EasyChair Preprint

№ 2249

Recommendation Algorithm Based on Dual Attention Mechanism and Explicit Feedback

Jia-Le Li, Zhi-Juan Du and Jian-Tao Zhou

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

December 25, 2019

Recommendation Algorithm Based on Dual Attention Mechanism and Explicit Feedback

No Author Given

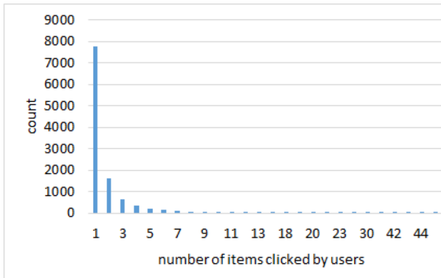
No Institute Given

Abstract. The recommendation algorithms are popular in intelligent applications, and the algorithms seamless integration with the knowledge graph has attracted much attention in recent years. However, the existing methods can not make full use of auxiliary information in KG and characterize user-item interaction behavior violence which lead to the recommendation algorithms are still limited by sparse or even cold start issue, and the recommendation results are weakly interpretable. To address the problem, this paper proposes an enhanced CTR recommendation algorithm based on knowledge graph dual attention mechanism and user explicit feedback. Here,(1) The dual attention mechanism is for KG, which can be divided into inter-item attention mechanism and inter-layer attention mechanism. The inter-item attention mechanism calculates the correlation between the item clicked by the user and the entity connected to the item in the KG. Meanwhile, the inter-layer attention mechanism calculates the correlation between different hops in the KG. (2) The user explicit feedback is the user's degree of preference for the item expressed in numerical form and is used to quantify user-item interaction behavior. Finally, in order to evaluate the usability of our proposed method, three datasets with different sparsity, another three datasets which rich in new items, and a new evaluation task-interpretive visualization were designed to conduct multi-view experimental verification.

Keywords: Click-through rate · Recommendation · Knowledge graph · Explicit rating · Attention mechanism.

1 Introduction

Recommendation algorithms are very popular in the era of big data, especially in intelligent applications. The recommendation algorithm aims to infer items that users might like. This conjecture is not based on a guess but based on the user-item interaction. For example, Tom clicks $Frozen$, and it can be inferred that Tom likes $Frozen$ movies, and even that Tom likes $Animate$d movies. To this end, CTR prediction models have emerged. However,many items are rarely clicked or even not clicked in reality. According to statistics, there are 65,000 users and 300,000 items in the classic Book-Crossing dataset, but nearly 8000 users have only one interaction history, and 1500 users have two interaction records, as shown in Fig.1.



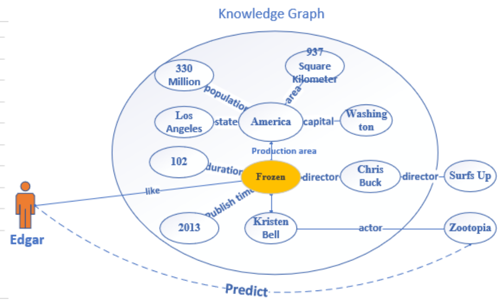


Fig. 1. Statistics of user-item interactions **Fig. 2.** Example of prediction in the KG

We define the case with no user-item interaction record as a cold start issue, and the case with little user-item interaction record as a sparse issue. Such problems usually require additional information, such as user or item attribute information. KG contains rich structured information and relationship information between entities. It usually uses triples to represent facts,like $< Frozen, actor, KristenBell >$.It is likely to include the background knowledge of the item and the link between user and item. Therefore, KG can be used as auxiliary information for user-item interaction. As shown in Fig.2, the user to be predicted has no direct relation with item $Zootopia$, that is, there is no interaction record between user and item $Zootopia$, but there is an interaction record between user and item $Frozen$, in addition, $Frozen$ and $Zootopia$ in KG are directed by $KristenBell$. Therefore, indirectly inferred that users might like item $Zootopia$. This will not only alleviate the cold start issue and the sparsity issue but also enhance the interpretability of the recommendation results.

However, there are many entities in the KG, and not every entity is liked by users. For example, in Fig.2, the user likes the movie $Frozen$, whose director is $ChrisBuck$, and the recommendation algorithm can infer the movie that the user likes from other director's works. In other words, the director's information has a positive impact on predicting user preferences. The $releasetime$ and $duration$, which are directly connected to $Frozen$, are not paid attention[19] to by the user and may cause negative effects. The attribute information of the entity $America$ indirectly connected to $Frozen$ in Fig.2 and the entity $ChrisBuck$ also play different roles, that is, entities at different layers will have different effects on building user preferences.Therefore, identifying which facts or entities are helpful to specific users and items is the key to success.

