Information Systems 115 (2023) 102207

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Collaborative knowledge-aware recommendation based on neighborhood negative sampling

Zewei Lin, Liping Qu *

College of Computer Science and Technology, Harbin Engineering University, China

ARTICLE INFO

Article history:
Received 25 November 2022
Received in revised form 5 March 2023
Accepted 12 March 2023
Available online 20 March 2023
Recommended by Dennis Shasha

Keywords:
Recommendation system
Gated aggregation
Knowledge-aware
Collaboration-aware
Knowledge graph
GNN
Negative sampling

ABSTRACT

Knowledge graph and negative sampling, as the sources of auxiliary information, have been playing a vital role since they were included in the recommendation system. Creating an end-to-end model based on a graph neural network is the current technical trend. In order to generate samples with negative signals for the recommended model, negative sampling is also done from the unobserved data. However, the current end-to-end model based on graph neural network is unable to effectively capture the high-order collaboration signals of items, making it unable to learn the high-quality user and item representation. Additionally, existing negative sampling technique is insufficient for producing high-quality negative samples that can both reflect users' true preferences and offer information for model training. Therefore, this paper proposes a new model, namely, collaborative knowledge-aware network based on neighborhood negative sampling (CKNNS). Users and items are technically modeled based on the collaboration-aware and knowledge-aware. To help the model better capture the users' preferences and the personality traits, a gated aggregation strategy is used to adaptively capture collaboration-aware signals and knowledge-aware information. Meanwhile, a negative sampling method based on user's neighborhood has been proposed. Specifically, the similarity of users is judged according to their collaboration information, so as to build a negative sampling domain for a specific user and improve the quality of negative sampling. The experimental outcomes on four benchmark datasets reveal that CKNNS outperforms sophisticated approaches like LightGCN, CKAN, KGIN and KGCL.

1. Introduction

Human civilization has entered an era of information explosion as a consequence of the rapid growth of Internet technology, and data redundancy makes it difficult for individuals to retrieve their own information in the great volume of information. To address this issue, researchers proposed adopting a recommendation system that can detect users' preferences and provide tailored content by evaluating users' historical activity data in order to suit users' specific demands.

Content-based recommendation, collaborative filtering recommendation, and mixed recommendation are the most common traditional recommendation systems. However, traditional recommendation systems [1,2] regularly encounter the problems of data sparseness and cold start, which means that when applying recommendation algorithms, it is impossible to extract potential preference vectors by analyzing the characteristics of users and items, thereby affecting the recommendation effect. Recently, the knowledge graph (KG) was introduced to alleviate the cold start problem and improve the interpretability of recommendations. KG is a heterogeneous graph of the relationships and entities. It contains a variety of linkages, which aids in properly capturing users' interests and increases recommendation impact and interpretability.

At present, the recommendation system based on knowledge graph can be divided into three methods: path-based method, embedding-based method, and the hybrid method. Some early recommendation algorithms [3,4] employ the knowledge graph embedding technique directly to improve item representation. Subsequent research [5,6] give more interactive information by constructing meta paths from users to items. However, since the defining of meta-path is a labor-intensive procedure that requires a significant amount of professional time, the generalization of this technique is limited.

With the success of graph neural network applications in many industries, the end-to-end recommendation model based on graph neural network in the recommendation field is also encouraged [7–13]. The central principle is to disseminate high-order in formation on KG repeatedly in order to efficiently integrate the knowledge information of multi-hop neighbors and better represent items, hence enhancing recommendation performance. Existing end-to-end recommendation models based on graph neural networks have produced good recommendation

* Corresponding author.
E-mail address: quliping@hrbeu.edu.cn (L. Qu).

https://doi.org/10.1016/j.is.2023.102207
0306-4379/

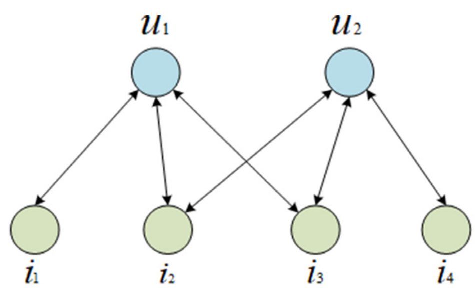

Fig. 1. Example of user preference similarity.

performance; however, they all aggregate item information for KG, ignoring collaboration information between users and items and only iteratively collecting knowledge associations from items and their neighbors in KG. This overlooked collaboration information from previous interactions between users and items is critical in user preference modeling. Using Fig. 1 as an example, user u_1 likes items i_2 and i_3, while user u_2 also likes items i_2 and i_3, implying that the preferences of u_1 and u_2 are similar. As a result, u_1's favorite i_1 has a high probability that it is also liked by u_2, and it can be used as a potential recommendation item to u_2, while user u_2's favorite i_4 can be recommended to u_1 in the same way.

User–item history interactions, on the other hand, can only offer positive feedback, and deriving negative signals from this data is similarly difficult. The existing static negative sampling algorithms [14,15], such as random negative sampling and popular negative sampling, are prone to producing low-quality negative samples because of their poor ability to mine data information; the adaptive negative sampling algorithm [16–18] selects existing samples as hard negative samples through model scoring, which will aggravate the problem of false negative samples in sampling which will reduce the recommendation performance of the model; while the graph based negative sampling technique [19–21] may substantially increase the model's recommendation impact, it is frequently accompanied by a considerable amount of computational overhead, which hampers model training. In summary, our contributions are as follows:

- Through the graph neural network and knowledge graph, it is possible to better express and spread the collaboration signals and knowledge associations at the same time, and then use gated aggregation scheme to adaptively collect collaboration signals and knowledge-aware information.
- Using user–item historical interactions data to capture the collaboration information of specific users, so that the similarity of preferences among users can be calculated. A unique neighborhood can be created for each user according to the similarity of user preferences. Neighbors in the domain are similar to the preferences of current users, so the items that these neighbors like will be liked by current users to a great extent. Therefore, when taking negative samples for current users, it is necessary to exclude the items that neighbors like, so as to realize the construction of negative sampling domain and generate high quality negative samples.
- Experiments on four published benchmark datasets are performed in this research to demonstrate the advantage of CKNNS.

