KRec-C2: A Knowledge Graph Enhanced Recommendation with Context Awareness and Contrastive Learning

Yingtao Peng[1], Zhendong Zhao[2], Aishan Maoliniyazi[1], and Xiaofeng Meng[1(✉)]

[1] School of Information, Renmin University of China, Beijing, China
{yingtaopeng,aishan,xfmeng}@ruc.edu.cn
[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
zhaozhendong@iie.ac.cn

Abstract. Knowledge graph (KG) has been widely utilized in recommendation system to its rich semantic information. There are two main challenges in real-world applications: high-quality knowledge graphs and modeling user-item relationships. However, existing methods try to solve the above challenges by adopting unified relational rules and simple node aggregation, which cannot cope with complex structured graph data. In this paper, we propose a **K**nowledge graph enhanced **Rec**ommendation with **C**ontext awareness and **C**ontrastive learning (KRec-C2) to overcome the issue. Specifically, we design an category-level contrastive learning module to model underlying node relationships from noisy real-world graph data. Furthermore, we propose a sequential context-based information aggregation module to accurately learn item-level relation features from a knowledge graph. Extensive experiments conducted on three real-world datasets demonstrate the superiority of our KRec-C2 model over existing state-of-the-art methods.

Keywords: Recommendation System · Knowledge Graph · Contrastive Learning · Context Awareness

1 Introduction

Nowadays, we are suffering from a severe information overload issue in all fields as never before. The recommendation system (RS), aim to filter relevant information from massive amounts of data, which makes it easy for the user to obtain high-value information from modern big data platform such as e-commerce [16,19], multimedia [33], and online news [14]. As one of the most important technologies in the current big data era, the recommendation system brings great convenience to users, and attracts great attention from academia and industry.

Among the existing research from recommendation communities, the Collaborative Filtering (CF) [9,13,15] framework assumes that users with similar behaviors may have similar interests, and is the most commonly used technique for recommendation systems to predict user preferences. Arguably, most existing CF-based models can be roughly divided into two categories: (1) The traditional

X. Wang et al. (Eds.): DASFAA 2023, LNCS 13944, pp. 3–20, 2023.
https://doi.org/10.1007/978-3-031-30672-3_1

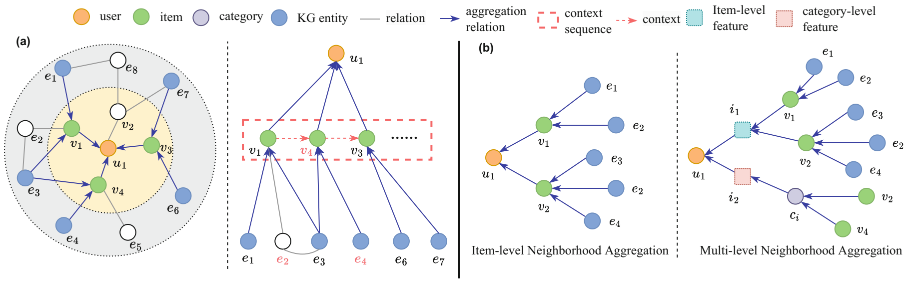

Fig. 1. (a) is an example of user representation on context awareness, where an arrow is the contextual relation within a user's interaction sequence. (b) is an example of the category-level signal fusion schemes, where i_2 represent the category-level signal.

CF approaches. These models find a certain similarity measure through the behavior of groups and make recommendations for users [13]. (2) The neural network-based CF approaches. These methods represent each user and item as a low-dimensional vector, and then evaluate the degree of association according to defined vector operations [4]. For example, some works obtain higher-order feature vectors by combining the non-linearity of a neural network to improve the performance of vector representation [9,10]. In contrast, we design a novel context-aware fusion network to capture the user's representation in this paper.

In the real world, the user's ratings are mostly empty, which leads to the extremely sparse interactive data we collect. In addition, due to the limited attention of the user, an item without interaction may also be a great favorite. However, because the interaction data is sparse in real-world situations and existing neural network-based CF models are over-reliant on the interaction data, these approaches perform unsatisfactorily in universal scenarios [36]. To address this problem, some research works [2,25] introduce knowledge graph (KG) as auxiliary information in recommender systems, which can better integrate different information into the overall network to learn better embedding representations. The KG allows the developer to model relationships between users and items with the entity and relationship information and enhance the semantic association of nodes by embedding [22,36].

Although the KG-based RS has been applied in several areas, its effectiveness remained over-reliant on the quality of knowledge graphs. Mostly, the KG is often noisy and redundancy in real-world scenes, which contains topic-irrelevant connections between entities and longtail entity distribution, leading to poor performance of the model on feature learning [17,21]. Even though existing self-supervised learning-based RS methods [32,37] try to alleviate the issue by graph-based augmentation techniques (such as edge dropout, node dropout and subgraph sampling) [32,34], there is fall short in three factors: i) **Ignoring the context feature**. The context feature is a relationship dependence between items within a interaction sequence [20] in the recommendation system, which helps us to obtain potential relationships between the interaction items.

Unfortunately, existing KG-based works do not account for this crucial information, leading to the insufficient representation performance of user and item. Taking the right of Fig. 1 as an example, arrows are the potential contextual relations within a user's clicked sequence. On the contrary, ignoring contextual features limits the discovery of contextual intention in the left Fig. 1 (a). ii) **Overlooking the category-level signal**. We define a category-level signal as the common characteristics of similar items in the same catalog, such as function, classification, etc., which provide a solution to capture the category-level features for similar items. Regrettably, existing KG-based models focus on the node-based aggregation scheme [26] at the item level, leading to missing the category-level feature. As shown in the right side of Fig. 1 (b), the representation of u_1 fusions the feature of item i_1 and the category-level feature i_2. By comparison, u_i cannot capture the category-level feature on the left side of Fig. 1 (b). iii) **Shortage of the self-supervised signal**. The lack of self-supervised signals leads to the insufficient ability of the model to deal with noise in a real-world scenario. Self-supervised learning generally learns a feature representation for downstream tasks by extracting information and transferable knowledge. Existing KG-based models with self-supervised learning ignore the significant category-level view, leading to a lower ability to alleviate the sparse data and noisy issue.

