Jointly Learning Propagating Features on the Knowledge Graph for Movie Recommendation

Yun Liu, Jun Miyazaki(✉), and Qiong Chang

Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo, Japan
liu@lsc.c.titech.ac.jp, {miyazaki,q.chang}@c.titech.ac.jp

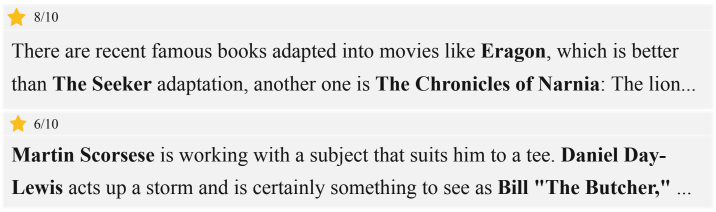
Abstract. Knowledge graphs are widely used as auxiliary information to improve the performance in recommender systems. This enables items to be aligned with knowledge entities and provides additional item attributes to facilitate learning interactions between users and items. However, the lack of user connections in the knowledge graph may degrade the profiling of user preferences, especially for explicit user behaviors. Furthermore, learning knowledge graph embeddings is not entirely consistent with recommendation tasks due to different objectives. To solve the aforementioned problems, we extract knowledge entities from users' explicit reviews and propose a multi-task framework to jointly learn propagating features on the knowledge graph for movie recommendations. The review-based heterogeneous graph can provide substantial information for learning user preferences. In the proposed framework, we use an attention-based multi-hop propagation mechanism to take users and movies as center nodes and extend their attributes along with the connections of the knowledge graph by recursively calculating the different contributions of their neighbors. We use two real-world datasets to show the effectiveness of our proposed model in comparison with state-of-the-art baselines. Additionally, we investigate two aspects of the proposed model in extended ablation studies.

Keywords: Multi-task learning · Knowledge graph · Review-based recommendation · Personalized recommender systems

1 Introduction

Knowledge graphs (KGs) contain a large number of item attributes, which are widely used as auxiliary information to improve recommendation performance. One of the most commonly-used practices is aligning items with knowledge entities in a KG, which enables to explore item attributions along with the connections of the entities [1–5,22].

The key point of KG-based recommender systems (RSs) is how to profile user preferences on the basis of the KG. Existing works profile user preferences by first integrating user behaviors into the graph and then designing an effective method to learn user preferences along with the connections in the graph [6–8].


8/10

There are recent famous books adapted into movies like **Eragon**, which is better than **The Seeker** adaptation, another one is **The Chronicles of Narnia**: The lion...

6/10

Martin Scorsese is working with a subject that suits him to a tee. **Daniel Day-Lewis** acts up a storm and is certainly something to see as **Bill "The Butcher,"** ...

Fig. 1. Illustration of movie reviews with knowledge mentions. The bold words are knowledge mentions aligned with entities in the KG.

To handle the interactions between users and items, they treat the interactions as KG edges, and define the built heterogeneous graph as a collaborative KG [3]. Existing KG-based recommendation methods are roughly classified into two types: path-based and embedding-based.

Path-based methods explore paths between users and items to learn the multiple hops information as user preferences for enhancing recommendation performances. They usually treat KG-based recommendation tasks as multi-hop reasoning problems [2,9] or define meta-paths to extract specific patterns between users and items to improve recommendation accuracy [10]. Embedding-based methods represent users and items as entity embeddings by using current KG embedding (KGE) algorithms, such as TransE [12] and TransR [13]. The user preferences of these works are depicted by the linked neighbors of users in the graph.

Although these methods can improve the corresponding recommendation performances, they also have several deficiencies. First, they usually integrate the user-item implicit interactions (e.g., clicks and browses) directly into the graph, which is unsuitable for explicit user behaviors (e.g., ratings and reviews). Second, all user neighbors in the graph are items, which is insufficient to profile user preferences based on explicit behaviors. Third, although KGs have their benefits in learning user preferences on the basis of the connections on the graph, directly using entity embeddings for recommendation tasks results unnecessary losses in accuracy.

User reviews are widely used as auxiliary information in RSs and have been successfully applied to improve recommendation performance [23,24]. Existing review-based RSs usually extract topics or semantic embeddings from reviews as features to profile user preferences for recommendation [14,24]. However none consider the substantial knowledge information contained in reviews [23–25]. Figure 1 shows two movie reviews from users with knowledge mentions aligned with KGs. We can see that movie reviews contain substantial knowledge mentions corresponding to knowledge entities.