2. Related work

2.1. Recommendation system based on knowledge graph

As an effective auxiliary information source in recommendation systems, knowledge graph contain a large amount of information about recommended items and rich semantic associations among them, thus effectively making up for the sparsity of interactions between users and items. CKE [22] embeds knowledge graph into a structured information vector by TransR [23], and combines the text information and visual information of the knowledge graph to generate the potential representation of the item; Rec [6] uses a convolutional neural network to sample each meta-path, and obtains the path between users and articles, thus constructing the user preference characteristics based on the meta-path; KGIN [9] captures the intention of user–item interactions by constructing an auxiliary knowledge graph, and models the intention by attention combination of KG relation; Ripplenet [13] introduces the knowledge graph with the item of user interactions as a bridge. On the basis of the initial item, it spreads the user's interest on the knowledge graph, so as to capture the user's preference and construct the potential representation of the user; KGCL [12] is based on the contrast learning paradigm of knowledge graph enhancement, in order to suppress the KG noise in the process of information aggregation, so as to learn more robust knowledge-aware representation of items and alleviate the long tail and noise problems of KG.

2.2. Negative sampling in recommended system

Negative sampling can provide high-quality negative samples for the recommendation model. After that, the model can increase the discrimination between positive and negative examples with the help of a loss function to learn the information in the data and improve recommendation accuracy. Negative sampling algorithms are divided into three classes based on their sample methods: static negative sampling algorithms, adaptive negative sampling algorithms, and graph-based negative sampling algorithms. PNS [15] is an algorithm for static negative sampling. Its concept is to weight sample the commodities in the sampling pool by using commodity popularity as the sampling weight. The more popular something is, the simpler it is to taste it. Typically, popularity is determined by the number of times a commodity is consumed; DNS [18] is a kind of adaptive negative sampling algorithm. The plan is to score each sample using the current model as a sampling model, train the current model using the higher-scoring samples as negative samples, and then repeat with new models; KGPolicy [21] is a negative sampling technique based on graphs. The negative sampling model navigates from the positive interactions of user–items via exploratory operation, adaptively receives negative signals of knowledge-aware, and eventually develops a negative item to feed to the recommendation model.

2.3. Gated aggregation in recommended system

Gated aggregation mechanism can enable the model to adaptively control the transmitted data and achieve the effect of selective memory, so as to obtain more valuable information and improving the efficiency of the model. KGIE [24] models the user's interest expression from the item-level and the itemset-level. In order to combine the two levels of interest expression and consider the unavailability of the user's real interaction intention, KGIE uses a learnable gated fusion unit to balance the contributions of the two levels of interest expression; GateNet [25] based on the existing deep learning model, the gated aggregation mechanism is introduced in the embedding layer and the deep neural

Table 1
All important notations.

Notations	Descriptions
$E = \{e_1, e_2, ..., e_k\}$	Entity set
$G = \{(u, i) \mid u \in U, i \in I\}$	Bipartite graph of user–item historical interaction
$U = \{u_1, u_2, ..., u_X\}$	User set, $U \subset E$
$I = \{i_1, i_2, ..., i_Y\}$	Item set, $I \subset E$
T	Triple set
$R = \{r_1, r_2, ..., r_g\}$	Relation set between entities
$G_k = \{(h, r, t) \mid h, t \in E, r \in R\}$	Heterogeneous graph
f_{agg}	Mean aggregator
$N = \{(i, r, t) \mid (i, r, t) \in KG\}$	Item in KG and its first-order neighborhood
$Nu = (Nu_1, Nu_2, ..., Nu_M)$	Neighborhood of user
$Ni = \{i \mid (u, i) \in G\}$	Item and its interactions
M	Negative samples
m	Filter margin of cosine contrast loss
$\mathbf{k}_i^{(0)}$	Initial knowledge-aware vector of item I
$\mathbf{c}_i^{(0)}$	Initial collaboration-aware vector of item I
$\mathbf{e}_u^{(0)}$	Initial embedding of user U
$\mathbf{e}_i^{(0)}$	Item vector generated by gating module
$\mathbf{k}_i$	Final knowledge-aware vector of item I
$\mathbf{c}_i$	Final collaboration-aware vector of item I
$\mathbf{e}_u$	Final user embedding
V	Threshold of user's neighborhood

network hidden layer, so that the model can independently select important feature information to participate in feature cross-combination. KAeDCN [26] have already initialized the vector for each item to learn the sequential dependency and prepares the pre-trained vector from the knowledge space to obtain the fine-grained preference for the item attribute dimension. Then, the gated aggregation mechanism is used to process the above information from two channels and containing two characteristics, and the knowledge is selectively merged for each project to obtain a more comprehensive representation.

3. Methodology

Table 1 summarizes all important notations used in this paper.

3.1. Problem formulation

3.1.1. user–item bipartite graph

A particular user–item historical interactions is described as a bipartite graph $G = \{(u, i) \mid u \in U, i \in I\}$, where user set $U = \{u_1, u_2, \ldots, u_X\}$ and itemset $I = \{i_1, i_2, \ldots, i_Y\}$, X and Y indicate the number of users and items, respectively.