To solve the above problems, we propose a knowledge graph enhanced recommendation with context awareness and contrastive learning (KRec-C2). Our model consists of three components: (1) **Item-level context awareness module (ICAM)**. Each user-item interaction is enriched with the underlying intents for the user. Although we can express these items as vectors, their contextual semantics within a sequence are opaque to understand. Hence, we combine the intent and the context based on node aggregation representation. Technically, context awareness is an attentive design of relation embeddings, where the important intent is assigned with a larger weight factor. (2) **Category-level intents fusion module (CIFM)**. The node-based aggregation scheme focuses on the receptive field of the item to obtain an embedded representation for the itemself. In contrast, CIFM provides a solution to capture the common intrinsic properties on the category level between items. To overcome the technical challenge, we first obtain the embedded vector for item-self with node-based aggregation and capture the intrinsic features on the category level for items by a higher-order aggregation scheme. Then we fusion each signal into a representation vector. (3) **Contrastive learning module (CLM)**. We carefully design a contrastive learning (CL) optimization objective for capturing the relevance of the category view to alleviate the noise and sparse issues. Specifically, contrastive learning takes inspiration from the KG-based representation and self-supervised data augmentation to guide our model in refining the category-level representations with the contrastive view on the category level. The major novelty lies in achieving the optimization objective with mutual information maximization at the training stage.

We summarize the contributions of this work as: (1) We propose a knowledge graph enhanced recommendation with context awareness and contrastive learning (KRec-C2) to enhance the recommendation performance. (2) We design three novel modules in the KRec-C2 framework for KG-based recommendation, including ICAM, CIFM, and CLM. Based on these modules, context features and the category-level signal have been effectively utilized and integrated into the KG-based recommender. (3) Extensive experiments conducted on three real-world datasets demonstrate the effectiveness of our proposed approach.

2 Preliminaries

We first define our recommendation task, and then present necessary notations and structural data, such as user-item interactions and KGs.

Task Description. Assume that we have the interaction data $\mathbf{Y}$ and the KG $\mathcal{G}$. Our goal is to learn the probability $\hat{y}_{uv} = \sigma(\mathbf{u} \cdot \mathbf{v})$ that a user u adopt a candidate item v, where σ is activation function, $\mathbf{u}$ and $\mathbf{v}$ are vectors representation of an user u and a candidate item v.

Interaction Data. In our recommendation task, we focus on the user-item interactions (e.g., browse, click, purchase and comment, etc.). Here we let $\mathcal{U} = \{u_1, u_2, u_3, \cdots\}$ be a set of users and $\mathcal{V} = \{v_1, v_2, v_3, \cdots\}$ be a set of items. We define $\mathbf{Y} = \{y_{uv} | u \in \mathcal{U}, v \in \mathcal{V}\}$ as the user-item interaction matrix.

Knowledge Graph. In application scenes of the RS, we can obtain the auxiliary property information of items (e.g., categories, and attributes), which make up various real-world entities and numerous relations in KG. We organize knowledge graph $\mathcal{G}$ in the form of triple, let $\mathcal{G} = \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where its relations are composed of (item-property-item) triples, and denote that a relation r from head node h to tail node t. For example, (Titanic, love movie, Flipped) describes that Titanic and Flipped both belong to love movies.

3 Methodology

In this section, we first introduce our model framework and then discuss each module of KRec-C2 in detail.

3.1 Framework

The framework of our model is illustrated in Fig. 2, where we innovatively model context, category-level signals, and self-supervised features by three modules to improve the recommendation effect. KRec-C2 inputs interaction data and KG, and outputs the probability that user u would interact item v. Specifically, our model consists of three components: (1) ICAM. First, we aggregate neighbor's information of items by KGCN. Then, the context awareness layer captures implicit contextual relationships among the interaction sequence items and predicts the user's potential preferences. (2) CIFM. The category-aware aggregation module propagates embeddings from a category node's neighbors to update its representation. Then, we fuse contextual features and category-level signals as user feature representations during the training process. (3) CLM. We design a

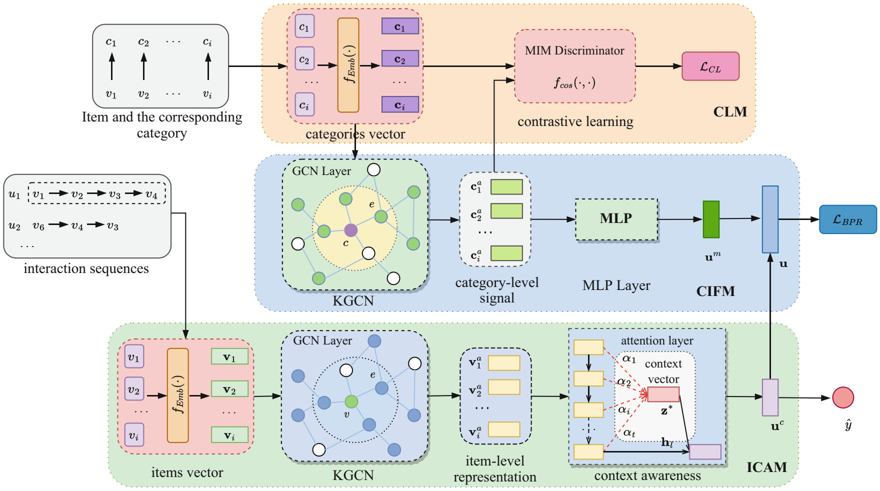

Fig. 2. An overview of the proposed KRec-C2 framework.

contrastive learning module on a category-level view to capture the relevance of the category for alleviating noise and sparse issues. Finally, on the prediction layer, we predict the probability of a user will interact a candidate item based on contextual features and item vector.