To address the limitations represented by the current KG-based works, and inspired by the success applying reviews to RSs, we propose a novel recommendation framework, jointly learning propagation features on the KG (JPKG), which can learn multi-hop propagation features as user preferences on the basis of explicit review behaviors of users for movies. The review entities extracted from reviews based on KGs can be considered as neighbors of users/movies in the graph to assist in profiling user preferences and movie properties. On the basis of the review entities, we first construct a review-based heterogeneous KG, as shown in Fig. 2. To fully exploit user preferences on the graph, we then introduce an attention-based multi-hop propagation mechanism that updates a node embedding of a user/movie on the basis of the different contributions of its neighbors. To bridge the differences between the knowledge embedding learning and recommendation, we adopt a multi-task learning framework to jointly learn the propagation feature on the KG to predict movie ratings.

The contributions of our work are summarized as follows:

- We built a review-based heterogeneous KG to address the lack of user connections, which considers the movie-related entities and contains users' connections to their review entities.
- We designed a multi-task framework to jointly learn multi-hop features of user/movies, which can recursively learn the different contributions of neighbors to users/movies.
- We conducted experiments on two public datasets, demonstrating the effectiveness of JPKG, especially on sparse datasets.

2 Methodology

2.1 Problem Formulation

In this paper, we focus on generating propagating links through the jointly learning of a recommendation task and KG linking task to recommend a movie to a user. Let $\mathcal{U} = \{u_1, u_2, ..., u_{|U|}\}$ and $\mathcal{M} = \{m_1, m_2, ..., m_{|M|}\}$ denote the user set containing $|U|$ users and the movie set containing $|M|$ movies, respectively. The user-movie rating matrix $Y \in \mathbb{R}^{|U|\times|M|}$ is defined in accordance with the rating behaviors from users to movies, and the element y_{u_i,m_j} is a rating value given from user u_i to movie m_j. In addition, the heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is comprised of heterogeneous nodes and undirected edges, where $\mathcal{V}$ consists of users, movies, and review entities, and $\mathcal{E}$ is the set of edges connecting users/movies and review entities. Here, we use $S = \{s_1, s_2, ...\}$ to represent the set of review entities in the graph $\mathcal{G}$, and $\mathcal{V} = \mathcal{U} \cup \mathcal{M} \cup S$. We use A to denote the adjacency matrix of the graph $\mathcal{G}$, where $A_{i,j} = 1$ if $(i,j) \in \mathcal{E}$ and $A_{i,j} = 0$ otherwise.

Given the user-movie rating matrix Y and the heterogeneous graph $\mathcal{G}$, we aim to predict the ratings between users and movies that have not interacted before.

2.2 Heterogeneous Graph Construction

We construct a heterogeneous graph containing users, movies, and their corresponding review entities. For review entities, we adopt the entity linking method

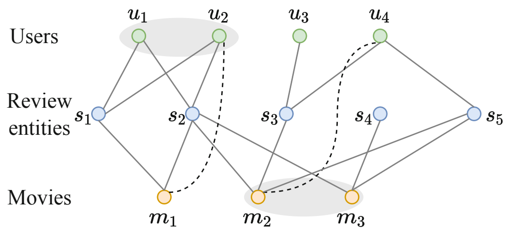


Fig. 2. Example of a review-based heterogeneous KG. Solid lines denote the real connections in the build graph, and dashed lines denote the interactions between users and movies. Grey circles denote similar users and movies discovered through connections to review entities.

[11] to find entities of reviews and each entity as a node in the heterogeneous graph. In review-based RSs, users and items can be represented by their corresponding reviews information [14,15]. Therefore, both users and movies can be linked with their review entities, as shown in Fig. 2. In the figure, we can see that since both u_1 and u_2 are linked to s_1 and s_2, and u_2 has watched movie m_1, we can recommend m_1 to user u_1 on the basis of the similar preferences of u_1 and u_2. Moreover, the multi-hop propagation mechanism can capture the connectivity lines $u_1 \rightarrow s_1 \rightarrow m_1$ and $u_1 \rightarrow s_2 \rightarrow m_1$ in the graph, and the lines reflect the relationship between user u_1 and movie m_1. Similarly, we can also recommend m_2 to u_3 because of the similar properties of m_2 and m_3 and the link propagation $u_3 \rightarrow s_3 \rightarrow m_2$.

2.3 The Proposed Framework

We designed a multi-task learning framework as shown in Fig. 3, which jointly learns the graph link prediction task and rating prediction task to predict the accurate ratings. The proposed framework consists of a graph attention learning module, multi-hop propagation module, and mutual learning module. The graph attention learning module computes the weights of edges in the graph by considering the contributions of review entities to their connected users/movies. We use lines with different thicknesses to represent different attention values, and the larger the value, the thicker the line. The multi-hop propagation module recursively propagates the node embeddings from their neighbors on the basis of the weighted KG. The mutual learning module seamlessly combines the graph link prediction task and the recommendation task to provide accurate ratings.