In addition, user-item interaction behavior is not only measured by clicked or not. It also includes the users' subjective evaluation of items,that is, the user's explicit rating. For example, in the MovieLens dataset, users score 5 points for a movie, indicating that the user likes the movie very much, and movies rated 5 are more preferred than movies rated less than 5.Considering the users' subjective evaluation helps to quantify the user-item interaction behavior, thereby portraying more accurate user preferences.

To this end, this paper proposes an enhanced CTR recommendation algorithm based on **K**nowledge **G**raph **D**ual **A**ttention **M**echanism and user explicit feedback(KGDAM).Here: (1) The dual attention mechanism is for KG, and is divided into the inter-item attention mechanism and the inter-layer attention mechanism. The inter-item attention mechanism calculates the correlation between the item clicked by the user and the entity connected to the item in the KG, and the inter-layer attention mechanism calculates the correlation between different hops in the KG. (2) The user's explicit rating is the degree of user's preference for an item expressed in numerical form and is used to quantify user-item interaction behavior.In summary, the contributions of this paper are as follows:

•**Quantify user-item interaction behavior**: Introduce the concept of explicit rating, and the user-item interaction behavior is quantified. This can prevent the problem of user preferences are complete consistency in the case of sparse data.

•**Make the best of auxiliary information**: Using dual attention mechanism to select more relevant items from a large number of the KG information of a given item. On the one hand, it can reduce the calculation cost, on the other hand, it can reduce the interference of negative or irrelevant information, and improve the accuracy and interpretability of the recommendation.

•**Multi-view experimental verification**: In order to evaluate the usability of the proposed method, three datasets with different sparsity degree, three datasets with new items, and a new evaluation task-interpretive visualization were designed.

The rest of the paper is organized as follows. Section 2 describes related work of the recommended method.Section 3 defines the symbolic representation of this paper.Section 4 expounds our model. Section 5 presents experimental results and analyzes its ability.Section 6 concludes the paper.

2 RELATED WORK

In this section,we review some existing work relevant to our paper. Knowledge graphs are used to make recommendations in three categories: feature-based recommendation methods, path-based recommendation methods, and feature-based recommendation methods.

Early works on recommender algorithm to alleviate data sparse and cold start problems usually typically use Factorization Machines(FM) algorithm,which combines features from the perspective of click-through rate prediction, considering the relation between features, but without considering the deep-seated relation between multiple features. Some studies[6, 2, 13, 3, 14, 15] integrates deep neural network model to solve the problem that FM model can only express two sets of combination between features.In order to further alleviate these problems, researchers want to make full use of data features,using KG to achieve better recommendation results[5, 4].

Feature-based recommendation methods represented by LibFM[9, 7],which unifies the attributes of users and items as the input of recommendation algorithm, but it is not designed specifically for KG, so it can not use all the information of KG efficiently. For example, this kind of method is difficult to utilize multi-hop knowledge.

Path-based recommendation method represented by PER and MetaGraph[18, 16]. This method regards KG as a heterogeneous information network, and then constructs meta-path or meta-graph-based features between items.For example, the *actor* → *movie* → *director* → *movie* → *actor* meta-path can connect two actors, so it can be seen as a way to explore the potential relation between actors. This kind of method can make full use of the network structure of KG, but it needs to design meta-path or meta-graph manually.

The Knowledge Graph Embedding(KGE) recommendation models[17] represented by distance-based translational models,which can learn each entity and relation to get a low-dimensional vector, while maintaining the original structure or semantic information in the KG.For example, the TransE model uses a similarity-based scoring function to evaluate the probability of a triple,the tail vector is regarded as the translation result of the head vector and the relation vector.RippleNet[11] and KGCN[12] use the KGE method,which fully capture the hierarchical structure information of the KG to achieve better effect.DKN[10] and ACSR[8] use attention mechanisms to portray user preference.

3 PROBLEM FORMULATION

In order to introduce our recommendation method more clearly, we will give a detailed explanation of the problem before formally introducing our method. Our purpose is to use KG to assist recommendation.

The recommendation algorithm is to predict the user's preference for items based on the degree of user-item interaction, where the degree is measured by the probability that the user may click on the item. The interaction strength of all users and items is represented by the user-item interaction matrix $Z_{ui} \in \mathbb{R}^{M \times N}$, and the matrix elements are divided into three types: 0 indicates that the user has not clicked on the item, 1 indicates that the user has clicked on the item,R_{ui} indicates that the user has rated the item,R_{ui} and 1 does not appear at the same time. Here, a matrix containing only 0-1 is called a user-item interaction implicit feedback matrix, and is written as Z_{ui-im}, and a matrix containing only 0-R_{ui} is called a user-item interaction explicit feedback matrix, and is written as Z_{ui-ex} . Therefore, the recommended formal definition is as defined in 1.