3.2 Item-Level Context Awareness Module

Previous GNN-based research only aggregates the information of neighbor nodes [25, 26, 30], while our goal is to capture the contextual relationship between interacted items and predict users' latent intentions. First, we utilize the KGCN layer to obtain item-level aggregation features, which contain the associated information of items in the KG. For example, in the movie's case, the item-level aggregation feature includes various relationships from directors, actors, locations, etc. Then, we can obtain potential intent from the user's interaction sequence. Intuitively, user interactions contain potential user preferences, which are essential to features. It is necessary to characterize users by extracting personalized features to improve the recommendation ability. This idea motivates us to model user-item relationships with context awareness.

Item-Level Aggregation Representation. The neighbor information aggregation captures the local proximity structure and stores representation in each entity to reflect the user's personalized interests better. For example, in the real world, a goods node has many rich relations with neighbor nodes in KG, such as brand, size, color, etc. We use $\mathcal{N}(v) = \{e_1, e_2, \cdots, e_n\}$ to denote the entities' set for the item's neighbors in KG. Then, we learn the current item's high-order aggregation representation utilizing a knowledge-aware graph convolution neural network (KGCN) [25].

$$\mathbf{v}_{\mathcal{N}(v)} = \sum_{e \in \mathcal{N}(v)} \zeta_{r_{v,e}} \mathbf{e}, \tag{1}$$

where $\zeta_{r_{v,e}} = \frac{\exp\left(\tilde{\zeta}_{r_{v,e}}\right)}{\sum_{e \in \mathcal{N}(v)} \exp\left(\tilde{\zeta}_{r_{v,e}}\right)}$, ζ_r is the normalized node-relation score $\tilde{\zeta}_r$. In the receptive field of item v, we capture the neighborhood feature of the item as $\mathbf{v}_{\mathcal{N}(v)}$, where $\mathcal{N}(v)$ is the single-layer receptive domain of the item v. In order to keep the computation pattern of each batch fixed, we first uniformly sample a fixed-size set of neighbor nodes for the item entity. Then we aggregate the item representation $\mathbf{v}$ and its neighbor nodes representation $\mathbf{v}_{\mathcal{N}(v)}$ into a vector. For calculating the feature $\mathbf{v}^a$, our aggregate methods as follows:

$$\mathbf{v}^a = \sigma\left(W \cdot f_{agg}(\cdot) + b\right) \tag{2}$$

where $f_{agg}(\cdot)$ notes the aggregate operator, we implement it with three methods: sum operator $f_{agg}^{sum} = \mathbf{v} + \mathbf{v}_{\mathcal{N}(v)}$, concat operator $f_{agg}^{concat} = concat\left(\mathbf{v}, \mathbf{v}_{\mathcal{N}(v)}\right)$, and neighbor operator $f_{agg}^{neighbor} = \mathbf{v}_{\mathcal{N}(v)}$. W and b denote transformation weight and bias, σ denotes the nonlinear function.

Context Awareness Representation. To capture the user's history interest and predict potential intention, we designed the context-awareness module, composed of a bidirectional gate recurrent unit (Bi-GRU) and a self-attention network based on the output of KGCN. The module inputs a sequence of the user's interaction item, which is the output result of item-level aggregation representation. Our goal is to capture the contextual relationship and output the user's potential intention.

Based on item-level aggregation representation, we obtain the item-level representations $\mathcal{V}_l = \{\mathbf{v}_1^a, \mathbf{v}_2^a, \cdots, \mathbf{v}_l^a\}$ of user's interaction sequence $\{v_1, v_2, \cdots, v_l\}$. In this module, we adopt a Bi-GRU based self-attention network composed of a bidirectional gate recurrent unit layer and a self-attention layer. In i-step of Bi-GRU, our input information are the i-step item-level representation $\mathbf{v}_i^a$ and the hidden representation of $i-1$-step of Bi-GRU, and our output information are an intermediate feature $\mathbf{h}_i \in R^m$, where m denotes the dimension of the hidden representation of Bi-GRU, the formula is as follows:

$$\mathbf{h}_i = f_{BiGRU}\left(\mathbf{h}_{i-1}, \mathbf{v}_i^a; \Theta_{gru}\right), \tag{3}$$

where $\mathbf{h}_i$ denotes intermediate feature of GRU's i-step, Θ_{gru} denotes all the related parameters of the Bi-GRU network.

In the Bi-GRU layer, the last hidden state $\mathbf{h}_i$ denotes the sequential representation of the input sequence. In order to explore the latent relationship of user interaction sequences, we consider that the previous hidden states may play different role for user's latent intention, and then we can capture the user's potential feature by self-attention mechanism. Therefore, we take the hidden embedding vector $H = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_l] \in R^{m\times n}$ as input, and exploit self-attention mechanism to capture the user's contextual interest feature, the formula is as follows:

$$Z^* = V\ softmax\left(\frac{K^T Q}{\sqrt{D}}\right), \tag{4}$$

where $softmax(\cdot)$ denotes the normalization function. $Q = W_q H$, $K = W_k H$, and $V = W_v H$ are mapped vectors by H, and W_q, W_k, W_v are the parameter matrices of the linear mapping. D is the dimension of the input vector H.

Finally, we obtain item's contextual representation $\mathbf{u}^c = [\mathbf{z}^*; \mathbf{h}_l]$ by merging the user's the contextual representation $\mathbf{z}^*$ and the prediction feature $\mathbf{h}_l$.