Graph Attention Learning Module. Given the heterogeneous graph $\mathcal{G}$, we represent the nodes in the graph as vectors by using a graph embedding layer. For a node of user u_i in the graph, the corresponding d-dimensional embedding can be represented by $\mathbf{e}_{u_i} \in \mathbb{R}^d$. Similarly, we use $\mathbf{e}_{m_j} \in \mathbb{R}^d$ to represent the embedding of a movie m_j in the graph. For a review entity s_r, we use $\mathbf{s}_r \in \mathbb{R}^d$ to represent its embedding vector.

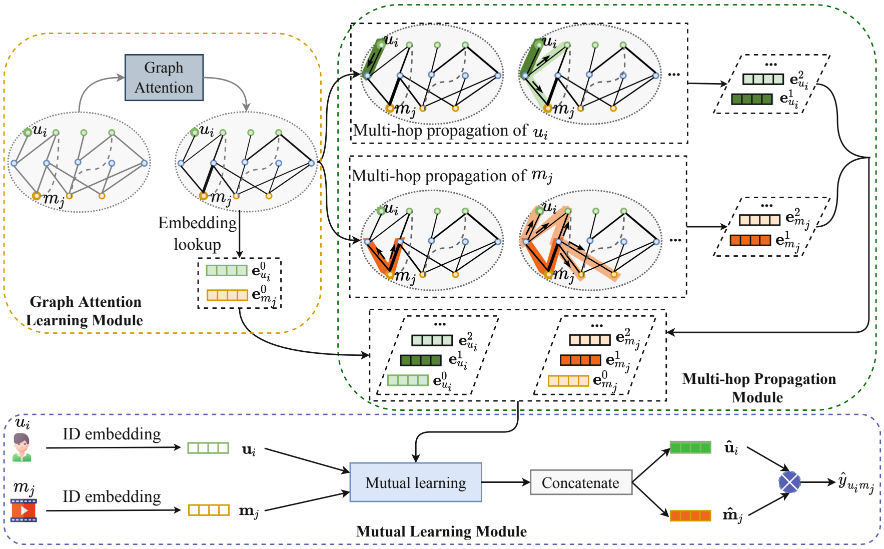


Fig. 3. Proposed JPKG framework.

We adopt the attention mechanism to learn the contributions of review entities to users/movies in the heterogeneous graph. The input of this module is the graph embeddings generated by mapping one-hot vectors through a fully-connected neural network. Given an embedding vector $\mathbf{e}_{u_i}$ of user u_i, and the embedding vector $\mathbf{s}_r$ of the r-th review entity linked with user u_i, the attention values between the user node and its neighbor can be calculated through this module. Specifically, the query vector of $\mathbf{e}_{u_i}$ can be formulated as follows:

$$\mathbf{q}_{u_i} = ReLU(\mathbf{W}_q \mathbf{e}_{u_i}), \tag{1}$$

where $\mathbf{W}_q \in \mathbb{R}^{l \times d}$ is a matrix to project the user node from the d-dimension entity space into the l-dimension query space, and $ReLU(\cdot)$ [18] is a rectified linear unit.

$$\mathbf{k}_{s_r} = ReLU(\mathbf{W}_k \mathbf{s}_r), \tag{2}$$

where $\mathbf{W}_k \in \mathbb{R}^{l \times d}$ is a matrix to transform the review entity into the key-space.

On the basis of the two aforementioned equations, we compute the attention score between user u_i and its linked review entity s_r as follows:

$$a(u_i, s_r) = \mathbf{q}^T_{u_i} \mathbf{k}_{s_r}. \tag{3}$$

We normalize the attention scores of all the neighbors corresponding to the user u_i by using the softmax function:

$$a(u_i, s_r) = \frac{exp(a(u_i, s_r))}{\sum_{s_{r'} \in \mathcal{N}_{u_i}} exp(a(u_i, s_{r'}))}, \tag{4}$$

where $\mathcal{N}_{u_i}$ is the set of review entities linked to user u_i.