3.1.2. Knowledge graph

Let T represent the triple set, $E = \{e_1, e_2, \ldots, e_k\}$ represent the entity set, and $R = \{r_1, r_2, \ldots, r_g\}$ represent the relation set. Let KG be a heterogeneous graph $G_k = \{(h, r, t) \mid h, t \in E, r \in R\}$, with each triple $(h, r, t) \in T$ indicating a connection between the head and tail entities.

3.1.3. Task description

Given the bipartite graph and knowledge graph of user–items, the goal of this paper is to learn the feature representation of users and candidate items, and to predict the click probability of user U on items that U have had no interaction with before.

3.2. The framework of model

As shown in Fig. 2, the CKNNS model proposed in this paper includes four key modules:

- Knowledge-aware embedding module: Create the item's knowledge-aware embedding vector.
- Collaboration-aware embedding module: Create the item's collaboration aware embedding vector.
- Neighborhood negative sampling module (NNS): Create negative sampling domain for user, then provide high-quality negative samples to the model.
- Gated aggregation module: Adaptively adjust the contribution of knowledge-aware embedding vector and collaboration-aware embedding vector to recommendation performance of the CKNNS.

For example, for the input user–item historical interactions and knowledge graph, the collaboration-aware embedding module capture the high-order collaborative information to generate the item's collaboration-aware embedding, while the knowledge-aware embedding module captures the knowledge association to generate the item's knowledge-aware embedding. Simultaneously, the neighborhood negative sampling module will capture the similarity of users and construct a specific neighborhood according to the input user–item historical interactions. Finally, the prediction layer is given collaboration-aware embedding, knowledge-aware embedding, user feature representation, and negative samples, and then evaluates the likelihood of interactions between users and items using cosine similarity.

3.2.1. Knowledge-aware embedding module

An item i can exist in numerous triples in the knowledge graph, and the entities linked with the item i, as its related qualities, can represent the prospective features of the item to some extent. Therefore, the neighborhood of item i can help to model the item. We utilize $N = \{(i, r, t) \mid (i, r, t) \in KG\}$ to represent the item i in KG and its associated first-order neighborhood, and then aggregate the knowledge association of entities in the first-order neighborhood to construct the item's first-order knowledge-aware representation $\mathbf{k}_i^{(1)}$:

$$\mathbf{k}_i^{(1)} = f_{agg}(\{(\mathbf{k}_i^{(0)}, \mathbf{e}_r, \mathbf{e}_t^{(0)}) | (\mathbf{k}_i^{(0)}, \mathbf{e}_r, \mathbf{e}_t^{(0)}) \in KG\}) \tag{1}$$

where $\mathbf{k}_i^{(0)}$ is the entity to which the item is mapped on KG, $\mathbf{k}_i^{(1)}$ is the first-order knowledge-aware representation of the item after aggregating the first-order neighborhood of $\mathbf{k}_i^{(0)}$, and f_{agg} is an aggregation function that extracts and combines items knowledge associations from the neighborhood of $\mathbf{k}_i^{(0)}$. The feature representation of $\mathbf{k}_i^{(0)}$, $\mathbf{e}_r$ and $\mathbf{e}_t^{(0)}$ (initialized by Xavier) are integrated by formulas (2) and (3), which may more effectively disclose the knowledge relationship across entities in KG. And then, we define f_{agg} as an average aggregator to aggregate the neighborhood of item i to produce $\mathbf{k}_i^{(1)}$:

$$\mathbf{e}_r^{\mathrm{new}(0)} = \mathbf{e}_r + \mathbf{k}_i^{(0)}, (i, r, t) \in N \tag{2}$$

$$\mathbf{e}_t^{\mathrm{new}(0)} = \mathbf{e}_t^{(0)} + \mathbf{k}_i^{(0)}, (i, r, t) \in N \tag{3}$$

$$\mathbf{k}_i^{(1)} = \frac{1}{|N|} \sum_N (\mathbf{e}_r^{new(0)} + \mathbf{e}_t^{new(0)}) \tag{4}$$

where $\mathbf{e}_t^{(0)}$ are the entities in the first-order neighborhood of item i, $\mathbf{e}_r$ are the relationships of entities, $\mathbf{e}_r^{new(0)}$ and $\mathbf{e}_t^{new(0)}$ are the new feature representations produced by combining $\mathbf{e}_r$, $\mathbf{e}_t^{(0)}$ and $\mathbf{k}_i^{(0)}$.