3.3 Category-Level Intents Fusion Module

Existing methods learn item representation by node-level aggregation scheme, which ignores the category-level signal, resulting in a limited effect. In this section, we aim to capture the category-level signals and incorporate them into user feature representations in the stage of training. Technically, we learn the category node and its neighbor representations as the category-level features, and fuse them with item-level user representations. Based on the above analysis, we model interactions at the category level.

Category-Level Aggregation Representation. Mostly, items in the same category have similar attributes in recommended scenarios. Existing works only capture the single relations with item by the category node such as (Titanic, classification, love movie), missing salient common feature between items in the same catalog (e.g., region, director, etc.). To alleviate this issue, we construct the category-item graph to capture the common feature as a category-level signal.

In this part, we obtain a set of triples $\mathcal{T}$ from the items and their categories, where $\mathcal{T} = \{(c_1, r_1, v_1), (c_2, r_2, v_2), (c_2, r_2, v_3), \cdots\}$, r_i notes relation between item v_i and category c_i, such as (sports, golf, PGA Tour winners). Similar to item-level aggregate representation, we use $\mathcal{N}(c) = (e_1, e_2, \cdots, e_n)$ to denote the set of entities directly connected to the category c. The items' similarities attributes in a same category can regard as the common preferences of users. Similarly, we adopt a knowledge-based graph convolution neural network (KGCN) to capture the category-level representation. Specifically, in a single GCN layer, the formula is as follows:

$$\mathbf{H}^{l+1} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^l W\right), \tag{5}$$

where $\mathbf{H}^{l+1}$ is output feature, σ is a nonlinear activation function. W is the linear transformation matrix parameter. $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is a self connected adjacency matrix of $\mathbf{A}$, we construct adjacency matrix $\mathbf{A}$ by our KG. $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$ is a degree matrix.

Then, in the category's receptive field, we compute the neighborhood representation of the category c as $\mathbf{c}_{\mathcal{N}(c)}$. Similarly, we calculate the category representation $\mathbf{c}$ by sum aggregator $f_{agg}^{sum} = \mathbf{c} + \mathbf{c}_{\mathcal{N}(c)}$, because we found that the sum aggregator performs better, which is specified in Sect. 4.5. The formula is as follows:

$$\mathbf{c}^a = \sigma\left(W \cdot \left(\mathbf{c} + \mathbf{c}_{\mathcal{N}(c)}\right) + b\right), \tag{6}$$

where $\mathbf{c}$ is the category embedding vector. Based on the category graph, we obtain the category-level aggregation representation $\mathbf{c}^a$.

Category signals fusion Layer. We obtain the user's context feature representation $\mathbf{u}^c$ from the ICAM module at Sect. 3.2. From the category-level aggregation representation layer, we obtain the category-level signal vector $\mathbf{c}^a$ of the user's interaction item. To extract feature and align the dimension with $\mathbf{u}^c$, we utilize the MLP layer to acquire the the category-level feature $\mathbf{u}^m$, $\mathbf{u}^m = f_{MLP}\left(\mathbf{c}^a\right)$, where f_{MLP} is a MLP neural network. Then, we fuse the two parts of the features through the connect operation to obtain fusion feature $\mathbf{u}$ in the training stage. The formula is as follows:

$$\mathbf{u} = \sigma\left(concat\left(\mathbf{u}^c, \mathbf{u}^m\right)\right), \tag{7}$$

where σ is a activation function.

3.4 Contrastive Learning Module

In general, knowledge graphs contain redundancy and noise, and low-quality KG affects the performance of recommender systems. Generally, items of the same category include similar user preference information. Therefore, inspired by recent advances in self-supervised learning and contrastive learning techniques [5, 28], we built a contrastive learning module to improve the model's performance. Specifically, we take the aggregated features of category nodes as a view and make them closer to the category-level signal during the training process. Through contrastive learning, we can alleviate the redundancy and noise issues and enhance the robustness of the model.

We use a contrastive learning framework to guide self-supervised learning of class-level features to maximize the mutual information [11] between the category information and the category-level signal. First, the category is one of the essential components of an item's attributes. For an item, we obtain the vector of its category by embedding function $f_{Emb}(\cdot;\cdot)$, $\mathbf{c} = f_{Emb}(c;\Phi)$, where $\mathbf{c}$ denotes the category initialization vector, and Φ is parameters of the network f_{Emb}.

Then, the neighbor nodes of a category contain its fine-grained common characteristic such as function, classification, etc. We fuse highlight attributes as the category aggregation signal by modeling feature correlation to inject the attribute information into the category representation. Technically, we obtain the category-level signal vector $\mathbf{c}_i^a$ through the category-level aggregation representation layer. We treat a category vector $\mathbf{c}_i$ and its category-level signal feature $\mathbf{c}_i^a$ as two different views. More formally, we minimize the associated feature prediction loss by:

$$\mathcal{L}_{CL} = \sum_{c \in \mathcal{C}} -\log \frac{\exp\left(f_{cos}\left(\mathbf{c}_i, \mathbf{c}_i^a\right)/\gamma\right)}{\sum_{c \in \mathcal{C}} \exp\left(f_{cos}\left(\mathbf{c}_i, \mathbf{c}_{i'}\right)/\gamma\right)}, \tag{8}$$

Algorithm 1. KRec-C2 Learning algorithm.