We compute the hidden representation of user u_i on the basis of its neighbors:

$$\mathbf{h}_{u_i} = \sum_{s_r \in N_{u_i}} a(u_i, s_r)\mathbf{s}_r, \tag{5}$$

where $\mathbf{h}_{u_i} \in \mathbb{R}^d$. The hidden representation of s_r can be calculated as follows:

$$\mathbf{h}_{s_r} = ReLU(\mathbf{W}\mathbf{s}_r), \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{l \times d}$ is the matrix for projecting the review entity $\mathbf{s}_r$ into the same hidden space with $\mathbf{h}_{u_i}$.

The probability of a link between the user u_i and a review entity s_r can be computed as follows:

$$p(u_i, s_r) = \sigma(\mathbf{h}_{u_i}^T \mathbf{h}_{s_r}), \tag{7}$$

where $\sigma(\cdot)$ is the sigmoid function. Similarly, the probability of the link connecting movie j and review entity s_r can be calculated by the aforementioned Eqs. (1)–(7), denoted by $p(m_j, s_r)$.

We update weight matrices in this module by optimizing the cross-entropy loss function as follows:

$$\mathcal{L}_G = \mathcal{L}_{GU} + \mathcal{L}_{GM}, \tag{8}$$

where $\mathcal{L}_{GU}$ and $\mathcal{L}_{GM}$ are the loss functions for user-centric and movie-centric link prediction, respectively, and each of them can be formulated as:

$$\begin{array}{l} \mathcal{L}_{GU} = -\sum_{(u_i,s_r)\in\mathcal{G}} A_{u_i,s_r} \log p(u_i, s_r) + (1 - A_{u_i,s_r}) \log(1 - p(u_i, s_r)) \\ \mathcal{L}_{GM} = -\sum_{(m_j,s_r)\in\mathcal{G}} A_{m_j,s_r} \log p(m_j, s_r) + (1 - A_{m_j,s_r}) \log(1 - p(m_j, s_r)) \end{array}, \tag{9}$$

where the symbol $A_{\cdot,\cdot}$ denotes a value in the adjacency matrix.

Multi-hop Propagation Module. To compute the effect of multi-hop neighbors on a user/movie, we recursively propagate the embeddings along the connecting lines centered on the user/movie. Taking $m_1 \rightarrow s_1 \rightarrow u_1$ and $m_1 \rightarrow s_2 \rightarrow u_1$ as an example, in the one-hop propagation, movie m_1 and user u_1 take s_1 and s_2 as their attributes to enrich the representations, and in the two-hop propagation, m_1 and u_1 use the embedding information of each other to further enrich their feature representations.

Considering a user u_i in the graph, we use $\mathcal{N}_{u_i}$ to denote a set of neighbors centered around user u_i. The neighbor embeddings of user u_i can be represented by $\mathbf{e}_{\mathcal{N}_{u_i}}$, and

$$\mathbf{e}_{\mathcal{N}_{u_i}} = \sum_{s_r \in \mathcal{N}_{u_i}} a(u_i, s_r)\mathbf{s}_r, \tag{10}$$

where $a(u_i, s_r)$ denotes the attention weights from a review entity s_r linked to user u_i, indicating the contribution from s_r to u_i.

We leverage the method proposed in [3] to aggregate the embeddings of users/movies and their neighbor embeddings. Given the embedding $\mathbf{e}_{u_i}$ of user u_i

and its neighbor embeddings $\mathbf{e}_{\mathcal{N}_{u_i}}$, the aggregation operation can be formulated as:

$$f = LeakyReLU(\mathbf{W}_1(\mathbf{e}_{u_i} + \mathbf{e}_{\mathcal{N}_{u_i}})) + LeakyReLU(\mathbf{W}_2(\mathbf{e}_{u_i} \odot \mathbf{e}_{\mathcal{N}_{u_i}})), \tag{11}$$

where $\mathbf{W}_1$, $\mathbf{W}_2 \in \mathbb{R}^{d \times d}$ are the trainable matrices, and $\odot$ indicates the element-wise product.

For the multi-hop propagation, we recursively propagate information from multi-hop distances to users/movies by stacking multiple aggregation layers. In the t-th aggregation layer, the embedding of u_i can be defined as:

$$\mathbf{e}_{u_i}^{t} = f(\mathbf{e}_{u_i}^{t-1}, \mathbf{e}_{\mathcal{N}_{u_i}}^{t-1}), \tag{12}$$

where the embedding of $\mathcal{N}_{u_i}$ in the $(t-1)$-th aggregation layer is calculated as follows,

$$\mathbf{e}_{\mathcal{N}_{u_i}}^{t-1} = \sum_{s_r \in \mathcal{N}_{u_i}} a(u_i, s_r)\mathbf{s}_r^{t-1}, \tag{13}$$

where $\mathbf{s}_r^{t-1}$ is the embedding of review entity s_r generated from the previous propagation layers. Similarly, the multi-hop propagation embedding of m_j is represented as $\mathbf{e}_{m_j}^{t}$. Note that when $t = 0$, vectors $\mathbf{e}_{u_i}^{0} = \mathbf{h}_{u_i}$ and $\mathbf{e}_{m_j}^{0} = \mathbf{h}_{m_j}$.