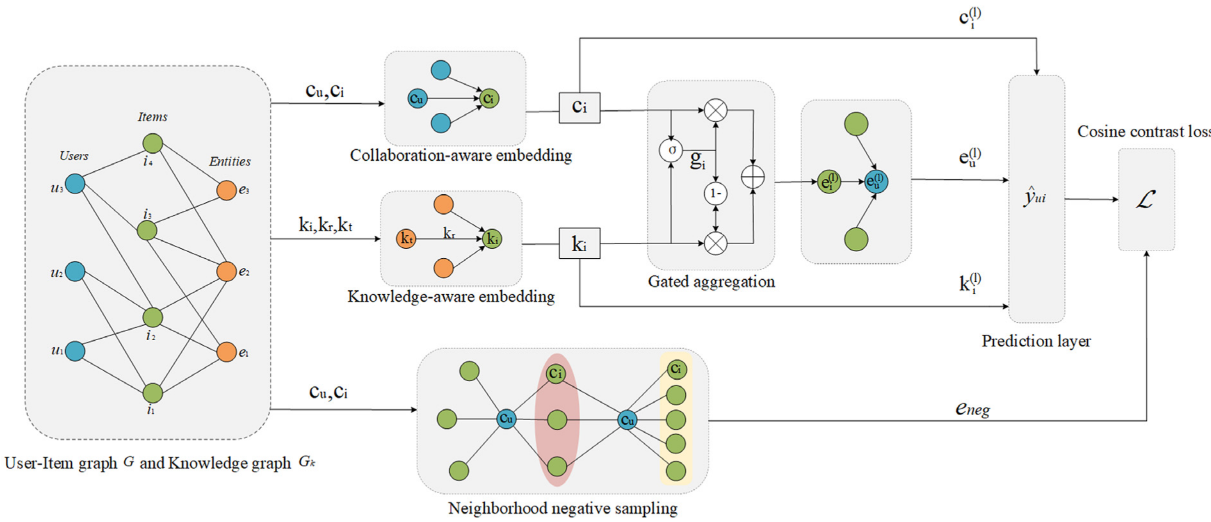

Fig. 2. The model framework of CKNNS.

The knowledge-aware representation $\mathbf{k}_i^{(1)}$ of each entity listed in item can also be created in a similar manner. More aggregation layers are then piled to investigate the item's high-level knowledge relationship. After stacking l aggregation layers, the knowledge-aware representation of the item can be defined as follows:

$$\mathbf{k}_i^{(l)} = \frac{1}{|N|}\sum_{N}(\mathbf{e}_r^{new(l-1)} + \mathbf{e}_t^{new(l-1)}) \tag{5}$$

3.2.2. Collaboration-aware embedding module

In order to better characterize user preferences, this module is used to translate the prospective collaboration information from the user–item bipartite graph into display codes. In more detail, the original item $\mathbf{c}_i^{(0)}$ and the user–item bipartite graph G are both provided, and $N_u = \{u \mid (u, i) \in G\}$ is defined as all users who have interacted with the initial item $\mathbf{c}_i^{(0)}$ (initialized by Xavier). Create a high-order collaboration representation $\mathbf{c}_i^{(l)}$ of the item by stacking l aggregation layers to incorporate the information about cooperation between item and its neighbors:

$$\mathbf{c}_i^{(1)} = \frac{1}{|N_u|}\sum_{u\in N_u}\mathbf{e}_u^{(0)} \tag{6}$$

$$\mathbf{c}_i^{(2)} = \frac{1}{|N_u|}\sum_{u\in N_u}\mathbf{e}_u^{(1)} \tag{7}$$

...

$$\mathbf{c}_i^{(l)} = \frac{1}{|N_u|}\sum_{u\in N_u}\mathbf{e}_u^{(l-1)} \tag{8}$$

where $\mathbf{e}_u^{(0)}$ is the neighbor user of initial item $\mathbf{c}_i^{(0)}$, $\mathbf{c}_i^{(l)}$ is the collaboration-aware embedding at the l layers of the item i, and $\mathbf{e}_u^{(l-1)}$ is the representation vector of user after $l-1$ layers, which we will detail in Section 3.2.4.

3.2.3. Neighborhood negative sampling module

Given the bipartite graph G of user–item interactions and the specified user u in the graph, the purpose of this module aims to generate the neighborhood of user u and generate high-quality negative samples for it. As shown in Fig. 1, it can be inferred from paths $u_1\rightarrow i_2\rightarrow u_2$ and $u_1\rightarrow i_3\rightarrow u_2$ that user u_1 and user u_2 have a certain degree of similarity in preferences. So the neighborhood of u_1 is refined according to the fact that the interacted items of all users overlap with each other. The Algorithm 1 describes the neighborhood generation strategy.

Therefore, when generating high-quality negative samples of a given user, the items interacted by the neighbors of the user can be excluded, and negative samples can be generated from the remaining items, thus effectively reducing the true positive rate of negative samples. Only random sampling is employed in this paper to create negative samples from the remaining items.

3.2.4. Gated aggregation module

This work creates a gated aggregation module for the knowledge-aware embedding $\mathbf{k}_i^{(l)}$ and collaboration-aware embedding $\mathbf{c}_i^{(l)}$ derived from the $l-$ layer, in order to better gather information from the item's neighborhood, encouraging the entire recommendation model to achieve better results:

$$\mathbf{e}_i^{(l)} = \sigma(W_1\mathbf{k}_i^{(l)} + W_2\mathbf{c}_i^{(l)})\mathbf{k}_i^{(l)} + ((1-\sigma(W_1\mathbf{k}_i^{(l)} + W_2\mathbf{c}_i^{(l)}))\mathbf{c}_i^{(l)}) \tag{9}$$

where W_1 and $W_2 \in R^{d\times d}$ are transform parameters that can be learned, and σ is the Sigmoid function. As a result, the model may alter the contribution of knowledge-aware and collaboration-aware embedding to user modeling based on the estimated gating signal. That is, when item knowledge-aware embedding is insufficient to reflect user preferences, the model will rely more on item collaboration-aware embedding to do so.