Require:
Interaction matrix $\mathbf{Y}$; knowledge graph $\mathcal{G}$
Ensure:
Model parameters $\boldsymbol{\Theta}$
1: Randomly initialize neural parameters $\boldsymbol{\Theta}$
2: Constructe adjacency matrix of entities $\mathbf{A}_e$ and adjacency matrix of relations $\mathbf{A}_r$ from $\mathcal{G}$
3: **while** An epoch is not end **do**
4: Sample minibatch of interactions from Interaction matrix $\mathbf{Y}$
5: Compute the loss $\mathcal{L}_{BPR}(\boldsymbol{\Theta})$ $(Eq. (10))$
6: Compute the loss $\mathcal{L}_{CL}(\boldsymbol{\Theta})$ $(Eq. (8))$
7: $\mathcal{L}(\boldsymbol{\Theta}) \leftarrow \mathcal{L}_{BPR}(\boldsymbol{\Theta}) + \lambda_1 \mathcal{L}_{CL}(\boldsymbol{\Theta})$
8: Update neural parameters
9: **end while**
10: **return** $\boldsymbol{\Theta}$

where $f_{cos}(\cdot,\cdot)$ is the cosine similarity function here, $i \neq i^{'}$, γ is the hyper-parameter to the temperature in softmax function. $\mathcal{C}$ is set of the category node and $\mathcal{C} = \{c_1, c_2, \cdots, c_n\}$.

3.5 Prediction Module

In prediction layer, we calculate the relevance probability by a user's contextual feature $\mathbf{u}_i^c$ and vectors $\mathbf{v}$ of candidate items as follow: $\hat{y}_{uv} = \sigma(\mathbf{u}_i^c \cdot \mathbf{v})$, where σ is a sigmoid activation function, $\mathbf{u}_i^c$ is obtained from Sect. 3.2, $\mathbf{v}$ notes the embedding representation of items.

3.6 Model Optimization

To optimize the recommendation model, we minimize the following objective function to learn the model parameter by combining the BPR loss [27] and the independence loss:

$$\mathcal{L} = \mathcal{L}_{BPR} + \lambda_1 \mathcal{L}_{CL} + \lambda_2 \|\boldsymbol{\Theta}\|_2^2, \tag{9}$$

where $\boldsymbol{\Theta}$ is the set of model parameters; λ_1 and λ_2 are two hyper-parameters to control the independence loss $\mathcal{L}_{CL}$ and L_2 regularization term, respectively.

In particular, we assumes that observed interactions indicate more user preferences and should be given higher predictive values than unobserved interactions:

$$\mathcal{L}_{BPR} = \sum_{(u,v^+,v^-)\in\mathcal{O}} -\ln\sigma\left(\hat{y}_{u,v^+}^{\mathcal{L}_{BPR}} - \hat{y}_{u,v^-}^{\mathcal{L}_{BPR}}\right), \tag{10}$$

where $\hat{y}_{uv}^{\mathcal{L}_{BPR}} = \sigma(\mathbf{u}\cdot\mathbf{v})$, $O = \{(u, v_+, v_-) \mid (u, v_+) \in \mathcal{R}^+, (u, v_-) \in \mathcal{R}^-\}$ denotes the pairwise training data, $\mathcal{R}^+$ denotes the positive samples, $\mathcal{R}^-$ denotes the negative samples, v_+ is user clicked item, v_- is no click item. $\hat{y}_{u,v}^{\mathcal{L}_{BPR}}$ indicates the calculated score. Furthermore, the parameters $\boldsymbol{\Theta}$ in our model are jointly optimized. The training procedure of our model is illustrated in Algorithm 1.

Table 1. Statistics of the three evaluation datasets.

Dataset	User	Items	Interactions	Entities	Relation	Triplets
Last-FM	1,872	3,846	42,346	18,165	62	15,518
MIND	299,999	47,034	20,000	57,434	62	793,304
Douban-movie	1,883	57,018	1,563,754	101,408	6	319,918

4 Experiments

In this section, we firstly describe our interaction datasets and KG. Then, we introduce the baselines and experiment setup. Finally, we present the experiment results and discuss the influence of hyper-parameters.

4.1 Experimental Settings

Datasets Description. We utilize the three common datasets in our experiments. (1) **MIND**[1] was constructed from the Microsoft News. (2) **Last-FM**[2] was collected from the online music system Last.fm. (3) **Douban-Movie**[3] was collected from Douban Movies.

We preprocess these datasets and use the user and item ID embeddings as raw input. For the Last-FM dataset, we follow the data processing method released by KGIN [28] and obtain get classifications of artists using the tag data. We follow the knowledge graph construction method with Microsoft Satori[4] released by KGCN [25]. Then, we construct the category-item graph utilizing the item triples as (category, type.object.name, item). For the MIND dataset, we follow the data processing method published by KGCL [34] and use the catalogs as the category label. We follow the pre-processing data strategy in KGCL to construct the knowledge graph based on the spacy-entity-linker tool[5] and Wikidata[6]. We construct the category-relation-item triples using catalog information. For the Douban-movie dataset, we remove genres with fewer than 16 items to guarantee the data quality, and extract attributes to construct the KG utilizing genres, actors, directors, etc. The statistics of the data set are shown in Table 1.

Evaluation Metrics. To evaluate the performance, we adopt two universal metrics for Top-K recommendation: Recall@K, and Normalized Discounted Cumulative Gain (NDCG@K) [3,10].

Baselines Models. We compare our proposed approach with the following baseline methods: **FM** [18] is a bechmark factorization model, which models contextual information to provide context-aware rating predictions. **NFM** [9] combines

[1] https://msnews.github.io/.
[2] https://grouplens.org/datasets/hetrec-2011/.
[3] https://movie.douban.com/.
[4] https://searchengineland.com/library/bing/bing-satori.
[5] https://github.com/egerber/spaCy-entity-linker.
[6] https://query.wikidata.org/.