For user u_i and movie m_j, the corresponding outputs generated by $(t+1)$ aggregation layers can be gathered by $\{\mathbf{e}_{u_i}^{0}, \mathbf{e}_{u_i}^{1}, ..., \mathbf{e}_{u_i}^{t}\}$ and $\{\mathbf{e}_{m_j}^{0}, \mathbf{e}_{m_j}^{1}, ..., \mathbf{e}_{m_j}^{t}\}$, respectively.

Mutual Learning Module. In this module, we jointly learn the propagation embedding and the corresponding ID embedding of each user and movie to complete the information exchange from two different kinds of latent features.

We describe the mutual learning operation by introducing multiple interaction layers between the ID embedding $\mathbf{u}_i \in \mathbb{R}^d$ of user u_i and the corresponding t-hop propagation embeddings $\{\mathbf{e}_{u_i}^{0}, \mathbf{e}_{u_i}^{1}, ..., \mathbf{e}_{u_i}^{t}\}$. In the n-th mutual learning layer, we build $d \times d$ pairwise interactions between them as follows:

$$\mathbf{C}^n = \mathbf{u}_i \left(\mathbf{e}_{u_i}^{n}\right)^\top = \begin{bmatrix} u_{i1} e_{u_i 1}^{n} & \cdots & u_{id}^{l} e_{u_i 1}^{n} \\ \cdots & & \cdots \\ u_{i1} e_{u_i d}^{n} & \cdots & u_{id} e_{u_i d}^{n} \end{bmatrix}, \tag{14}$$

where $\mathbf{C}^n \in \mathbb{R}^{d \times d}$ is the interaction matrix of $\mathbf{u}_i$ and $\mathbf{e}_{u_i}^{n}$ in the n-th layer, and $n \leq (t+1)$. The ID embedding of u_i in the n-th layer is generated as follows:

$$\mathbf{u}_i^n = \mathbf{C}^n \mathbf{w}_{ue} + (\mathbf{C}^n)^\top \mathbf{w}_{eu} + \mathbf{b} \tag{15}$$

where the vectors $\mathbf{w}_{ue} \in \mathbb{R}^d$ and $\mathbf{w}_{eu} \in \mathbb{R}^d$ denote the trainable projection weights for mapping $\mathbf{C}^n$ to the ID embedding space, and $\mathbf{b} \in \mathbb{R}^d$ is the trainable bias.

We concatenate $(t+1)$ ID embeddings corresponding to u_i as one vector, and then compute the final representation of u_i by using a linear projection:

$$\hat{\mathbf{u}}_i = \mathbf{W}' concatenate(\mathbf{u}_i^1, \mathbf{u}_i^2, ..., \mathbf{u}_i^n, ...), \tag{16}$$

where $\mathbf{W}' \in \mathbb{R}^{d\times(t+1)*d}$ is the trainable projection matrix. Similarly, the embedding of movie m_j can be represented as $\hat{\mathbf{m}}_j$. The final ratings of user u_i to movie m_j is calculated as:

$$\hat{y}_{u_i m_j} = \hat{\mathbf{u}}_i^\top \hat{\mathbf{m}}_j. \tag{17}$$

Optimization. To optimize the proposed model, the entire loss function is defined as follows:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_G + \mathcal{L}_{RS} + \mathcal{L}_{REG} \\ &= \lambda_1 \mathcal{L}_G + \sum_{u_i \in U, m_j \in M} \mathcal{J}\left(\hat{y}_{u_i,m_j}, y_{u_i,m_j}\right) + \lambda_2 \|\mathbf{W}\|_2^2,\end{aligned} \tag{18}$$

where $\mathcal{L}_G$ is the loss function of the graph link prediction task defined in Eq. 8, $\mathcal{L}_{RS}$ is the loss function of the rating prediction task, and $\mathcal{L}_{REG}$ is the regularization term. The symbol $\mathcal{J}(*)$ denotes the mean square error (MSE) function. We use λ_1 and λ_2 as the learning rate parameters to balance the loss.

3 Experiments

3.1 Datasets

We evaluated our model on two publicly available real-world movie datasets: IMDb and Amazon-movie.

