\sigma(\cdot)$ as the activation function.

In the feature-sharing layer, we use a multi-view attention unit to extract the item latent condensed feature, as shown below:

$$v_l = M^l_{e \sim S(v)}(v, e)[v] \quad (4)$$

where $S(v)$ is the knowledge triplet set of the associated entities of item v.

After getting user u's latent feature u_l and item v's latent feature v_l, we combine the two pathways by inner product f_{RS}. The final predicted probability of user u engaging item v is:

$$\hat{y}_{uv} = \sigma\left(f_{RS}\left(u_l, v_l\right)\right) \quad (5)$$

where $\hat{y}_{uv}$ is the estimated score which quantifies the probability that u like v.

KGE Module. For the KGE Module, we propose a translational distance architecture similar to TransH [22] to learn representation vectors of head h and relation r. For the sake of symmetry, the KGE Module is also divided into three layers: the embedding, feature-sharing and prediction layer. In the embedding layer, we use an embedding look-up layer to project each entity and its corresponding relations into low dimensional vectors. Then we use an l-layer MLP to extract relation feature vector from a latent condensed feature, as shown below :

$$r_l = MLP(MLP(\cdots MLP(r)) \quad (6)$$

where $MLP(x) = \sigma(Wx + b)$ is a multi-layer perception composed of hidden layers with weight W, bias b and relu $\sigma(\cdot)$ as the activation function.

In the feature sharing layer, we use a multi-view attention unit to extract the head latent condensed feature, as shown below:

$$h_l = M^l_{v \sim S(h)}(v, h)[h] \tag{7}$$

where $S(h)$ is the history purchase interaction set of associated items of entity h.

The basic idea of the translation distance method is to learn embeddings for entities and relations satisfying $h + r \approx t$ if there is a triplet (h, r, t) in KG. However, some relations may correspond to multiple-head or tail entities, resulting in a severe many-to-one and many-to-many issue, which makes the learned triplet embedding unable to be effectively utilized. To solve the problem, each relation is assumed to build a hyperplane, and only when the head and tail entities are projected to the same hyperplane, the translation between the head and tail entity is valid. It defines the energy score function for a triplet in the prediction layer, as shown below:

$$f_{KG}(h, r, t) = \| h^{\perp} + r - t^{\perp} \| \tag{8}$$

where t is the real feature vector of the tail entity, $h^{\perp}$ and $t^{\perp}$ are projected entity vectors:

$$h^{\perp} = h - w_r^T h w_r$$

$$t^{\perp} = t - w_r^T t w_r$$

where w_r and r are two learned vectors of relation r, w_r denotes the projection vector of the corresponding hyperplane. $\| \cdot \|$ indicates the $L1$-norm distance function, and the lower score of $f_{KG}(h, r, t)$ indicates that the triplet may be reasonable, otherwise no.

3.2 Learning Algorithm

The goal of multi-task learning (MTL) is to maximize the following posterior probability of our model parameters θ given knowledge triplet set $\mathcal{G}$ and recommendation sample set $\mathcal{Y}$. According to the Bayes rule, this objective is defined as :

$$\begin{aligned} \max p(\theta|\mathcal{G}, \mathcal{Y}) &= \max \frac{p(\theta, \mathcal{G}, \mathcal{Y})}{p(\mathcal{G}, \mathcal{Y})} \\ &= \max p(\theta) p(\mathcal{G}|\theta) p(\mathcal{Y}|\theta, \mathcal{G}) \end{aligned} \tag{9}$$

where θ includes the embeddings of each items, entity and relation. $p(\theta)$ is a θ's prior probability which is set to follow the Gaussian distribution of zero mean and 0.1 standard deviation. $p(\mathcal{G}|\theta)$ is the likelihood of observing $\mathcal{G}$ given θ, and $p(\mathcal{Y}|\theta, \mathcal{G})$ is the likelihood of observing $\mathcal{Y}$ given $\mathcal{G}$ and θ, which is defined as the product of the Bernoulli distribution. Then, the MTL loss function is :

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{RS} + \lambda_1 \mathcal{L}_{KGE} + \lambda_2 \| \theta \|_2^2 \\ &= - \sum_{(u,v) \in \mathcal{Y}} \left[y_{uv} \log \hat{y}_{uv} + (1 - y_{uv}) \log(1 - \hat{y}_{uv}) \right] \\ &+ \lambda_1 \left(\sum_{(h,r,t) \in \mathcal{G}} \| h^{\perp} + r - t^{\perp} \| - \sum_{(h', r, t') \notin \mathcal{G}} \| h^{'\perp} + r - t^{'\perp} \| \right) \\ &+ \lambda_2 \| \theta \|_2^2 \end{aligned} \tag{10}$$

where $\| \theta \|_2^2$ is the control term to prevent over-fitting, and λ_1 and λ_2 are control parameters. We obtain the value of λ_1 and λ_2 through tuning experiments. In the next section, we will show that the learning result of one task can be used as a hint to guide another task to learn better.