After constructing the l–level item feature representation, the user feature representation of l layer can be constructed. Specifically, let $N_i = \{i \mid (u, i) \in G\}$ is a set of items with which users have interacted. The prospective attributes of users may be characterized using the feature representations of these items as follows:

$$\mathbf{e}_u^{(l)} = \frac{1}{|N_u|}\sum_{i\in N_i}\mathbf{e}_i^{(l-1)} \tag{10}$$

3.2.5. Prediction layer

The knowledge-aware embedding and collaboration-aware embedding of item i in each layer, as well as the possible feature representation of user u, may be produced by stacking L aggregation layers. The embedded information is then merged into each layer to generate its own final representation:

$$\mathbf{k}_i = \sum_{l=0}^{L}\boldsymbol{k}_i^{(l)} \tag{11}$$

$$\mathbf{c}_i = \sum_{l=0}^{L}\mathbf{c}_i^{(l)} \tag{12}$$

Algorithm 1: The algorithm for generating the user's neighborhood.

input: History interactions of User-item

output: The neighborhood of user $N'_u = \left(N_{u_1}, N_{u_2}, \ldots, N_{u_M}\right)$

Initialization: $N'_u \leftarrow \emptyset, N_{u_1} \leftarrow \emptyset, \ldots, N_{u_M} \leftarrow \emptyset$;

for $i \leftarrow 1$ **to** X **do**

  v_i is the number of items that u_i interacts with;

  for $j \leftarrow i+1$ **to** X **do**

    v_{ij} is the number of items that both u_i and u_j have interacted with;

    $\mathrm{H} = \frac{v_{ij}}{vi} \geq \mathrm{V}$ (described in section 4.4), $N_{u_i} \leftarrow N_{u_i} \cup \{u_j\}$;

    end

  end

$$\mathbf{e}_u = \sum_{l=0}^{L} \boldsymbol{e}_u^{(l)} \tag{13}$$

Cosine similarity is used to calculate the likelihood of interactions between user u and item i, after acquiring the final representation of user and item based on two types of item embedding methods:

$$y_{u,i}^c = \cos(\mathbf{e}_u, \mathbf{c}_i) \tag{14}$$

$$y_{u,i}^k = \cos(\mathbf{e}_u, \mathbf{k}_i) \tag{15}$$

And then adopt the sum of the two predictions as the final prediction score $\hat{y}_{ui}$:

$$\hat{y}_{ui} = y_{u,i}^c + y_{u,i}^k \tag{16}$$

Because the loss function proposed by SimpleX [27], cosine contrast loss $\mathcal{L}$, can use more negative samples to speed up the training convergence and enhance the model effect. At the same time, our neighborhood negative sampling module (NNS) can provide a large number of high-quality negative samples. Therefore, combining above two factors can get outstanding results, which can make better use of the high-quality negative samples provided by NNS, show the efficiency of NNS and improve the overall performance of the CKNNS. So this paper optimizes CKNNS using cosine contrast loss $\mathcal{L}$. According to the suggested neighborhood negative sampling, M negative samples are created for user in each interaction, to establish negative interaction based on the user–item historical interaction. To learn the model parameters, the following objective functions are minimized:

$$\mathcal{L} = \frac{1}{|M|}\sum_{j\in M}(0, y_{u,j}^c - m)_{max} + \frac{1}{|M|}\sum_{j\in M}(0, y_{u,j}^k - m)_{max} + (2 - \hat{y}_{ui}) \tag{17}$$

where $y_{u,j}^c$ is the cosine similarity between the user vector and the collaboration-aware embedding vector of the item, the $y_{u,j}^k$ is the cosine similarity between the user vector and the knowledge-aware embedding vector of the item. The margin m is used to filter out negative samples, and it usually takes a value in (0,1). Cosine contrast loss $\mathcal{L}$ primarily maximizes positive pair similarity while reducing negative pair similarity with m.

4. Experiments

4.1. Dataset description

The following four published benchmark datasets are utilized in this study to evaluate the CKNNS and the comparative model. We randomly split four benchmark datasets into training and testing sets at 8:2 ratio. The specific statistical information of the four benchmark datasets is shown in Table 2. If need code, you can contact us by email and we will give it to you after team evaluation.

- Last.FM: Supplied by [8] on github, it comprises 92,834 hits by 1892 people from online music services for 14,632 vocalists and its knowledge graph. The data is 99.72% sparse.
- Alibaba-iFashion: Released on github by [9], it comprises 1,781,093 purchases of 30,040 goods by 114,737 people from Alibaba's online shopping system, as well as the accompanying knowledge graph, and the data sparsity is 99.95%.
- Yelp2018: It comprises 1,185,068 interactions of 45,538 items by 45,919 users and the accompanying knowledge graph, and the data sparsity is 99.94%, as published by [7] on github.
- Moive100k: Provided by [11] on github, it contains 97,953 ratings of 1682 movies by 943 users from movie websites and corresponding knowledge graph, and the data sparsity is 93.82%.

4.2. Baselines

Three approaches based on supervised learning (MF, NCF), embedding (CKE), and graph neural networks (RippleNet, KGAT, CKAN, LightGCN, KGIN, KGCL) are chosen for comparison with the proposed CKNNS in this research. The following are the comparative methodologies used in this paper:

- **MF** [28] is a benchmark decomposition model that simply takes into account user–item interactions and does not require KG. To anticipate, this article use ID embedding of users and items.
- **NCF** [29] replaces the dot product in MF model with a multilayer perceptron to learn the match function of users and items.
- **CKE** [23] is a typical embedding-based technique. It employs TransR to transform entities in KG into low-dimensional vector representations and use them as item IDs in the MF framework. The KG relation, for example, is only utilized as a constraint in TransR and is used to describe regularized endpoints.
- **RippleNet** [13] combines regularization and path-based methods, which enrich user representations by adding that of items within paths rooted at each user.

Table 2
Specific statistics for the four publicly available baseline datasets.