- **IMDb** dataset. The dataset was published by a related work JMARS [16], which includes ratings and reviews information from users to movies, and the ratings are in the range of $[0, 10]$.
- **Amazon-movie** dataset. This dataset belongs to the "Amazon product data"[1], which has been widely used to evaluate review rating prediction works [14,17]. The ratings from users to movies are in the range of $[0, 5]$.

To analyze the impacts of different sparse data on recommendation performances, we filtered each dataset into eight different core versions ranging from 3-core to 10-core on the basis of the minimum number of reviews from users. For example, 3-core means each user has at least three reviews in the dataset. We removed the duplicate edges in each graph. The statistics of datasets are illustrated in Table 1.

3.2 Experimental Settings

Baselines. To evaluate the effectiveness of the proposed model, we chose three highly-relevant state-of-the-art works: rating-based matrix factorization methods, review-based neural networks, and knowledge-based mutual learning methods as our baselines.

[1] http://jmcauley.ucsd.edu/data/amazon/.

Table 1. Statistics of the two datasets with different sparsities

Dataset		3-core	4-core	5-core	6-core	7-core	8-core	9-core	10-core
IMDb	# users	1,833	1,648	1,504	1,393	1,318	1,237	1,160	1,095
	# movies	4,663	4,663	4,663	4,663	4,663	4,663	4,663	4,663
	# ratings	126K	125K	125K	124K	124K	123K	123K	122K
	# review entities	69K	69K	68K	68K	68K	68K	68K	68K
	# nodes	76K	75K	75K	75K	74K	74K	74K	74K
	# edges	1,008K	1,004K	1,000K	999K	995K	992K	988K	984K
Amazon-movie	# users	158K	123K	93K	68K	53K	43K	35K	29K
	# movies	59K	59K	58K	58K	58K	57K	57K	56K
	# ratings	1,448K	1,343K	1,223K	1,101K	1,009K	936K	876K	825K
	#review entities	190K	189K	187K	185K	182K	180K	178K	176K
	# nodes	408K	371K	340K	312K	294K	281K	271K	263K
	# edges	6,248K	6,058K	5,810K	5,471K	5,211K	4,992K	4,802K	4,641K

– Probabilistic matrix factorization (PMF). PMF is a matrix factorization model that learns the latent representations of users and items from a rating matrix to provide accurate recommendations [19].
– Generalized matrix factorization (GMF). GMF is a generalized version of matrix factorization (MF) [20] that uses a nonlinear layer to project the latent vectors of users and items into the same space, and models the interactions between users and items on the basis of their projected vectors [21].
– Multi-task feature learning for KG enhanced recommendation (MKR). This method treats items as head entities of the KG and learns latent vectors of items by mutual learning between an RS task and KGE task [22].
– Deep cooperative neural networks (DeepCoNN). DeepCoNN is a review-based neural network that adopts a convolution-based parallel structure framework to extract the latent representations of users and items from their corresponding reviews [23].
– Transformational neural networks (TransNets). This method is also a review-based neural network inspired by DeepCoNN that introduces a transform layer in a parallel neural network to transform reviews of users and items into the same representation space for recommendation [24].

Evaluation Metric. To measure the performances of all the tested models, we adopt root-mean-square error (RMSE) as the evaluation metric. Given a ground truth rating y_{u_i,m_j} rated by user u_i for movie m_j and its corresponding predicted rating $\widehat{y}_{u_i,m_j}$, the RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{u_i,m_j} \left(\widehat{y}_{u_i,m_j} - y_{u_i,m_j}\right)^2}, \tag{19}$$

where N indicates the number of user ratings for movies.

Table 2. Overall performance comparison. Best results are highlighted in bold.

		3-core	4-core	5-core	6-core	7-core	8-core	9-core	10-core
IMDb	PMF	1.837	1.812	1.812	1.84	1.767	1.789	1.77	1.758
	GMF	1.848	1.826	1.826	1.862	1.782	1.809	1.789	1.771
	MKR	1.827	1.819	1.804	1.818	1.806	1.806	1.806	1.802
	DeepCoNN	1.813	1.815	1.778	1.809	1.774	1.751	1.787	1.776
	TransNets	1.814	1.816	**1.763**	1.793	1.777	1.75	1.775	1.766
	JPKG	**1.772**	**1.775**	1.773	**1.762**	**1.763**	**1.746**	**1.736**	**1.748**
Amazon-movie	PMF	1.131	1.088	1.081	1.072	1.084	1.085	1.092	1.097
	GMF	1.175	1.172	1.172	1.167	1.160	1.160	1.154	1.148
	MKR	1.100	1.097	1.088	1.082	1.072	1.069	1.066	1.060
	DeepCoNN	1.045	1.034	1.024	1.026	1.018	1.017	1.020	1.016
	TransNets	1.047	1.042	1.030	1.041	1.022	1.023	1.014	1.021
	JPKG	**1.031**	**1.029**	**1.021**	**1.018**	**1.011**	**1.006**	**1.001**	**0.997**