Algorithm 1: Multi-Task Training for M^2RK

```
   Input: Interaction matrix $\mathcal{Y}$, knowledge graph $\mathcal{G}$
   Output: Prediction function $\mathcal{F}(u, v|\theta, \mathcal{Y}, \mathcal{G})$
1  Initialize all parameters $\theta$;
2  for iter = 1;iter≤ max_iter; iter++ do
3      for t = 1;t≤ $T^{RS}$; t++ do
4          1 Sample mini batch of positive and
            negative interactions from $\mathcal{Y}$;
5          2 Sample $e \sim S(v)$ for each item $i$ in
            mini batch;
6          3 Update parameters of $\mathcal{F}$ by gradient
            descent;
7      end
8      for t = 1;t≤ $T^{KGE}$; t++ do
9          1 Sample mini batch of true and false
            triplets from $\mathcal{G}$;
10         2 Sample $v \sim S(h)$ for each head entity
            $h$ in minibatch;
11         3 Update parameters of $\mathcal{F}$ by gradient
            descent;
12     end
13 end
14 reutrn $\mathcal{F}(u, v|\theta, \mathcal{Y}, \mathcal{G})$
```

Obviously, it is complicated to solve the above problems directly. We use the stochastic gradient descent (SGD) algorithm to optimize the loss function iteratively. The learning algorithm of M^2RK is given in Algorithm 1. In each training iteration, in order to make the calculation more efficient, we randomly sample a small batch of positive/negative interaction data from $\mathcal{Y}$ and extract a true/false triplet from $\mathcal{G}$ following a negative sampling strategy [51]. We repeatedly train RS tasks (lines 3-5) T_{RS} times, and then train KGE tasks (lines 6-8) T_{KGE} times. Then, we calculate the gradient of loss $\mathcal{L}$ related to model parameter θ, and update all parameters

Table 1. Basic statistics for the three datasets and its related knowledge graph.

		Movie	Book	Music
$\mathcal{Y}$	Users	6,036	17,860	1,872
	Items	2,347	14,910	3,846
	Ratings	753,772	139,746	42,346
	Sparsity	95.74%	99.99%	99.72%
$\mathcal{G}$	Entities	6,729	24,039	9,366
	Relations	7	10	60
	Triplets	20,195	19,793	15,518

through a backpropagation based on a sampled mini-batch. In addition, the hyperparameters T_{RS} and T_{KGE} of the algorithm represent the number of times the RS task and the KGE task are trained separately in each epoch. In a practical experiment, the values of T_{RS} and T_{KGE} depend on the size of the interaction dataset and its related knowledge graph tripltes. We take a fixed value of the epoch and set the epoch to 20 in further experiments.

4 Experiments

In this section, we first introduce the used datasets and the implementation details of the experiments. Then, we raise some research questions (RQ) about the rationality and efficiency of the proposed model. We try to answer the following research questions:

RQ1: Is the proposed multi-task learning model more effective than other models?

RQ2: Is it useful to incorporate side information (knowledge graph) into RS?

RQ3: Among the alternative and joint training, which is more suitable for incorporating KGE learning into the M^2RK?

RQ4: Is the multi-task learning scheme effective in KGE method? Is the proposed method also superior to the baseline?

4.1 Datasets

Our task is to prepare two types of datasets, namely RS and KG dataset. A detailed description of the original dataset is given follow:

RS Datasets. We consider three widely used RS datasets, e.g. MovieLens, Book-Crossing and Last.FM which cover the fields of the movie, book, and music, respectively.

- **MovieLens 1M** [*] dataset describes the users' preferences on the movies. The interaction data format is in the form of $\langle user, item, level, timestamp \rangle$, which represents the user's rating score on a specific movie at a specific time. The dataset is a well-known benchmark dataset containing 753,772 ratings from 6036 users on 2347 movies.
- **Book-Crossing** [†] dataset describes users' preferences on the book products, which has a data form, i.e.,$\langle user, item, rating \rangle$. The dataset is very sparse, containing 139,746 ratings from 17,860 users over nearly 14,910 items.
- **Last.FM** [‡] dataset describes the users' interaction records on music. It records the listening count of a song by a user but does not contain the rating information. The dataset contains 42,346 ratings from 1,872 users, including nearly 3,846 items.