Definition 1 (Recommendation) *Let* $\mathrm{U} = \{u_1, u_2, ..., u_M\}$ *denote a set of* M *users and* $\mathrm{I} = \{i_1, i_2, ..., i_N\}$ *denote a set of* N *items.*R_{ui} *is the rating of user* u *on item* i*,the implicit and explicit rating as Eqs.*(1) *and* (2)*.*

$$\mathrm{Z_{ui-im}} = \begin{cases} 0, & \text{if interaction(u,i) is not observed;} \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

$$\mathrm{Z_{ui-ex}} = \begin{cases} 0, & \text{if interaction(u,i) is not observed;} \\ \mathrm{R_{ui}}, & \text{otherwise.} \end{cases} \tag{2}$$

Then the probability of user clicking is $\widehat{Z}_{ui} = F(u, i|\theta, Z)$

The KG is a directed graph composed of countless triples. The triple is composed of two entities and a relations and is set to h,t,r.The triple is expressed as (h, r, t).For example,(United States, Capital, Washington) in Fig.2.Therefore,the KG can be formalized as definition 2.

Definition 2 (Knowledge Graph) *Given a graph G= (X,Y), where* X, Y *is the set of entities and relations,*h, t, r *form a triple* (h, r, t)*,then* $G = \{X, Y\} = \{(h, r, t)|h, t \in X, r \in Y\}$

KG and our item or user are connected by entity linking,for example, the item $Terminator$ also appears in the KG as an entity with the same name(entity disambiguation has been done). Therefore, for an item, we can get all its related KG triples through a KG entity.

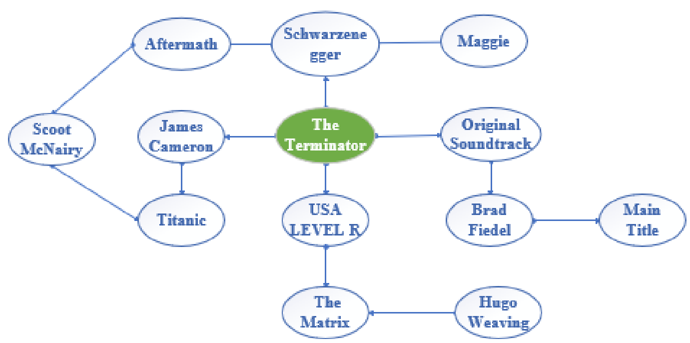


Fig. 3. Illustration of each hop context entity sets

As shown in Fig.3, a user browses the movie $Terminator$, whose 1-hop context entity includes the actor $Schwarzenegger$, the director $Cameron$, the movie ' s rating level R, and the 2-hop context entity is linked with the 1-hop, such as the movie $TheMatrix$,$Titanic$,$Maggie$,etc. For more formal representation of the context entity after q-hop, we recursively define the set of q-hop context entities for user u as definition 3.

Definition 3 (Context Entity Set[1]) *Given a item entity* h_0 *,the set of q-hop context entity is defined as the set of entities which are pointed to in the KG starting from* X_u^{q-1}.

$$C_u^q = \{(h_{q-1}, t)|(h_{q-1}, t) \in G \text{ and } h_{q-1} \in X_u^{q-1}\}, q = 1, 2, .., Q \tag{3}$$

where X_u^0 is the set of user's u clicked items in the past,which can be seen as the central entity set of user u in KG and C_u^1 is the set of context entities starting from $X_u^0.t$ is a directly connected entity of h_{q-1}.This gives a set of context entities associated with the user's click history.

Definition 1 is only defined based on user-item interaction. KG is added to form a recommendation for KG. The formal definition is shown in definition 4.

Definition 4 (KG Recommendation) *Let* $\mathrm{U} = \{u_1, u_2, ..., u_M\}$ *denote a set of M users and denote* $\mathrm{I} = \{i_1, i_2, ..., i_N\}$ *a set of N items.*R_{ui} *is the rating of user* u *on item* i*,the implicit and explicit rating as Eqs.(1) and (2).the probability of user clicking is* $\widehat{Z}_{ui} = F(u, i|\theta, Z, G)$

.

4 Methodology

According to section 1-3,the KG can help for new items and sparse user-items interaction. Through a series of analysis, it can be seen that 1-hop and q-hop entities play different roles on user-item and they can not be replaced with each other. Therefore, this paper combines the above two as user-item auxiliary knowledge.Since different entities can have a positive or negative impact on acquiring user preferences,a dual attention mechanism,which is divided into inter-item attention mechanism and inter-layer attention mechanism, will be introduced to describe detailed user preferences in combination with explicit user ratings.

4.1 Framework

The KGDAM model is shown in Fig.4 (only the 1-hop detailed process is given). The model needs to input the user, the item, and the user's explicit rating on the item to obtain the user's click probability to the candidate item. For each user, the item that user clicked(orange rectangle in Fig.4) is used as an central entity in the KG(orange circle).The entity is surrounded by rich context entity set(A,B,C,D,E) that may be of interest to the user, and these entities can be a set of entities of any hop in the KG.