		Last-FM	Movie-100k	Alibaba-iFashion	Yelp2018
user–item Interactions	♯ Users	1892	943	114 737	45 919
	♯ Items	14 632	1682	30 040	45 538
	♯ Interactions	92 834	97 953	1 781 093	1 185 068
	♯ Sparsity	99.72%	93.82%	99.95%	99.94%
Knowledge graph	♯ Entities	9366	4129	59 156	90 961
	♯ Relations	60	29	51	42
	♯ Triplets	15 518	41 751	279 155	1 853 704

- **KGAT** [7] is a graph neural network-based recommendation system. To build user and item representations, it combines KG with a user–item graph and employs an attention neighborhood aggregation approach on the entire graph.
- **LightGCN** [10] is a proposed technique based on a graph convolution network that streamlines convolution operations between users and items.
- **CKAN** [8] is a propagation-based recommendation approach. It learns ripple set embedding via a heterogeneous propagation technique and an attention network, resulting in user and item embedding.
- **KGIN** [9] is a graph neural network-based recommendation system. It presents a relationship-aware information aggregation approach to capture remote relationships in KG and simulates user interactions behavior based on probable purpose.
- **KGCL** [12] proposes a cross-view contrastive learning paradigm by constructing self-supervised signals from the structure of both the CF graph and the knowledge graph.

4.3. Parameter settings

The embedding dimension of all proposed models is set to 64, the loss function is optimized by Adam optimizer, and Xavier initializes the model parameters (including $\mathbf{k}_i^{(0)}$, $\mathbf{c}_i^{(0)}$, $\mathbf{e}_u^{(0)}$, $\mathbf{e}_t^{(0)}$, $\boldsymbol{e}_r$, etc.). Meanwhile, the model batchsize is 4096. We apply grid search for hyperparameters, changing the learning rate in $\{10^{-2}, 10^{-3}, 10^{-4}\}$ and the GNN layer of the propagation-based technique in {1, 2, 3}. In the neighborhood negative sampling proposed in this paper, the number of negative samples M of each user in the training set are set to {150, 150, 200, 400}, the filtering margin m of cosine contrast loss L are set to {0.8, 0.8, 0.8, 0.8}, and the threshold V of defining the neighborhood are set to {0.4, 0.8, 0.4, 0.8}, which correspond to the four datasets Last-FM, Movie-100k, Alibaba-iFashion and Yelp2018 respectively. Furthermore, for all methods, an early stop strategy is included, which means that if the reccall@20 indication on the testset does not increase for 20 consecutive epochs, it will stop early.

4.4. Experimental results and analysis

4.4.1. Performance comparison

We begin with the performance comparison Recall@20 and NDCG@20. The experimental findings of the CKNNS and the benchmark model are shown in Table 3.

- CKNNS had the best recommendation effect on four benchmark datasets based on the evaluation of Recall@20, NDCG@20 and AUC. Under the assessment metric of Recall@20, CKNNS rose by 2.34%, 2.90%, 4.20% and 4.36% on the datasets Last-FM, Movie-100k, Alibaba and Yelp2018, respectively, when compared to the best benchmark algorithm. In addition, compared with other models, the AUC metric of CKNNS on four datasets Last-FM, Movie-100k, Alibaba and Yelp2018 also increased by 2.15%, 0.88%, 1.20% and 0.59% separately. The results show that the gating mechanism can adaptively aggregate collaboration-aware embedding and knowledge-aware embedding, as well as neighborhood negative sampling, to obtain high-quality negative samples that can better capture user preferences and potential representation of items, and improve personalized recommendation accuracy.
- CKE outperformed MF on the Last-FM, movie-100k and Yelp2018 datasets by merely applying the knowledge graph embedding approach. NDCG@20 grew by 3.25%, 0.82% and 3.42% year on year, respectively. At the same time, the recommendation performance of the two methods was nearly same on the Alibaba dataset. As can be observed, the addition of KG can provide a significant amount of auxiliary information to the model, ease the problem of data sparsity to some extent, and so enhance model performance.
- In the experimental results of four datasets, KGAT, LightGCN, KGIN, KGCL and so on, which use graph neural network, have greater advantages than CKE in the comparison of Recall@20, NDCG@20 and AUC. It can be observed that the recommendation method based on graph neural network outperforms the embedding approach. This is due to the fact that the recommendation algorithm based on graph neural network can capture high-order information about items from a multi-hop neighborhood, resulting in high-quality item representation and superior user preferences.

4.4.2. Ablation research

- Replace the gated aggregation module with vector addition and label it as W/O A.
- Remove collaboration-aware embedding and replace it with knowledge-aware embedding. There is no need to use the gated aggregation module, which is indicated as W/O C, because there is only one type of item embedding.
- Replace the neighborhood negative sampling module with random negative sampling and label it as W/O N.

The following conclusions may be derived based on the performance evaluation findings of three CKNNS's variants shown in Table 4:

- Removing the gated aggregation module from three datasets reduces the model's recommendation performance. This is due to the gated aggregation module can give the model gate signal, therefore adaptively balancing the model's input from collaboration-aware embedding and knowledge-aware embedding. Only sampling the common vector addition aggregation will surely introduce some unneeded noise, affecting the model's recommendation performance.
- Removing collaboration-aware embedding from three datasets will reduce the model's recommendation performance. Because only sampling knowledge-aware embedding to model the item cannot obtain high-order collaboration information, and it is difficult to accurately model user's preferences, which will affect the model's recommendation performance.

Table 3
Performance comparison between the benchmark model and the CKNNS model.