KG Dataset. We adopt the large-scale public Microsoft Satori service, which is a public knowledge database building facts in the form of triplets $\langle head, relation, tail \rangle$.

Table 1 shows the statistics of the MovieLens-1m, Book-Crossing and Last.FM datasets. After preprocessing, MovieLens-1m has 6040 users, 2347 projects and 753,772 interactive records. The movie KG has 20,195 triplets, 6,729 entities and seven relations. The Book-Crossing has 17,860 users, 14,910 items and 139,746 interaction records. The related book KG is composed of 19,793 triplets, 24,039 entities, and ten relations. The Last.FM has 1,872 users, 3,846 items and 42,346 interactive records. The related music KG has 15,518 triplets, 9,366 entities and 60 relations. We also list the sparsity rate of each dataset. The sparsity of Book-Crossing and Last.FM is considerable.

4.2 Implementation Details

For the RS task, we use a stochastic gradient optimizer with the learning rate selected among $\{0.005, 0.02, 0.01, 0.1\}$. To be consistent with the RS task implementation, we also use the stochastic gradient descent for the KGE module with learning rate selected among $\{0.001, 0.02, 0.01, 0.1\}$. All the above modules are trained using negative sampling. For each user, we randomly select ten products that do not appear in the interaction record to form negative samples. For each KG triplet, we sample three negative triplets based on the same operation. The best parameters settings are: the batch size is 4096; the dimension of embeddings is 8; the learning rate for the RS task is 0.02, and the learning rate for the KGE task is 0.01. When conducting experiments, 70% of the items of each user are leveraged for training, while the rest are used for testing. All the implementations are in Tensorflow.

4.3 Compared baselines

In order to emphasize the superior performance of M^2RK, we compare it with the following state-of-the-art models and describe them in detail:

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

[†] http://www2.informatik.uni-freiburg.de/ cziegler/BX/

[‡] https://grouplens.org/datasets/hetrec-2011/

Table 2. Performance on the *AUC* and the *Accuracy* in CTR prediction between the baselines and our model (bold numbers indicate the best performance of each column).

Model	**MovieLens-1M**		**Book-Crossing**		**Last.FM**	
	AUC	*ACC*	*AUC*	*ACC*	*AUC*	*ACC*
PER	0.710 (-21.2%)	0.664 (-19%)	0.623 (-12.3%)	0.588 (-12.4%)	0.633 (-17%)	0.596 (-16.3%)
CKE	0.801 (-12.1%)	0.742 (-11.2%)	0.671 (-7.5%)	0.633 (-7.9%)	0.744 (-5.9%)	0.673 (-8.6%)
DKN	0.655 (-26.7%)	0.589 (-25.6%)	0.622 (-12.4%)	0.589 (-12.3%)	0.602 (-20.1%)	0.581 (-17.8%)
RippleNet	0.920 (-0.2%)	0.842 (-1.2%)	0.729 (-1.7%)	0.662 (-5.0%)	0.768 (-3.5%)	0.691 (-6.8%)
libFM	0.892 (-3.0%)	0.812 (-4.2%)	0.685 (-6.1%)	0.640 (-7.2%)	0.777 (-2.6%)	0.709 (-5.0%)
Wide&Deep	0.898 (-2.4%)	0.820 (-3.4%)	0.712 (-3,4%)	0.624 (-8.8%)	0.756 (-4.7%)	0.688 (-7.1%)
MKR	0.917 (-0.5%)	0.843 (-1.1%)	0.734 (-1.2%)	0.704 (-0.8%)	0.797 (-0.6%)	0.752 (-0.7%)
Ours	**0.922**	**0.854**	**0.746**	**0.712**	**0.803**	**0.759**