For the context entity sets of different hops,the unique preference vector T_u of user to the context entity is obtained by dual attention operation,which is shown in Fig.5,will be described in detail in 4.3. Based on the user's explicit rating,different weights are determined for T_u and the central entity, and the user's preference vector E_u is obtained by add.Finally calculate the predicted probability $\widehat{Z}_{ui}$.

[1]Some researches only use the 1-hop to q-hop information in the KG. Users pay different attention to each hop, and can't simply think that users have the same attention to all information.Our model focuses on the correlation between entities.

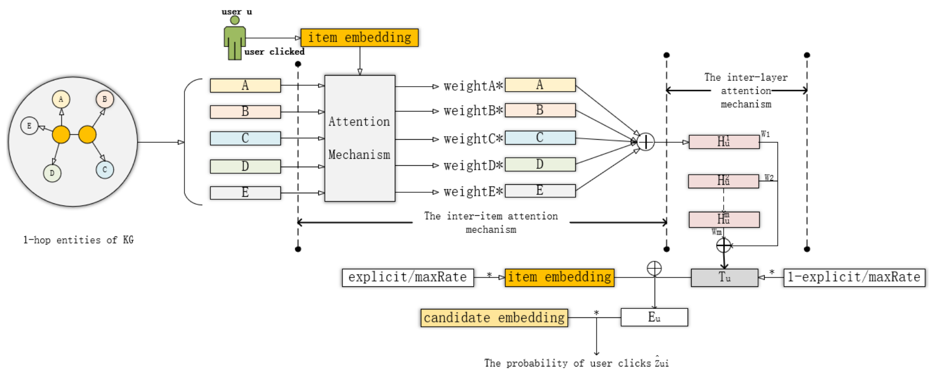


Fig. 4. The KGDAM model diagram(only the 1-hop detailed process is given).

4.2 User Explicit Feedback

The user's explicit feedback is the subjective evaluation of the item by the user, indicating the user's preference for the item. We use the Eq.(4) to measure the degree of user preference.

$$pd = \frac{R_{ui}}{\Phi} \tag{4}$$

where Φ is the upper limit of the explicit rating. For example, in the MovieLens, the upper limit of the rating that the user can give is 5.0, and the upper limit of the rating in the Book-Crossing is 10.0.

4.3 Dual Attention Mechanism

In order to distinguish each context entity, which can be given different degrees of attention.Given the q-hop context entity set C_u^q of user u, each entity $t_u^{q_k}$ in C_u^q is assigned a different weight a_{q_k}.As shown in Fig.4, each item user clicked is associated with an item embedding, and the context entity in the KG is in the different vector space with the item.The user preference for each hop context entities H_u^q are shown in Eq.(5).

$$H_u^q = \sum_{q=1}^{Q} a_{q_k} \cdot t_u^{q_k} \tag{5}$$

where $t_u^{q_k}$ donates the k context entities in q-hop,H_u^q is the user embedding for each hop context entities,a_{q_k} is the weights of each context entities and a_{q_k} is defined in Eq.(6).

$$a_{q_k} = \delta(f(h_0, t_u^{q_k})) \tag{6}$$

where $\delta(x) = \frac{1}{1+e^{-x}}$,$h_0$ represents the entity embedding in the KG corresponding to the item clicked by the user u.Function f is used to calculate the

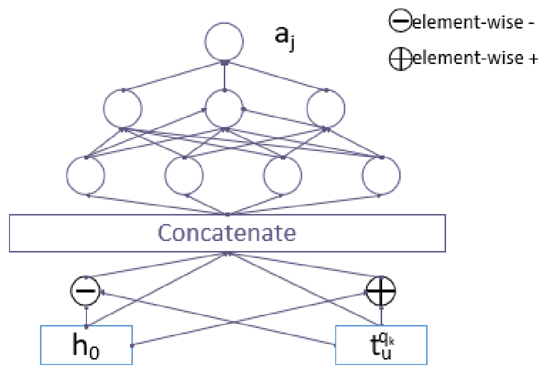


Fig. 5. The inter-item attention mechanism of KGDAM

specific weights and obtained as shown in Fig.5. It sends each item h_0 that the user clicks and its corresponding each hop context entity $t_u^{q_k}$,$h_0-t_u^{q_k}$ and $h_0+t_u^{q_k}$ into the neural network, and finally outputs the weights of each entity a_{q_k}.At present, the user preference of each hop H_u^q has been calculated.if user preference of each hop H_u^q are treated equally, it is obviously unreasonable to describe the user preference in detail.For example,each hop contains different context entities, and users may have different preferences for the entities of each hop, as shown in Fig.3. The 1-hop contains director *Cameron*, actor *Schwarzenegger*, etc. the 2-hop includes *Titanic* which directed by *Cameron*, etc.User may prefer the 2-hop entities.Inspired by the non dominated sorting stage of multi-objective genetic algorithm[1], different weights will be given to each generation of population in order to find better individuals.In this way,we use attention mechanism again to get the user preference for the context entities T_u in the KG, as shown in Eq.(7).

$$T_u = \sum_{q=1}^{Q} w_q \cdot H_u^q \tag{7}$$

where Q is the number of hop,T_u is the user embedding for context entities,w_q is the weights of each hop.In this paper, w_q is obtained by learning.