	Last-FM			Movie-100k			Alibaba-iFashion			Yelp2018		
	Recall@20	NDCG@20	AUC	Recall@20	NDCG@20	AUC	Recall@20	NDCG@20	AUC	Recall@20	NDCG@20	AUC
MF	0.1279	0.0769	0.7363	0.3140	0.2796	0.8110	0.0707	0.0428	0.8140	0.0539	0.0351	0.8312
NCF	0.1304	0.0788	0.7370	0.3147	0.2823	0.8230	0.0694	0.0419	0.8220	0.0535	0.0377	0.8310
CKE	0.1464	0.0794	0.7371	0.3141	0.2819	0.8040	0.0703	0.0426	0.8070	0.0542	0.0363	0.8230
RippleNet	0.1597	0.0961	0.7760	0.3227	0.2943	0.8702	0.0947	0.0498	0.8530	0.0627	0.0421	0.8770
KGAT	0.1922	0.0977	0.8293	0.3364	0.3032	0.9170	0.1007	0.0614	0.8980	0.0683	0.0463	0.9140
LightGCN	0.3716	0.2338	0.8860	<u>0.3516</u>	<u>0.3352</u>	<u>0.9369</u>	0.1036	0.0641	0.9173	0.0619	<u>0.0486</u>	0.9580
CKAN	0.2264	0.1761	0.8622	0.3273	0.2907	0.9130	0.0970	0.0509	0.8870	0.0646	0.0441	0.9070
KGIN	0.3546	0.2167	0.8805	0.3354	0.3048	0.9354	0.1048	0.0647	0.9179	0.0705	0.0463	<u>0.9607</u>
KGCL	<u>0.3851</u>	<u>0.2443</u>	<u>0.8929</u>	0.3409	0.3119	0.9368	<u>0.1237</u>	<u>0.0724</u>	<u>0.9184</u>	<u>0.0757</u>	0.0454	0.9576
CKNNS	**0.3942**	**0.2510**	**0.9121**	**0.3618**	**0.3417**	**0.9451**	**0.1289**	**0.0822**	**0.9294**	**0.0790**	**0.0507**	**0.9664**

Table 4
Impact of gated aggregation, collaboration-aware embedding and NNS.

	Last-FM		Movie-100k		Alibaba-iFashion		Yelp2018	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
w/o A	0.3802	0.2352	0.3504	0.3314	0.1261	0.0801	0.0763	0.0496
w/o C	0.3782	0.2334	0.3471	0.3287	0.1203	0.0766	0.0745	0.0487
w/o N	0.3729	0.2315	0.3442	0.3237	0.1228	0.0796	0.0762	0.0486
CKNNS	**0.3942**	**0.2510**	**0.3618**	**0.3417**	**0.1289**	**0.0822**	**0.0790**	**0.0507**

Table 5
Impact of model depth.

	Last-FM		Movie-100k		Alibaba-iFashion		Yelp2018	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
1	**0.3942**	**0.2510**	**0.3618**	**0.3417**	0.1255	0.0800	0.0723	0.0470
2	0.3764	0.2412	0.3473	0.3312	0.1276	0.0811	0.0765	0.0496
3	0.3804	0.2355	0.3421	0.3274	**0.1289**	**0.0822**	**0.0790**	**0.0507**

- Removing the neighborhood negative sampling module from four datasets will significantly reduce the model's recommendation performance. Because the neighborhood negative sampling module can provide a better negative sampling domain for the model, greatly reducing the possibility of sampling false negative samples, which can effectively improve the model's convergence and recommendation performance. It directly illustrates the effectiveness of the neighborhood negative sampling module.

4.4.3. Influence of aggregation layers

The number of aggregation layers of the CKNNS model is empirically compared in this study, and the number of layers L of the model is searched in the range of {1, 2, 3}, and the findings are provided in Table 5 for analysis.

- In theory, stacking more aggregation layers can better capture user activity patterns, making it easier to model users' preferences. This is mirrored in the Alibaba-ifashion and Yelp2018 datasets. The Recall@20 and NDCG@20 values of CKNNS rise with the superposition of layers.
- The best effect can be achieved by stacking one aggregation layer for the phenomena on the Last-FM and Movie-100k dataset, which may be because the interactions information of users in Movie-100k dataset is more perfect than Alibaba-ifashion and Yelp2018. In addition, Last-FM has more data information, and its corresponding knowledge graph has more relationships, which can better capture user behavior patterns. So only one aggregation layer is enough to model user preferences in Last-FM and Movie-100k.

4.4.4. Influence of K value in top-k recommendation task

Different values of k in top-k recommended tasks are taken for experimental comparison, and the results are shown in Table 6.

According to the experimental results on four datasets, we can find that with the increase of k, the index values of CKNNS also rises steadily, which indicates the stability of our proposed CKNNS to a certain extent.

4.4.5. Influence of embedding dimension

Different embedding dimensions {32, 64, 128} are compared experimentally, and the results are shown in Table 7.

By comparing the different embedding dimensions of vectors, we find that when the embedding dimension is 64, CKNNS can achieve the best results on Last-FM, Movie-100k and Yelp2018. It may be because there are more relationships in knowledge graph of Last-FM, while Movie-100k has the densest historical interaction, and Yelp2018 has the most triples. All these can provide more data information for CKNNS, so the model efficiency can be optimized when embedding dimension 64, while higher embedding dimension will bring more noise, and too low embedding dimension will not carry enough information. In addition, Alibaba-iFashion does not have the characteristics of the above datasets, so CKNNS can achieve better results by increasing the vector embedding dimension from 64 to 128, but it has little effect and needs a lot of calculation cost.

5. Conclusion and future work

In this paper, a recommendation algorithm based on neighborhood negative sampling and collaborative knowledge-aware (CKNNS) is proposed, which not only uses the graph structure information brought by the knowledge graph to enhance the semantic representation of the item, but also captures the item's high-order collaborative information from the historical interactions between users and items, enriching the item's potential features. The gating module also allows the model to adaptively combine these two representations to better model user preferences.

At the same time, the neighborhood negative sampling method is used to provide high-quality negative samples for the

Table 6
Impact of different K value in top-k recommendation task.