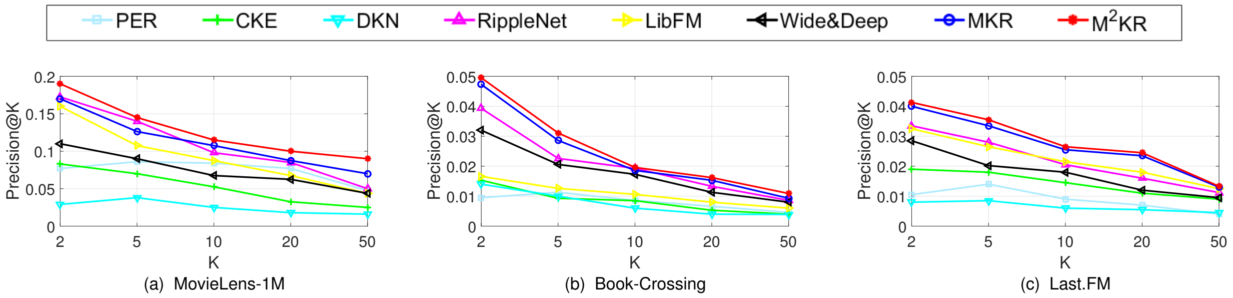
- **PER** [52] regards KG as a heterogeneous information network and then constructs meta-path-based features between items. By using the meta-path, various recommendation strategies can be invented, providing interpretability as well as improving the recommendation accuracy. The advantage of this type of method is that it makes a full and intuitive use of the KG network structure.
- **CKE** [9] introduces a structural information (head and tail entities and the relations between them), text data (textual description of an entity), image data (picture information related to the current entity, such as movie posters or books cover) to improve the RS quality. The paper describes the items based on KGE.
- **DKN** [46] is a recommendation model which combines KG entity representation with a neural network. According to the given knowledge map, the entity and context embedding in the knowledge map are transformed into the same space of semantic embedding by using a matrix to form a new multi-channel embedding representation like pictures. Then, the convolution neural network is used to generate the feature representation of the user history records and candidate products.
- **RippleNet** [47] is a state-of-the-art algorithm that regards KG as an auxiliary RS information source. It regards the user's historical interest as a seed set in KG, and then iteratively expands the user's interest along the KG link to discover his/her potential interest in the candidate item.
- **libFM** [45] Its full name is Factorization Machine Library. It is the basic model for the CTR prediction. Its advantage lie in processing discretization features and simplicity.
- **Wide&Deep** [1] realizes a unified modelling of the linear and deep model, which makes the model to have the advantages of the logistic regression and deep neural network. That is to say that it has the capabilities of exploiting the correspondence available in the historical records and exploring new feature combinations that have never or rarely occurred in the past.
- **MKR** [48] embraces a more innovative idea. It adopts a multi-task learning framework, and regards RS and KGE learning as two separate but associated tasks. The algorithm also designs cross and compression units for recommendation tasks and KGE learning tasks to learn and transfer the high-level feature interaction between them automatically.

4.4 Experiments and Performance Study

4.4.1 Comparison with baseline methods: First, we compare the performance of all models on different datasets to answer RQ1. The area under the curve (AUC) and accuracy (ACC) scores (percentage values) are listed in Table 2. Figure 4 and Figure 5 show the Top-*k* recommendation, respectively. Some experimental observations are given follow.

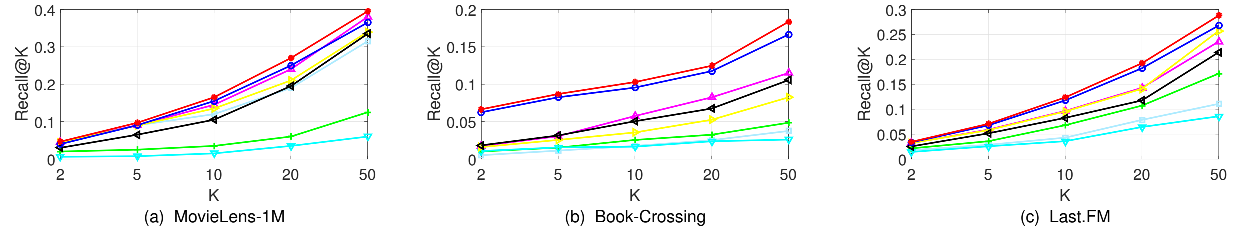
The PER performance is relatively poor. The reason may be that PER requires a manually designed meta path to maintain the necessary semantic information. However, user-defined meta-paths are challenging to implement on optimal state. The CKE performance is moderate. There may be two reasons. First, in our experiment, there is only structural data, lacking visual and text data. Second, CKE itself does not utilise the relationship information between KG entities. DKN's performance in movie, book and music recommendations is not competent (see Table 2). The reason may be that the product itself can not provide a useful information, and the algorithm does not cover the relational KG representation at all.