As mentioned in introduction, the user explicit rating can reflect the user's fine-grained preference, and more accurately learn the user preference. User preference is obtained by Eqs.(8) and (9).

$$E_u = (\frac{R_{ui} + threshold}{\Phi + threshold}) \cdot h_0 + (1 - \frac{R_{ui} + threshold}{\Phi + threshold}) \cdot T_u \tag{8}$$

$$E_u = \sigma(E_u) \tag{9}$$

where E_u is the user preference embedding,φ_n is an explicit rating of the item the user clicked,σ is the activation function relu,which is defined as Eq.(10):

$$\sigma(x) = max(0, x) \tag{10}$$

we have designed a threshold to indicate whether the user likes it,the purpose is to treat items with a rating of 0 differently, because if there is no threshold in Eq.(8), items with a rating of 0 will be discarded directly, which is not in line with the actual situation.For example, in the MovieLens dataset, the threshold can be set to 2, ie, an item with a rating greater than or equal to 2 indicates that the user like the item.

Defining user preference in this way has two advantages:(1)From Eq.(8), it can be seen that user embedding E_u consists of two parts, namely, the embedding of user clicked items and the embedding of context entity information T_u. When user click less items, they can use the rich context entities in the KG as auxiliary information. To a certain extent, they can learn more accurate user embedding and then alleviate the problem of data sparsity and cold start. (2)Eq.(8) also reveals that the user's explicit rating as a quantified value can depict detailed user preference(user embedding). When the user has a max rating for an item, the user embedding can be directly replaced by the embedding of the item. On the contrary, it will rely on the information provided by KG to build user preference. In this way, the focus of user embedding is put on the user's subjective preference for the items. So as to get accurate user preference embedding.

4.4 Learning Algorithm

As mentioned in Section 4.1, given the interaction sequence $\{i_1, i_2, ..., i_N\}$ and explicit rating of user u, each item in the user's click sequence act as the central node of the KG.Starting from the central node, storing context entities with only 1-hop distance,2-hop distance and so on,it can obtain a set of Q hop context entities. In this paper, the user clicks on the item and the context entity of the KG in the same vector space, initializes each entity vector, and the dimension of the vector can be set by itself. After dual attention and combined with explicit rating, the user preference embedding E_u is calculated.Then, the user preference embedding E_u and item embedding I are fed into function $f : R \times R \to R$ represent the predicted clicking probability,as shown in Eq.(11).

$$\widehat{Z}_{ui} = f({E_u}^T I) \tag{11}$$

where δ is the sigmoid function,which can normalize $\widehat{Z}_{ui}$. For entity vectors and other parameters,to make computation more efficient.The complete loss function is shown in Eq.(12).

$$\min L = \sum_{(u,i)\in Z} -(Z_{ui} \cdot \log \delta(\widehat{Z}_{ui}) + (1 - Z_{ui}) \cdot \log(1 - \delta(\widehat{Z}_{ui}))) + \lambda||X||_2^2 + \lambda||w_i||_2^2 \tag{12}$$

where L is cross-entropy loss between the truth of interactions Z and predicted value,The rest term is the L2-regularizer for preventing over-fitting.Then, the problem of minimization (L) can be solved by using Adam algorithm, which is also proposed in some models[12], Adam uses the global learning rate η to update all parameters.The formal description of the above steps is presented in Algorithm 1.

4.5 Links To The Existing Work

In this section,we select DKN, KGCN, and RippleNet which are specifically designed for knowledge graph recommendation and are all CTR models and the recommendation tasks are to predict the click probability.we will theoretically compare our model KGAM them.

•**Explicit ratings**:Four models are CTR models, all of them use the implicit feedback (click and non click), but only KDGAM combines explicit and implicit feedback to obtain detailed user preferences.