Parameter Settings. We randomly selected 80%, 10%, and 10% of samples as the training, validation, and test sets, respectively. We set the learning rates of the recommendation task and graph linking prediction task to 2.0×10^{-4} and 8.0×10^{-6}, respectively. The values of λ_1 and λ_2 were fixed to 0.04 and 1.0×10^{-6}, respectively. The number of propagation layers in the multi-hop propagation module was set to 3. The dimensions of both ID embeddings and graph node embeddings were set to 16. The batch size in the training processes for the recommendation task and graph link prediction task were set to 64 and 1024, respectively. The training interval was set to 4, which means that we repeatedly train recommendation task 4 times before training the graph link prediction task once in each epoch.

3.3 Experimental Results

We report the experimental results of our proposed model and those of the baselines datasets with various sparsities in Table 2. We can see that the proposed JPKG outperforms the other models in most cases. In particular, it achieves the best performance on the Amazon-movie dataset with all the sparsities and on the IMDb dataset except the 5-core sparsity. In general, review-based methods perform better than rating-based methods, indicating that review information can reflect user preferences and item properties that do not exist in ratings. Moreover, as the data becomes denser, the improvement of the review-based method becomes smaller. However, the improvements of RMSEs for our method on both the IMDb and Amazon-movie datasets with various sparsities remains at about 2% and 9%, respectively, which demonstrates that our proposed method is effective for sparse datasets and maintains its effectiveness consistently as the datasets become denser.

Table 3. RMSE results of ablation study.

	IMDb		Amazon-movie	
	3-core	10-core	3-core	10-core
JPKG-ML	1.811	1.783	1.042	1.018
JPKG-PF	1.787	1.769	1.040	1.017
JPKG-attn	1.777	1.753	1.037	1.004
JPKG	**1.772**	**1.748**	**1.031**	**0.997**

3.4 Ablation Study

To investigate the effectiveness of the three modules in our work, we report the experimental results from two perspectives based on ablation studies: recommendation accuracy and convergence.

For the ablation methods, we first disabled the mutual learning layers and aggregated the multi-hop propagation features directly as the final representations of users/movies, termed JPKG-ML. We then disabled the multi-hop propagation module and jointly learned the attention-based node representations and ID embeddings to predict ratings, termed JPKG-PF. Finally, we disabled the attention mechanism on the graph and treated the contributions of all neighbors of a node as the same, termed JPKG-attn.

Recommendation Accuracy. Table 3 shows the RMSE results of the ablation methods and JPKG on the IMDb and Amazon-movie datasets with the 3-core and 10-core sparsities, respectively. We can see that disabling any of the three key modules degrades the performance of the model. We can also see that JPKG-ML underperforms other methods, which indicates the mutual learning module plays a more important role than the other two modules. This finding also reveals an empirical fact that directly using graph embeddings for recommendation may introduce noise and mislead the final recommendation. Furthermore, JPKG-attn performs better than JPKG-PF, which verifies that removing the multi-hop propagation module can have a more significant effect than removing the attention module on recommendation results. One possible reason is that learning multi-hop propagation features can substantially improve the quality of representation learning.

Convergence. We investigated the influences of the key modules on our model by observing the convergence of ablation methods on the IMDb and Amazon-movie datasets, and the results are presented in Fig. 4. We reported the RMSE results on the validation data by varying the training epochs to illustrate the convergence. Note that we adopted the early-stopping strategy to obtain the final experimental results. We can see that the convergence speed of JPKG is faster than those of JPKG-attn, JPKG-ML, and JPKG-PF. Moreover, JPKG can

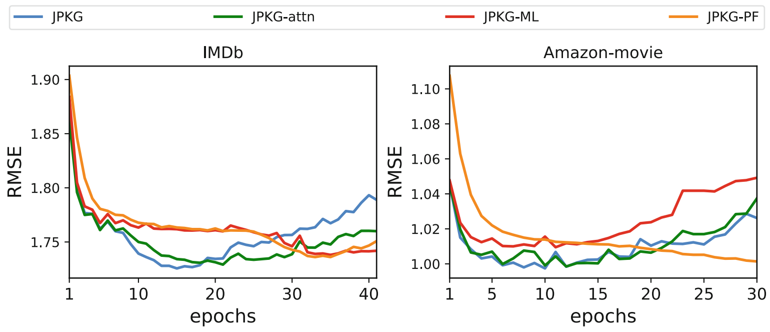


Fig. 4. Convergence comparisons on the two datasets among three ablation models and JPKG.

reach a smaller value than the other three ablation methods on the two datasets. Note that JPKG-PF needs more epochs for the convergence, which means that adopting multi-hop propagation can enable us to speed up the convergence. We can also see that JPKG-ML cannot converge to a relatively small loss on the two datasets. The aforementioned results illustrate the necessity of the three key modules in our model.