RippleNet performes relatively well. The reason may be that it makes a full use of the KG information. However, as the number of hops increases, the number of paths calculated by the model increases sharply, which leads to the amount of calculations required than other methods. Also, it is not certain whether the long inference path is still helpful to the user's current preference. libFM is the most favourite solution for the industry. It considers detailed factors and performs well on dense datasets (i.e., MovieLens 1M), but it performs moderate on sparse datasets (i.e., Book-Crossing, Last.FM). It



(a) MovieLens-1M (b) Book-Crossing (c) Last.FM

Figure 4. The results of *Precision@K* in top-*K* recommendation.



(a) MovieLens-1M (b) Book-Crossing (c) Last.FM

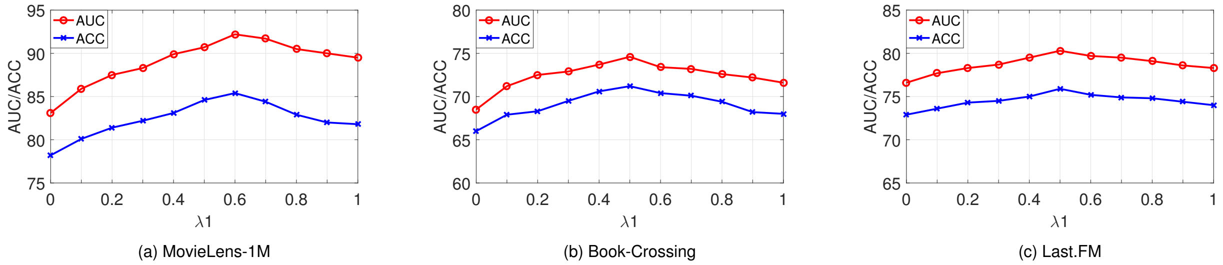
Figure 5. The results of *Recall@K* in top-*K* recommendation.

requires a lot of manual feature engineering. Although the model is simple, manual work is a lot more onerous. Wide&Deep achieves satisfactory performance, proving the effectiveness of learning low-level and high-level combined features. However, the input of the wide part still relies on manual feature engineering. It is a classical parallel connection mode of the linear model and neural network. MKR performs best in the baseline (see Table 2 and Figure 4 and 5), proving the effectiveness of combining the RS loss with the KGE loss. On the basis of MKR, we make structural adjustments in the task sharing unit, and KGE task: (1) A multi-view attention mechanism is proposed to fuse representations of various tasks. (2) In the KGE task design, we are more concerned about the multi-head (i.e., many-to-many) phenomenon according to the dataset conditions. The AUC/ACC values obtained on the movie and music datasets are 0.922/0.854 and 0.803/0.759, respectively. An excellent performance of 0.746/0.712 is obtained on the book dataset with a more serious sparsity, which once again illustrates the KG supplementary role to RS. The performance curve of Precision@K and Recall@K in Figure 4 and 5 demonstrates that our work make a noticeable improvement. Precision@K and Recall@K are two evaluation metrics for deciding whether the content and retrieval conditions are correlated. The recall is the percentage of items selected from the relevant items in the repository, and the precision is the percentage of the items select from the items selected by the query.

Comparison results show that M^2RK outperform all baselines in datasets (answer ”yes” to RQ1). Especially in the datasets with cold-start items (Book-Crossing), M^2RK has more remarkable superiority. In addition to M^2RK, compared with models that do not include side information, models that utilize the knowledge graph (e.g., MKR, RippleNet) also have advantages. This also demonstrates that the effect of incorporating the KGE module and that the sparsity problem of cold-start items can be significantly alleviated (answer ”yes” to RQ2).

The model loss function is designed by a key parameter $\lambda 1$ to balance the RS loss and the KGE loss. The loss coefficient $\lambda 1$ is an important parameter that controls the score learning capacity. We investigate the impact of parameters $\lambda 1$ in M^2RK by varying $\lambda 1$ from 0 to 1 while keeping other parameters fixed. As mentioned before, the KG triplets are used as a side information combined with the user interaction history to predict the user's preference. With the increase of $\lambda 1$, the KG triplets in the total loss will be larger. As displayed in Figure 6. As $\lambda 1$ increases, the performances of AUC and ACC firstly show an upward trend and then decreases. M^2RK achieves the best performance when $\lambda 1$ is around 0.5 or 0.6. The values of $\lambda 1$ depend on the size of the interaction dataset and its related KG triplets. This is because moderate KG tripltes can be used to cope with the data sparsity and cold start problem, while a too large capacity of the side information will lead to a brute-force transfer which can damage model recommendation performance. Optimal $\lambda 1$ is empirically stable across datasets, indicating that the proposed method is to some extent robust to $\lambda 1$.