Algorithm 1 KGDAM algorithm

input: $Interaction\ matrix\ Z,\ knowledge\ graph\ G;$
output: $Prediction\ function\ F(u, i, implicit, explicit|\theta, Z, G)$

$Initialize\ all\ parameters$
Calculate context entity set $C_u[q]$ for each user u
for (u,i,implicit,explicit) **do**
  ∀ e ∈ $C_u[0]$
  for $q = 1 \rightarrow Q - 1$ **do**
    for $e \in C_u[q]$ **do**
      $H_u^q \leftarrow \sum_{k=1}^{K} a_{q_k} \cdot e$
    end for
    $T_u \leftarrow \sum_{q=1}^{Q} w_q \cdot H_u^q$
  end for
  $E_u = \frac{explicit_j}{\Phi} \cdot i_j + \frac{1-explicit_j}{\Phi} \cdot T_u$
  $Calculate\ predicted\ probability\ \widehat{Z}_{ui} = \delta(f({E_u}^T I))$
end for
return $result$

•**Inter-item attention mechanism**:KGDAM uses the inter-item attention mechanism.KGCN does not use the attention mechanism,RippleNet calculate the similarity as the attention mechanism between the embedding of the clicked item and the corresponding entity and its relation in the KG , which is different from KDGAM.DKN uses attention mechanism between news to be recommended and news clicked by users and directly merges the two as input to the neural network.

•**Aggregator**:KGDAM aggregates information from clicked items and KG to alleviate data sparseness.KGCN uses inward aggregation to take the relation and entity information in the KG as the attributes of the user's clicked item to construct the item embedding.DKN and RippleNet do not user it.

•**Inter-layer Attention mechanism**:KGDAM uses inter-layer attention mechanism to distinguish entities between different levels, the other three models are not considered.

•**Relations**:KGDAM does not use the relation in the KG, but only the entity information. The other three models are all used.

•**Distinguish vector space**:In KGDAM, RippleNet and DKN, the item embedding and the KG embedding are in the different vector space.In KGCN, the item embedding and the KG embedding are in the same vector space.

In order to more intuitively reflect the difference between the above models, it is shown in Table 1.

Table 1. Comparison of three models.

	DKN	RippleNet	KGCN	KGDAM
Implicit ratings	√	√	√	√
Explicit ratings	−	−	−	√
Inter-item Attention mechanism	√	√	−	√
Aggregator	−	−	√	√
Inter-layer Attention mechanism	−	−	−	√
Relations	√	√	√	−
Distinguish vector space	√	√	−	√

√ donates the comparison term has been used in the model
− donates the comparison term has not been used in the model

5 EXPERIMENTS

In this section, our approach, KGDAM, is evaluated for the CTR prediction scenario.The specific evaluation indicators are AUC, F_1.

5.1 Experiments Setup

Datasets : We use the following three scenarios in our experiments for movie, book, and music recommendation, respectively:MovieLens-20M, Book-Crossing, Last.FM and corresponding KG data,which is also proposed in some models[12].

Implementation and Baselines: Then we marked an unwatched set as 0 for each user.Through sampling, items marked as 1 are taken as positive feedback data, and items marked as 0 are taken as negative feedback data and both are the same size.The basic statistics of the three datasets and the hyper-parameter ranges are provided in Table 2,where K denotes the size of the selected context entity set,Q denotes the number of hops.LIke[13],we also select SVD[7],LibFM[9], PER[16],CKE[17],RippleNet[11] and KGCN[12] as baselines.

Evaluation Metrics:In KGDAM,we use the evaluation environment in[12]. For each dataset, the ratio of training, evaluation, and the test set is $7:1:2$.Each experiment repeated 5 times and reported the average performance,all trainable parameters and embeddings are optimized by Adam algorithm.We also test the model for CTR prediction, the test indicators are AUC, F_1.The specific tuning hyper-parameters results are given in the next section.

Table 2. Basic statistics and hyper-parameter of the three datasets.

	MovieLens-20M	Book-Crossing	Last.FM
users	138159	19676	1872
items	16954	20003	3846
entities	102,569	25,787	9,366
relations	32	18	60
KG triples	499474	60787	15518
K	1∼20	1∼20	1∼20
Q	1∼3	1∼3	1∼3

5.2 Results

The results of all methods in CTR prediction are presented in Table 3.Several observations stand out:

Table 3. The results of AUC and F_1 in CTR prediction.

Model	MovieLens-20M		Book-Crossing		Last.FM	
	AUC	F_1	AUC	F_1	AUC	F_1
SVD	0.963	0.919	0.672	0.635	0.769	0.696
LibFM	0.959	0.906	0.691	0.618	0.778	0.710
PER	0.832	0.788	0.617	0.562	0.633	0.596
CKE	0.924	0.871	0.677	0.611	0.744	0.673
RippleNet	0.968	0.912	0.715	0.650	0.780	0.702
KGCN	0.978	0.932	0.722	0.682	0.794	0.712
Without Attention	1.000	0.995	0.750	0.671	0.930	0.827
Without Explicit	0.961	0.901	0.743	0.669	0.794	0.731
KGDAM	**1.000**	**0.998**	**0.758**	0.672	**0.934**	**0.839**

•On the whole,KGDAM uses both user explicit rating and dual attention mechanisms, it has various degrees of improvement in CTR scenarios, and performs best in all baselines besides F_1 of Book-Crossing.This is because KGDAM uses explicit rating, but a large number of books in Book-Crossing have a score of 0, which affects F_1.This shows that explicit rating and attention mechanism can describe fine-grained user preferences and alleviate the data sparsity issue.