Table 2. Performance comparison of all methods

Dataset	Last-FM		MIND		Douban-movie	
Model	Recall	NDCG	Recall	NDCG	Recall	NDCG
FM	0.0591	0.0577	0.0651	0.0303	0.0291	0.0091
NFM	0.0743	0.0661	0.0749	0.0335	0.0307	0.0107
CKE	0.0732	0.0631	0.0820	0.0384	0.0316	0.0116
RippleNet	0.0785	0.0702	0.0768	0.0401	0.0439	0.0139
KGCN	0.0881	0.0744	0.0799	0.0406	0.0454	0.0154
KGAT	0.0943	0.0812	0.0849	0.0431	0.0481	0.0171
CKAN	0.0882	0.0796	0.0887	0.0468	0.0458	0.0158
KGIN	0.1298	0.0948	<u>0.1008</u>	<u>0.0493</u>	0.0517	0.0317
KGCL	<u>0.1473</u>	<u>0.1019</u>	0.0983	0.0481	<u>0.0558</u>	<u>0.0358</u>
KRec-C2	**0.1675**	**0.1130**	**0.1209**	**0.0626**	**0.0657**	**0.0478**

the linearity of FM and the non-linearity of neural network in modelling higher-order feature interactions. **CKE** [36] leverages the heterogeneous information in a knowledge base to improve the quality of recommender systems. **RippleNet** [22] stimulates the propagation of user preferences over the set of knowledge entities by automatically and iteratively extending a user's potential interests along links in the knowledge graph. **KGCN** [25] is an end-to-end framework utilizing high-order structure information and semantic information to capture inter-item relatedness effectively. **KGAT** [26] models the high-order connectivities in KG with an attention mechanism to discriminate the importance of the neighbors. **KGIN** [28] mainly performs relational path-aware aggregation of user intent-item and KG triples to enhanced learning. **CKAN** [30] introduces a heterogeneous propagation mechanism and combines collaborative filtering representation features with knowledge graph embeddings. **KGCL** [34] adopts a knowledge graph augmentation schema to suppress KG noise in information aggregation.

Parameter Settings. We set the size d of the ID initialization embedding to 16, and the batch size s of the method is 128 for all methods. The learning rate η is adjusted in $\{10^{-2}, 10^{-3}, 5^{-3}, 10^{-4}, 5^{-4}\}$, and the $L2$ regularization coefficient in the method is searched in $\{10^{-5}, 10^{-4}, \cdots, 10^{-1}\}$. In addition, we set the receptive neighborhood size of KGCN to 16 and adjust the number of layers L to be $\{1, 2, 3\}$. We discuss other parameters in Sect. 4.5.

4.2 Performance Comparison

We report the overall performance evaluation of all methods in Table 2, where we set K=20 following the settings of most methods [28,34], and the strongest baseline is underlined. From the table, we summarize the following observations: KRec-C2 consistently outperforms other baselines across three datasets. Specifically, our model achieves a significant experimental effect compared with the

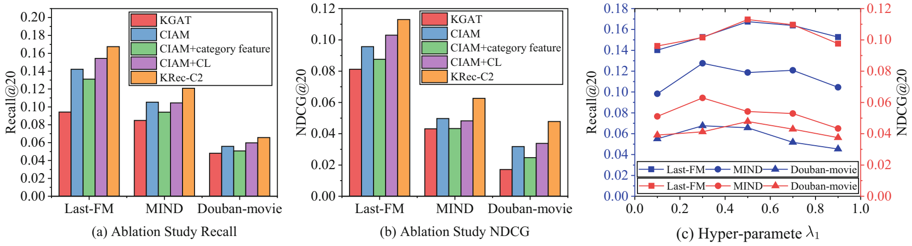

Fig. 3. (a) and (b) are results of ablation study. (c) is result of hyperparameter λ_1

strongest baselines in this paper, where the result of Recall@20 are 16.75%, 12.08%, and 6.57% in Last-FM, MIND, and Douban-movie, respectively. We obtain the best result at recall@20 on Last-FM data, which is about 2.02% higher than the KGCL. While, we obtain the minimum value at recall@20 on the Douban-movie data, which is still about 1.01% higher than the KGCL. The above results verify the effectiveness of our KRec-C2 model. Overall, the improvements obtained from KRec-C2 can be attributed to three modules: (1) CIAM. In contrast, all baselines ignore the context and only model user-item latent features such as the node-based aggregation method. (2) CIFM. Compared with baselines (such as KGAT, CKAN, and KGIN), it can learn the common category-level features from the KG. (3) CLM. Self-supervised signals relieve the data noise in KG and enhance the model's robustness. Further, we also observe that the difference between our model with other baselines is more stable through horizontal comparison.

The Table 2 shows that most KG-based recommender systems (such as KGAT, KGIN, and CKAN) achieve better performance than FM, NFM, and CKE. Detailed, KGIN obtains the best result at recall@20 on Last-FM data, which is about 5.66% higher than the CKE. Specifically, KGIN introduces the relational path embedding user intent, enhances user features by aggregation scheme. However, such methods ignore the contextual relationship and latent intent between items within a interaction sequence. On the other hand, the above table results show that the KGCL model is better than other models on Last-FM and Douban-movie. In the worst case, KGCL has a lower value of 0.0983 at Recall@20 on MIND, within spitting distance of the KGIN result. Excitingly, our model is 1.505% higher than KGCL on average from the above table. The excellent performance demonstrates that our model effectively combines context awareness and the category-level self-supervised signal to improve representation.