4.4.2 Analysis on training strategies incorporating the KGE learning: As we mentioned before, there are two training strategies for the multi-optimization loss:



(a) MovieLens-1M

(b) Book-Crossing

(c) Last.FM

Figure 6. Parameter sensitivity of M^2RK on three datasets (a) MovieLens-1M; (b) Book-Crossing; and (c) Last.FM.

alternative training and joint training. The training loss function formula of alternative learning is provided in Section 3.2, and the training loss function formula of joint learning is provided in the Appendix. In order to answer RQ3, we train the M^2RK with the two training strategies and compare their loss curves.

The loss value is determined which training strategy is more suitable for our experimental scenario (see Figure 7). The loss value of the two training strategies is mentioned in each epoch. For the loss value of each epoch, the average value between mini-batches as the final result. A comparison shows that the performance of the alternative training is better than that of the joint training, and the loss value obtained by alternative training is smaller than that obtained by joint training. Compared with the alternative training that is more smooth in the whole training process, there are apparent fluctuations in the joint training at the beginning of the training process. Moreover, the joint training needs more training time than the alternative training because large volume of knowledge facts of each item, but the effect is not ideal. To sum up, the alternative training is more suitable than the joint training.

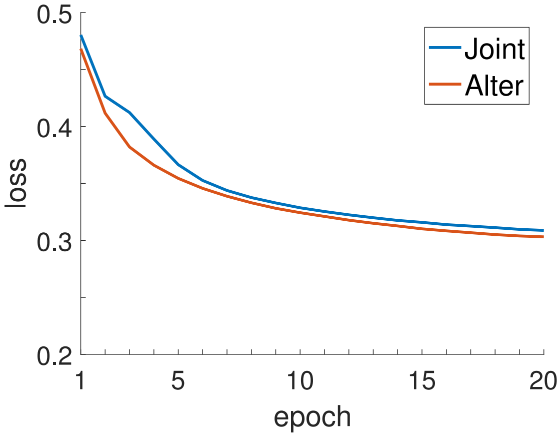
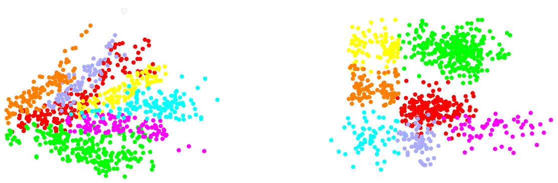


Figure 7. Comparison of the loss curves in the process of the alternative (orange colour) and joint training (blue colour).

4.4.3 Study of the multi-task learning paradigm effect on KGE: In order to verify the effectiveness of the multi-task learning for KGE learning, we visualize the KGE distribution of some entities in a movie dataset, learned separately through different mechanisms as shown in Figure 8, where the dots with different colours represent movies with different relations. The different KGE algorithms have a unique ability to summarize the relevance of facts. The difference between them is that there are many aspects in their observation angles, and different angles need different learning methods.



(a) TransH

(b) M^2RK

Figure 8. Embedding distributions in a movie KG under differnet learning mechanisms.

In Figure 8(a), the movie embedding is learned by feeding movie entities into TransH [22]. If two entities belong to the same relation, they are close in the vector space. As shown in the sub-figure, the learned embedding makes many movies with different scenes converge too much, so it is impossible to distinguish different types of movies. In Figure 8(b), the entities in KG are learned by M^2RK through MTL. It can be seen from the sub-figure that using the multi-task learning helps to model more explicit class boundaries and improve separation between classes. Such results are in line with our intuitive understanding. For the uncommon data, the data density of its local area is low, and the model can not simulate the boundary of these low-density areas during a learning process, resulting in an ambiguity and poor generalization. In contrast, the interactive RS data can effectively improve the low-density of the area sample, coupled with more robust regularization.

Compared with other KGE algorithms, M^2RK has two main advantages: (1). It fully considers a common KG many-to-many situation. (2). When learning the embedding vector, it merges the embedding vector from the item view, which means that M^2RK integrates the actual RS perspective while observing KG. Therefore, M^2RK constrains the learning process through additional tasks, which makes the structure learning of the data space more complete and the extracted information more comprehensive. Compared with the imbalance of the semantic information brought by other methods, M^2RK effectively reduces the dependence of the network on high-level semantic features and overfits the tail data. The learned feature representation is more robust and easy to generalize (answer ”yes” to RQ4).