•Compared with other baselines, the performance of CKE is poor, probably because the auxiliary information provided by the KG is not fully utilized.

•SVD and LibFM are not specially designed for KG, and they can not use all the information of KG efficiently, but the performance of SVD and LibFM is better than PER, which shows that PER can not make full use of KG through the meta-path designed manually.

•RippleNet is a knowledge graph embedding model, which is a significant improvement over traditional feature-based and path-based methods. Therefore, knowledge graph embedding methods and capturing proximity information in KG are essential for recommendations.

•KGCN has been improved based on RippleNet, with emphasis on the semantic relation in the KG, showing strong performance.

Next, we will change the size of the context entity set in each hop. Table 4 lists the AUC results for these three datasets.When K is very small, a small amount of context information is not enough to describe user preferences and solve the problem of data sparsity. When K is large, KG contains a lot of irrelevant information, which interferes with the generation of user preferences.In summary,different datasets should choose the appropriate K.

Table 4. The results of KGDAM in different sizes of user context entity set.

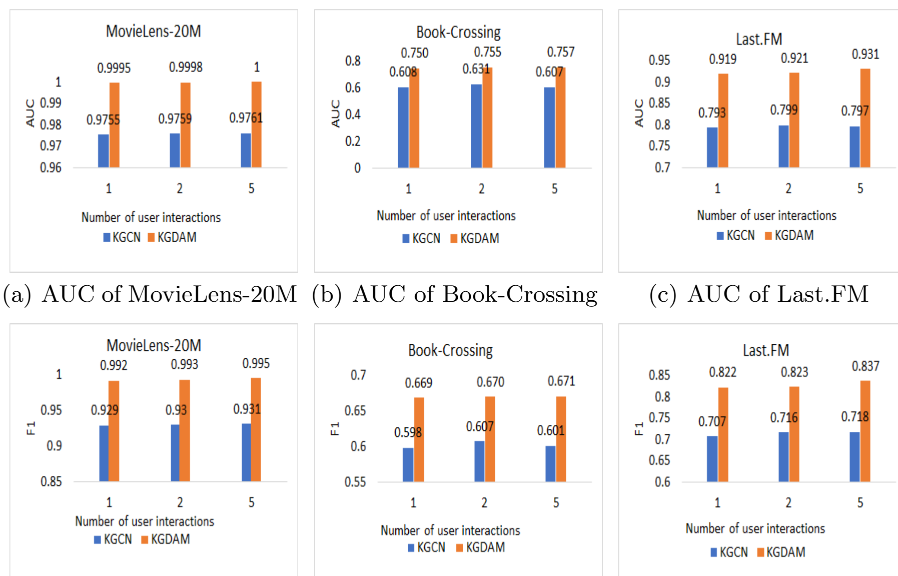
K	MovieLens-20M		Book-Crossing		Last.FM	
	AUC	F_1	AUC	F_1	AUC	F_1
2	0.9989	0.9942	0.7508	0.6680	0.9238	0.8353
4	0.9992	0.9953	0.7528	0.6705	0.9275	0.8362
6	**1.0000**	**0.9998**	0.7521	0.6716	0.9346	0.8365
8	1.0000	0.9993	0.7554	0.6718	**0.9383**	**0.8392**
10	1.0000	0.9649	**0.7586**	**0.6720**	0.9306	0.8354
20	0.9994	0.9956	0.7487	0.6710	0.9236	0.8341

We test the effect of changing the maximum hop number Q on the model. The results are shown in Table 5. When Q is 2, the performance is the best in movie and music scenario. This is because we use the attention mechanism to set the weight of each hop in the model, which will pay more attention to the context entity most relevant to the seed node.According to the experimental results, Q of 1 or 2 is enough to achieve good results.

Table 5. The AUC of different hops of KGDAM

M	MovieLens-20M	Book-Crossing	Last.FM
1	1.0000	0.7621	0.9317
2	1.0000	0.7618	0.9383
3	1.0000	0.7581	0.9333

KGCN is better than RippleNet in solving the problem of data sparseness[11]. Therefore, we only compare the model effects of KGCN and KGDAM under different sparsity. This experiment cuts the original data set into different sparsity. Fig.6 and Fig.7 show the comparison of AUC and F_1. The abscissa indicates that each user has only one, two or less and five or less interactions. The results show that KGDAM is better than KGCN in solving data sparse issue.