4 Conclusion

In this paper, we proposed JPKG, a multi-task framework that jointly learns multi-hop propagation features on a KG for movie recommendations. JPKG overcomes the limitation of insufficient user connections in current KG-based recommendations by integrating review entities, users, and movies into a heterogeneous graph. The attention learning module and multi-hop propagation module of JPKG achieve attention-based multi-hop propagation feature learning by recursively calculating the different contributions of neighbors on the graph. The mutual learning module of JPKG combines the entity embeddings learned from the two aforementioned modules to help provide more accurate recommendations. The experimental results on two real-world datasets demonstrate the effectiveness of our proposed model.

For future work, we will focus on providing explainable recommendations on the basis of the current work. Furthermore, we will explore other methods that can enhance the user preference mining ability on KGs.

Acknowledgements. This work was supported in part by Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Numbers JP18H03242, JP18H03342, JP19H01138, and JP21K17868.

References

1. Wang, H., Zhang, F., Wang, J., et al.: RippleNet: propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 417–426 (2018)
2. Tai, C.Y., Huang, L.Y., Huang, C.K., et al.: User-centric path reasoning towards explainable recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 879–889 (2021)
3. Wang, X., He, X., Cao, Y., et al.: KGAT: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 950–958 (2019)
4. Zhang, F., Yuan, N.J., Lian, D., et al.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–362 (2016)
5. Wang, H., Zhang, F., Xie, X., et al.: DKN: deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide Web Conference, pp. 1835–1844 (2018)
6. Ai, Q., Azizi, V., Chen, X., et al.: Learning heterogeneous knowledge base embeddings for explainable recommendation. Algorithms **11**(9), 137 (2018)
7. Fu, Z., Xian, Y., Gao, R., et al.: Fairness-aware explainable recommendation over knowledge graphs. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 69–78 (2020)
8. Brams, A.H., Jakobsen, A.L., Jendal, T.E., et al.: MindReader: recommendation over knowledge graph entities with explicit user ratings. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2975–2982 (2020)
9. Xian, Y., Fu, Z., Muthukrishnan, S., et al.: Reinforcement knowledge graph reasoning for explainable recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 285–294 (2019)
10. Yu, X., Ren, X., Sun, Y., et al.: Personalized entity recommendation: a heterogeneous information network approach. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 283–292 (2014)
11. Ratinov, L., Roth, D., Downey, D., et al.: Local and global algorithms for disambiguation to wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 1375–1384 (2011)
12. Bordes, A., Usunier, N., Garcia-Duran, A., et al.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
13. Lin, Y., Liu, Z., Sun, M., et al.: Learning entity and relation embeddings for knowledge graph completion. In: 29th AAAI Conference on Artificial Intelligence (2015)
14. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM Conference on Recommender Systems, pp. 165–172 (2013)
15. Liu, D., Li, J., Du, B., et al.: DAML: dual attention mutual learning between ratings and reviews for item recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 344–352 (2019)

16. Diao, Q., Qiu, M., Wu, C.Y., et al.: Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 193–202 (2014)
17. Kim, D., Park, C., Oh, J., et al.: Convolutional matrix factorization for document context-aware recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 233–240 (2016)
18. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: ICML (2010)
19. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, vol. 20 (2007)
20. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
21. He, X., Liao, L., Zhang, H., et al.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
22. Wang, H., Zhang, F., Zhao, M., et al.: Multi-task feature learning for knowledge graph enhanced recommendation. In: The World Wide Web Conference, pp. 2000–2010 (2019)
23. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the 10th ACM International Conference on Web Search and Data Mining, pp. 425–434 (2017)
24. Catherine, R., Cohen, W.: TransNets: learning to transform for recommendation. In: Proceedings of the 11th ACM Conference on Recommender Systems, pp. 288–296 (2017)
25. Chen, C., Zhang, M., Liu, Y., et al.: Neural attentional rating regression with review-level explanations. In: Proceedings of the World Wide Web Conference, pp. 1583–1592 (2018)