5 Conclusion

The paper studies a multi-task learning method to solve RS and KGE simultaneously. To this end, we propose a novel multi-task learning scheme that uses a common sense in related fields and invents a multi-view attention learned from various perspectives, which enables these tasks to interact with each other and learn more comprehensive representations from the item view and knowledge view. Based on component studies, we also investigate the significance of learning knowledge embeddings and the impact of different training strategies. We conduct extensive experiments on three datasets, and the results justify the superiority of M^2RK over the state-of-the-art models.

The proposed method tries to solve the cold-start problems for new users. Due to the more knowledge facts of products, it also uses information from multiple fields to supplement the user information, which makes the expression of user interest more accurate. For further work, we plan (1) to further investigate the methods of expressing item-entity interactions; (2) and design more effective network architecture to explore the users' potential interests and to improve the performance.

Appendix

In the process of optimizing loss $\mathcal{L}$, there are two training strategies for the multi-task learning: alternative and joint training. For the alternative training, it is already given in the Section 3.2. For the joint training, we have:

$$\begin{aligned}\mathcal{L}_{joint} = -\sum_{(u,v)\in\mathcal{Y}} \Bigg\{ & [y_{uv}\log\hat{y}_{uv} + (1-y_{uv})\log(1-\hat{y}_{uv})] \\ & + \lambda_1\Big(\sum_{(h,r,t)\in\mathcal{G}(v)} \| h^{\perp} + r - t^{\perp} \| - \sum_{(h',r,t')\notin\mathcal{G}(v)} \| h'^{\perp} + r - t'^{\perp} \| \Big)\Bigg\} \\ & + \lambda_2 \| \theta \|_2^2\end{aligned}$$

where $\| \theta \|_2^2$ is the control term to prevent over-fitting, and λ_1 and λ_2 are control parameters. $\mathcal{G}(v)$ is a subgraph which contains all related triplets to item v.

Acknowledgements

The work has been partially supported by grants from: National Natural Science Foundation of China (No. 12071458, 71731009).

References

[1] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, “Wide & deep learning for recommender systems,” in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.

[2] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, “Deepfm: a factorization-machine based neural network for ctr prediction,” *arXiv preprint arXiv:1703.04247*, 2017.

[3] R. Wang, B. Fu, G. Fu, and M. Wang, “Deep & cross network for ad click predictions,” in *Proceedings of the ADKDD'17*, 2017, pp. 1–7.

[4] P. Covington, J. Adams, and E. Sargin, “Deep neural networks for youtube recommendations,” in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 191–198.

[5] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, “Deep interest evolution network for click-through rate prediction,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 5941–5948.

[6] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, “Deep interest network for click-through rate prediction,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1059–1068.

[7] L. Zheng, V. Noroozi, and P. S. Yu, “Joint deep modeling of users and items using reviews for recommendation,” in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 2017, pp. 425–434.

[8] C. Chen, M. Zhang, Y. Liu, and S. Ma, “Neural attentional rating regression with review-level explanations,” in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1583–1592.

[9] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, “Collaborative knowledge base embedding for recommender systems,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 353–362.

[10] X. He, T. Chen, M.-Y. Kan, and X. Chen, "Tri-rank: Review-aware explainable recommendation by modeling aspects," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 1661–1670.

[11] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 2014, pp. 83–92.

[12] R. Burke, "Knowledge-based recommender systems," *Encyclopedia of library and information systems*, vol. 69, no. Supplement 32, pp. 175–186, 2000.

[13] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip, "Heterogeneous information network embedding for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 357–370, 2018.

[14] G. Piao and J. G. Breslin, "A study of the similarities of entity embeddings learned from different aspects of a knowledge base for item recommendations," in *European Semantic Web Conference*. Springer, 2018, pp. 345–359.

[15] C. Liu, L. Li, X. Yao, and L. Tang, "A survey of recommendation algorithms based on knowledge graph embedding," in *2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI)*. IEEE, 2019, pp. 168–171.

[16] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, "An integrated environment for the development of knowledge-based recommender applications," *International Journal of Electronic Commerce*, vol. 11, no. 2, pp. 11–34, 2006.

[17] W. Carrer-Neto, M. L. Hernández-Alcaraz, R. Valencia-García, and F. García-Sánchez, "Social knowledge-based recommender system. application to the movies domain," *Expert Systems with applications*, vol. 39, no. 12, pp. 10 990–11 000, 2012.

[18] J. K. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning," *Artificial intelligence review*, vol. 50, no. 1, pp. 21–48, 2018.

[19] Y. Sun, C. Lu, R. Bie, and J. Zhang, "Semantic relation computing theory and its application," *Journal of Network and Computer Applications*, vol. 59, pp. 219–229, 2016.