4.3 Ablation Study of KRec-C2 Model

In this section, we conduct ablation studies to verify the effectiveness of our three core components. The results are shown in Fig. 3 (a) and (b). First, compared with KGAT, the effect of CIAM has been a noticeable improvement. Specifi-

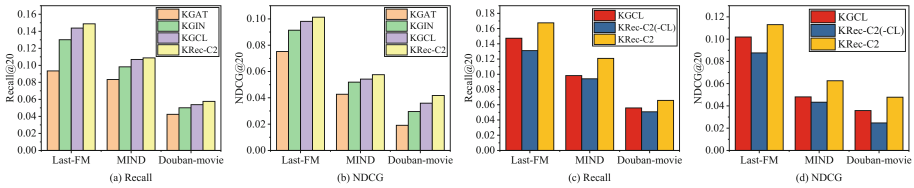

Fig. 4. Performance in data sparsity and noise.

cally, the maximum improvement value is 4.78% at Recall@20 on Last-FM, and the minimum improvement value is 0.66% at NDCG@20 on MIND. The above results prove that applying context awareness to user-item relational modeling is vital. Second, the average result of CIAM+category feature decreases by 0.81% compared with ICAM. Fusing the category feature enhances the model's generalization performance, leading to poor effects. Finally, after adding the contrastive learning of the category-level view, compared with ICAM+CL, the maximum value of improving is 1.64% at Recall@20 on MIND, and the minimum value of improving is 0.61% at Recall@20 on Douban-movie. The above results show the effectiveness of the CIFM and CLM.

4.4 The Effects of Alleviating Data Sparsity and Noisy

In this subsection, we evaluate the performance of KRec-C2 on noisy and sparse data. To be specific, noise data refers to the error and redundancy in KG. Sparse data means few interactions between users and items.

First, to verify the robustness of our KRec-C2 in handling a few interactions, we follow similar settings in KGCL to generate sparse user sets for MIND data, 20 interactions for Last-FM data, and 32 interactions for Douban-movie data. From the Fig. 4 (a) and (b), we observe that KRec-C2 is slightly higher than the KGCL. Specifically, the maximum improvement efficiency was improved up to 0.6% at NDCG@20 on Douban-movie, and the minimum was improved up to 0.18% at Recall@20 on MIND. Moreover, the performance of KRec-C2 is significantly superior to KGAT and KGIN. Then, we compare the results of KRec-C2 with KGCL and KRec-C2(-CL) in the Fig. 4 (c) and (d). Specifically, the effects of Recall@20 rose by 2.02 Technically, we designed a CLM component to verify our hypothesis that CLM is an essential component in alleviating the noise issue and highlighting the category-level signal. Figure 4 (c) and (d) show that the result of KRec-C2(-CL) is lower than KGCL and KRec-C2.

4.5 Hyper-Parameter Sensitivity

To evaluate the effect of the hyperparameter λ_1, aggregator, and interaction sequence size, we conduct experiments on three datasets by varying their values.

(1) The hyperparameter λ_1. As shown in Fig. 3 (c), the performance of KRec-C2 first increases and then decreases with the increase of λ_1. The three

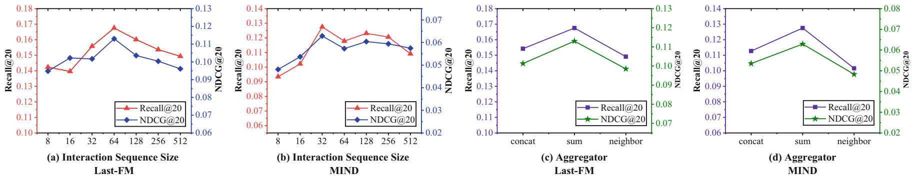

Fig. 5. Impact of interaction sequence size and aggregator on Last-FM and MIND, separately.

datasets of Last-FM, MIND, and Douban-movie achieve the best results when λ=0.5, 0.3, and 0.5, respectively. The results show that the model can enhance knowledge representation and improve robustness by introducing self-supervised learning features. **(2) Interaction sequence size.** As shown in Fig. 5 (a) and (b), Last-FM and MIND usually get the best performance when L=64 and 32 respectively. While aggregating more features may cause poor performance. The reason for the above result may be related to the training of the discriminator. More hidden layers need more parameters to be trained so that the discriminator hardly reaches a steady state. **(3) Aggregator.** We choose different aggregators from *sum*, *concat* and *neighbor* to study the impact of perception in KRec-C2. The results are shown in Fig. 5 (c) and (d), which indicates that KRec-C2 is more sensitive to the sum aggregator. The outperforms of the sum aggregator surpass the other two aggregators.

5 Related Work

Knowledge Graph-Based Recommendation. Existing KG-enhanced works for recommendation fall into three categories: embedding-based, path-based, and joint models. i) The embedding-based models generally obtain vector representations of products, users, and their relationships by the knowledge graph and apply these representations to similarity calculation [23,24,36]. ii) The Path-based models commonly enhance recommendation effects through user-product connection similarity, where path similarity is used to measure the similarity of connection entities in KG [12,29,35]. iii) The joint models usually capture the correlation between entities by mining attributes and discovering the high-level structural and potential information in KG [6,22,25,26,30]. Although many effective models have been proposed, the primary problem is that these methods (such as KGAT [26], KGIN [28], and CG-KGR [6]) only focus on aggregating potential features in the KG, and ignore the context information in the interaction sequence. This may lead to insufficient profiling for both users and items, and then the capability of recommendation system may be suppressed. To address these issues, we proposed ICAM in our model.

Contrastive Learning for Recommendation. Similar to works in NLP [8] and CV [5], contrastive learning aims to learn self-supervised representations by comparing positive and negative samples from different views in RS [1,7,37]. Under the scenario of KG-based RS, most contrastive learning methods generate two views by uniform data augmentation schemes [31,32,34]. Other works focus on exploring contrastive learning with a novel multi-level interactive view [38]. For instance, MCCLK [38] performs contrastive learning across three views on both local and global levels, mining comprehensive graph feature and structure information in a self-supervised manner. However, these works ignore the learning of the category-level signals, which are essential for acquiring salient features. To alleviate this problem, we proposed CML by the category-level view.