	Last-FM		Movie-100k		Alibaba-iFashion		Yelp2018	
	Recall@K	NDCG@K	Recall@K	NDCG@K	Recall@K	NDCG@K	Recall@K	NDCG@K
20	0.3942	0.2510	0.3618	0.3417	0.1289	0.0822	0.0790	0.0507
40	0.4899	0.2763	0.4927	0.3792	0.1894	0.1012	0.1262	0.0659
60	0.5447	0.2888	0.5814	0.4125	0.2383	0.1117	0.1641	0.0766
80	0.5817	0.2967	0.6464	0.4363	0.2627	0.1194	0.1949	0.0848
100	0.6092	0.3023	0.6979	0.4545	0.2897	0.1254	0.2209	0.0914

Table 7
Impact of different embedding dimensions.

	Last-FM		Movie-100k		Alibaba-iFashion		Yelp2018	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
32	0.3889	0.2418	0.3504	0.3338	0.1259	0.0805	0.0733	0.0462
64	0.3942	0.2510	0.3618	0.3417	0.1289	0.0822	0.0790	0.0507
128	0.3883	0.2422	0.3394	0.3185	0.1302	0.0831	0.0782	0.0509

model, which compensates for the problem of data sparsity and introduces a large amount of negative information for the model, which significantly improves the model's recommendation performance. The experimental findings also validate the suggested algorithm's efficacy. The semantic representation of the item is improved in this paper by using the knowledge graph of the item, but the knowledge graph of the user can also bring a lot of auxiliary information to the model, so introducing the knowledge graph of the user side is also a direction of model improvement. At the same time, when the neighborhood is formed, this study solely employs the random negative sampling method to generate negative samples in the neighborhood negative sampling module. Although it has shown positive outcomes, it does have certain limits. As a result, starting with this aspect, we may develop the model and fully exploit the performance of neighborhood negative sampling.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

References

[1] Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl, Item based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, 2001, pp. 285–295.

[2] Zhabiz Gharibshah, Xingquan Zhu, Arthur Hainline, Michael Conway, Deep learning for user interest and response prediction in online display advertising, Data Sci. Eng. 5 (1) (2020) 12–26.

[3] Aditya Grover, Jure Leskovec, Node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864.

[4] Enrico Palumbo, Giuseppe Rizzo, Raphaël Troncy, Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 32–36.

[5] Rose Catherine, William Cohen, Personalized recommendations using knowledge graphs: A probabilistic logic programming approach, in: Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 325–332.

[6] Binbin Hu, Chuan Shi, Wayne Xin Zhao, Philip S. Yu, 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, 2018, pp. 1531–1540.

[7] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, Tat-Seng Chua, Kgat: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 950–958.

[8] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, Xiyang Liu, 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, 2020, pp. 219–228.

[9] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, Tat-Seng Chua, Learning intents behind interactions with knowledge graph for recommendation, in: Proceedings of the Web Conference 2021, 2021, pp. 878–887.

[10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, Meng Wang, 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, 2020, pp. 639–648.

[11] Shanlei Mu, Yaliang Li, Wayne Xin Zhao, Jingyuan Wang, Bolin Ding, Ji-Rong Wen, Alleviating spurious correlations in knowledge-aware recommendations through counterfactual generator, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 1401–1411.

[12] Yuhao Yang, Chao Huang, Lianghao Xia, Chenliang Li, Knowledge graph contrastive learning for recommendation, 2022, arXiv preprint arXiv:2205.00976.

[13] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, Minyi 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.

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

[15] Hugo Caselles-Dupré, Florian Lesaint, Jimena Royo-Letelier, Word2vec applied to recommendation: Hyperparameters matter, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 352–356.

[16] Yudong Chen, Xin Wang, Miao Fan, Jizhou Huang, Shengwen Yang, Wenwu Zhu, Curriculum meta-learning for next poi recommendation, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2692–2702.

[17] Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi, Evren Korpeoglu, Kannan Achan, Jingrui He, Pure: Positive-unlabeled recommendation with generative adversarial network, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2409–2419.

[18] Weinan Zhang, Tianqi Chen, Jun Wang, Yong Yu, Optimizing top-n collaborative filtering via dynamic negative item sampling, in: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2013, pp. 785–788.

[19] Yu Wang, Zhiwei Liu, Ziwei Fan, Lichao Sun, Philip S. Yu, Dskreg: Differentiable sampling on knowledge graph for recommendation with relational gnn, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 3513–3517.

[20] Can Wang, Jiawei Chen, Sheng Zhou, Qihao Shi, Yan Feng, Chun Chen, Samwalker++: recommendation with informative sampling strategy, IEEE Trans. Knowl. Data Eng. (2021).

[21] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, Tat-Seng Chua, Reinforced negative sampling over knowledge graph for

recommendation, in: Proceedings of the Web Conference 2020, 2020, pp. 99–109.

[22] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, Xuan Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[23] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, Wei-Ying 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.

[24] C. Wang, Y. Zhu, H. Liu, et al., Enhancing user interest modeling with knowledge-enriched itemsets for sequential recommendation, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Managemen, 2021, pp. 1889–1898.

[25] T. Huang, Q. She, Z. Wang, et al., GateNet: gating-enhanced deep network for click-through rate prediction, 2020, arXiv preprint arXiv:2007.03519.

[26] Y. Liu, B. Li, Y. Zang, et al., A knowledge-aware recommender with attention-enhanced dynamic convolutional network, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Managemen, 2021, pp. 1079–1088.

[27] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, Xiuqiang He, Simplex: A simple and strong baseline for collaborative filtering, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 1243–1252.

[28] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, Lars Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, 2012, arXiv preprint arXiv:1205.2618.

[29] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, Tat-Seng Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.