[20] Y. Sun, R. Bie, and J. Zhang, "Measuring semantic-based structural similarity in multi-relational networks," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 12, no. 1, pp. 20–33, 2016.

[21] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in neural information processing systems*, 2013, pp. 2787–2795.

[22] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes." in *Aaai*, vol. 14, no. 2014. Citeseer, 2014, pp. 1112–1119.

[23] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.

[24] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[25] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017.

[26] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[27] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 995–1000.

[28] S. Rendle, "Factorization machines with libfm," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, pp. 1–22, 2012.

[29] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[30] A. Andoni, R. Panigrahy, G. Valiant, and L. Zhang, "Learning polynomials with neural networks," in *International conference on machine learning*, 2014, pp. 1908–1916.

[31] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Advances in neural information processing systems*, 2016, pp. 550–558.

[32] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based neural networks for user response prediction," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 1149–1154.

[33] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 355–364.

[34] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," *arXiv preprint arXiv:1708.04617*, 2017.

[35] Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu,

and J. Mao, “Deep crossing: Web-scale modeling without manually crafted combinatorial features,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 255–262.

[36] S. Zhai, K.-h. Chang, R. Zhang, and Z. M. Zhang, “Deepintent: Learning attentions for online advertising with recurrent neural networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1295–1304.

[37] W. Zhang, T. Du, and J. Wang, “Deep learning over multi-field categorical data,” in *European conference on information retrieval.* Springer, 2016, pp. 45–57.

[38] Q. Liu, F. Yu, S. Wu, and L. Wang, “A convolutional click prediction model,” in *Proceedings of the 24th ACM international on conference on information and knowledge management*, 2015, pp. 1743–1746.

[39] Y. Zhen, W.-J. Li, and D.-Y. Yeung, “Tagicofi: tag informed collaborative filtering,” in *Proceedings of the third ACM conference on Recommender systems*, 2009, pp. 69–76.

[40] M. Jamali and M. Ester, “Trustwalker: a random walk model for combining trust-based and item-based recommendation,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 397–406.

[41] B. Jia, X. Huang, and S. Jiao, “Application of semantic similarity calculation based on knowledge graph for personalized study recommendation service,” *Educational Sciences: Theory & Practice*, vol. 18, no. 6, 2018.

[42] M. Manica, C. Auer, V. Weber, F. Zipoli, M. Dolfi, P. Staar, T. Laino, C. Bekas, A. Fujita, H. Toda *et al.*, “An information extraction and knowledge graph platform for accelerating biochemical discoveries,” *arXiv preprint arXiv:1907.08400*, 2019.

[43] X. Huang, J. Zhang, D. Li, and P. Li, “Knowledge graph embedding based question answering,” in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 105–113.

[44] D. Hakkani-Tür, A. Celikyilmaz, L. Heck, G. Tur, and G. Zweig, “Probabilistic enrichment of knowledge graph entities for relation detection in conversational understanding,” in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[45] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “Bpr: Bayesian personalized ranking from implicit feedback,” *arXiv preprint arXiv:1205.2618*, 2012.

[46] H. Wang, F. Zhang, X. Xie, and M. Guo, “Dkn: Deep knowledge-aware network for news recommendation,” in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1835–1844.

[47] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, “Ripplenet: Propagating user preferences on the knowledge graph for recommender systems,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 417–426.

[48] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, “Multi-task feature learning for knowledge graph enhanced recommendation,” in *The World Wide Web Conference*, 2019, pp. 2000–2010.

[49] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, “Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1441–1450.

[50] H. Zhu, X. Li, P. Zhang, G. Li, J. He, H. Li, and K. Gai, “Learning tree-based deep model for recommender systems,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1079–1088.

[51] H. Caselles-Dupré, F. Lesaint, and J. Royo-Letelier, “Word2vec applied to recommendation: Hyperparameters matter,” in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 352–356.

[52] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, “Personalized entity recommendation: A heterogeneous information network approach,” in *Proceedings of the 7th ACM international conference on Web search and data mining*, 2014, pp. 283–292.

Yingjie Tian is a professor of University of Chinese Academy of Sciences. His research interests include machine learning,deep learning, data science and intelligent knowledge management.

Kunlong Bai is a master student at the School of Computer Science & Technology, University of Chinese Academy of Sciences. His current research interests include data mining and machine learning.

Dalian Liu is currently an Associate Professor with the Beijing Union University. Her current research interests include optimization and data mining.