(a) AUC of MovieLens-20M (b) AUC of Book-Crossing (c) AUC of Last.FM

(d) F_1 of MovieLens-20M (e) F_1 of Book-Crossing (f) F_1 of Last.FM

Fig. 6. AUC and F_1 with different degree of sparsity

5.3 Relevance study

KGDAM sets different weights for each context entity in the KG. Fig.7 shows the partial context entities and the correlation between them. The central entity

Terminator is a movie user clicked, and the value of the connecting line is the correlation between the context entity and the central entity. It can be seen that the context information contains information that users are more interested in, which is conducive to learning fine-grained user preferences.

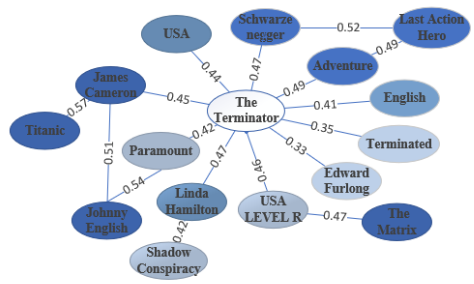


Fig. 7. The relevance of the *Terminator* and its contextual entities

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a dual attention mechanism recommendation model based on KG, which can describe more detailed user preferences by introducing user explicit rating and attention mechanism.Our model is superior to previous methods in terms of effectiveness and outperform state-of-the-art baselines,especially in the movie dataset.We will consider the following three issues in our future work: (1) The semantic relation of the KG will be considered, combined with the user's explicit rating, portraying detailed user preferences and improving interpretability.(2)Using RNN to build user preferences.

References

1. Deb, K., Agrawal, S.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evolutionary Computation **6**(2), 182–197 (2002)
2. Guo, H., Tang, R., Ye, Y.: Deepfm: A factorization-machine based neural network for CTR prediction. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne 19-25, 2017. pp. 1725–1731 (2017)
3. He, X., Chua, T.: Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Japan, August 7-11, 2017. pp. 355–364 (2017)
4. Hu, P., Du., R.: Hybrid item-item recommendation via semi-parametric embedding. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, China, August 10-16, 2019. pp. 2521–2527 (2019)
5. Huang, J., Dou, H.: Improving sequential recommendation with knowledge-enhanced memory networks. In: The 41st International ACM SIGIR Conference on

Research & Development in Information Retrieval, SIGIR 2018, USA, July 08-12, 2018. pp. 505–514 (2018)
6. Juan, Y., Zhuang, Y., Chin, W., Lin, C.: Field-aware factorization machines for CTR prediction. In: Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016. pp. 43–50 (2016)
7. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, USA, August 24-27, 2008. pp. 426–434 (2008)
8. Liu, J., Zhao, P.: Attention and convolution enhanced memory network for sequential recommendation. In: Database Systems for Advanced Applications - 24th International Conference, DASFAA 2019, Thailand, April 22-25, 2019, Proceedings, Part II. pp. 333–349 (2019)
9. Rendle, S.: Factorization machines with libfm. ACM TIST **3**(3), 57:1–57:22 (2012)
10. Wang, H., Zhang, F.: DKN: deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018. pp. 1835–1844 (2018)
11. Wang, H., Zhang, F., Wang, J.: 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 2018, Italy, October 22-26, 2018. pp. 417–426 (2018)
12. Wang, H., Zhao, M.: Knowledge graph convolutional networks for recommender systems. In: The World Wide Web Conference, WWW 2019, USA, May 13-17, 2019. pp. 3307–3313 (2019)
13. Wang, X., Peng, Z., Wang, S.: Cross-domain recommendation for cold-start users via neighborhood based feature mapping. In: Database Systems for Advanced Applications - 23rd International Conference, DASFAA 2018, Australia, May 21-24, 2018, Proceedings. pp. 158–165 (2018)
14. Xiao, J., Ye, H., Chua, T.: Attentional factorization machines: Learning the weight of feature interactions via attention networks. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. pp. 3119–3125 (2017)
15. Xue, H., Dai, X.: Deep matrix factorization models for recommender systems. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Australia, August 19-25, 2017. pp. 3203–3209 (2017)
16. Yu, X., Ren, X.: Personalized entity recommendation: a heterogeneous information network approach. In: Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, USA, February 24-28, 2014. pp. 283–292 (2014)
17. Zhang, F., Yuan, N.J., Ma, W.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, USA, August 13-17, 2016. pp. 353–362 (2016)
18. Zhao, H., Lee, D.L.: Meta-graph based recommendation fusion over heterogeneous information networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Canada, August 13 - 17, 2017. pp. 635–644 (2017)
19. Zhou, G., Zhu, X., Song, C.: Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, August 19-23, 2018. pp. 1059–1068 (2018)