6 Conclusion

In this paper, we explore context-awareness, category-level intent fusion, self-supervised learning, and implement joint modeling for KG-based RS. We propose a novel framework KRec-C2, which enhances the performance with an end-to-end fashion under the paradigm of KG-based RS. Specifically, our method consists of three modules, namely ICAM, CIFM, and CLM. ICAM captures the context feature from an interaction sequence to enhance the user's representation. CIFM fuses the category-level signal in the training stage to obtain the category-level intents. CLM captures the relevance of the category-level view for alleviating noise issues. Our experiments on three real-world datasets demonstrate the rationality and effectiveness. This work explores the potential of context awareness and contrastive learning in KG-based recommendations. In the future, we will optimize the performance of recommendation system from the model structure.

Acknowledgements. This work was supported in part by the NSFC (Natural Science Foundation of China) grants (62172423 and 91846204).

References

1. Bian, S., Zhao, W.X., Zhou, K., et al.: Contrastive curriculum learning for sequential user behavior modeling via data augmentation. In: The 30th ACM International Conference on Information and Knowledge Management (CIKM) (2021)
2. Cao, Y., Wang, X., He, X., et al.: Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In: The World Wide Web Conference (WWW), pp. 151–161 (2019)
3. Chen, C., Zhang, M., Ma, W., et al.: Jointly non-sampling learning for knowledge graph enhanced recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 189–198 (2020)
4. Chen, J., Zhang, H., He, X., et al.: Attentive collaborative filtering: multimedia recommendation with item-and component-level attention. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2017)
5. Chen, T., Kornblith, S., Norouzi, M., et al.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning (ICML), pp. 1597–1607 (2020)

6. Chen, Y., Yang, Y., Wang, Y., et al.: Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 299–311. IEEE (2022)
7. Chen, Y., Liu, Z., Li, J., et al.: Intent contrastive learning for sequential recommendation. In: Proceedings of the ACM Web Conference (WWW), pp. 2172–2182 (2022)
8. Fu, H., Zhou, S., Yang, Q., et al.: LRC-BERT: latent-representation contrastive knowledge distillation for natural language understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2021)
9. He, X., Chua, T.S.: Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 355–364 (2017)
10. He, X., Deng, K., Wang, X., et al.: LightGCN: simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 639–648 (2020)
11. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., et al.: Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670 (2018)
12. Hu, B., Shi, C., Zhao, W.X., et al.: Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD) (2018)
13. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
14. Lee, D., Oh, B., Seo, S., et al.: News recommendation with topic-enriched knowledge graphs. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM), pp. 695–704 (2020)
15. Liang, D., Krishnan, R.G., Hoffman, M.D., et al.: Variational autoencoders for collaborative filtering. In: Proceedings of the 2018 World Wide Web Conference (WWW), pp. 689–698 (2018)
16. Lin, T.H., Gao, C., Li, Y.: CROSS: cross-platform recommendation for social E-commerce. In: The 42nd International ACM SIGIR Conference (SIGIR), pp. 515–524 (2019)
17. Pujara, J., Augustine, E., Getoor, L.: Sparsity and noise: where knowledge graph embeddings fall short. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1751–1756 (2017)
18. Rendle, S., Gantner, Z., Freudenthaler, C., et al.: Fast context-aware recommendations with factorization machines. In: Proceedings of the 34th international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 635–644 (2011)
19. Schafer, J.B., Konstan, J.A., Riedl, J.: E-commerce recommendation applications. Data Min. Knowl. Disc. **5**, 115–153 (2001)
20. Smirnova, E., Vasile, F.: Contextual sequence modeling for recommendation with recurrent neural networks. In: Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems (RecSys), pp. 2–9 (2017)
21. Wang, G., Zhang, W., Wang, R., et al.: Label-free distant supervision for relation extraction via knowledge graph embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2246–2255 (2018)

22. 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 (CIKM), pp. 417–426 (2018)
23. 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 (WWW), pp. 1835–1844 (2018)
24. Wang, H., Zhang, F., Zhao, M., et al.: Multi-task feature learning for knowledge graph enhanced recommendation. In: The World Wide Web Conference (WWW), pp. 2000–2010 (2019)
25. Wang, H., Zhao, M., Xie, X., et al.: Knowledge graph convolutional networks for recommender systems. In: The World Wide Web Conference (WWW), pp. 3307–3313 (2019)
26. 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 (SIGKDD), pp. 950–958 (2019)
27. Wang, X., He, X., Wang, M., et al.: Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 165–174 (2019)
28. Wang, X., Huang, T., Wang, D., et al.: Learning intents behind interactions with knowledge graph for recommendation. In: Proceedings of the Web Conference (WWW), pp. 878–887 (2021)
29. Wang, X., Wang, D., Xu, C., et al.: Explainable reasoning over knowledge graphs for recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 33, pp. 5329–5336 (2019)
30. Wang, Z., Lin, G., Tan, H., et al.: CKAN: collaborative knowledge-aware attentive network for recommender systems. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 219–228 (2020)
31. Wei, W., Huang, C., Xia, L., et al.: Contrastive meta learning with behavior multiplicity for recommendation. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM), pp. 1120–1128 (2022)
32. Wu, J., Wang, X., Feng, F., et al.: Self-supervised graph learning for recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 726–735 (2021)
33. Xu, Q., Shen, F., Liu, L., et al.: GraphCAR: content-aware multimedia recommendation with graph autoencoder. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR), pp. 981–984 (2018)
34. Yang, Y., Huang, C., Xia, L., et al.: Knowledge graph contrastive learning for recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 1434–1443 (2022)
35. 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 (WSDM), pp. 283–292 (2014)
36. 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 (SIGKDD), pp. 353–362 (2016)

37. Zhou, K., Wang, H., Zhao, W.X., et al.: S3-Rec: self-supervised learning for sequential recommendation with mutual information maximization. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM), pp. 1893–1902 (2020)
38. Zou, D., Wei, W., Mao, X.L., et al.: Multi-level cross-view contrastive learning for knowledge-aware recommender system. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 1358–1368 (2022)