IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 7 February 2023, accepted 21 February 2023, date of publication 24 February 2023, date of current version 1 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3248618

RESEARCH ARTICLE

Optimal Recommendation Models Based on Knowledge Representation Learning and Graph Attention Networks

QING HE, SONGYAN LIU, AND YAO LIU
School of Electronic Engineering, Heilongjiang University, Harbin, Heilongjiang 150080, China
Corresponding author: Songyan Liu (liusongyan@hlju.edu.cn)

ABSTRACT Knowledge representation learning techniques process the knowledge graph, embedding entities and relationships into a continuous dense low-dimensional vector space, and providing the rich semantic association information between entities embedded in the knowledge graph to the recommendation module, improving recommendation performance and providing better interpretability. However, the knowledge representation learning techniques based on the translation model TransD with too many parameters and no association between entity representations are difficult to apply to large knowledge graph, as well as the problem is that most existing knowledge graph which based on recommendation systems ignore the different levels of importance that users attach to different relationships of items. To address these shortcomings, we propose an improved knowledge representation learning model Cluster TransD and a recommendation model Cluster TransD-GAT based on knowledge graph and graph attention networks, where the Cluster TransD model reduces the number of entity projections, makes the association between entity representations, reduces the computational pressure, and makes it better to be applied to the large knowledge graph, and the Cluster TransD-GAT model can capture the attention of different users to different relationships of items. Extensive comparison and ablation experiments on three real datasets show that the model proposed in this paper has significant performance improvements in terms of average ranking, accuracy and recall of the scoring function compared to other state-of-the-art models.

INDEX TERMS Knowledge graph, knowledge representation learning, graph attention networks, personalized recommendation.

I. INTRODUCTION

In the era of information overload, the task of recommendation systems is to connect users and items, on one hand helping users to discover items of value to them, and on the other hand enabling items to be presented to users interested in them, thus achieving a mutual win-win situation for both consumers and producers of items, and recommendation systems play an important role in various online services [1]. As one of the classical recommendation models, the collaborative filtering model simulates user preferences by inner product operation based on a user item rating matrix, and vector embedding of users and items. However, collaborative filtering model suffer from the problem of limited recommendation effect in cold start and data thinning, and it is also difficult to explain the reason for recommendation results [2], [3].

In order to solve the above-mentioned shortcomings, knowledge graph-based recommendation systems have become a hot research topic in recent years, because knowledge graph (KG) contains information on item attributes and various types of relationships, which can provide rich semantic association information of items for recommendation systems [4]. The embedding-based approach embeds the entities and relationships in the KG into a continuous dense low-dimensional vector space, and then obtains a low-dimensional dense vectorized representation, which is then weighted and fused with the item score similarities in the original recommendation system. Currently, there are three main types of KG-based embedding models, the first

The associate editor coordinating the review of this manuscript and approving it for publication was Chang Choi.

being translation-based models, for instance TransE, TransH, TransR, and TransD [5], [6], [7]. However, TransD model has too many parameters and no association between entity representations, which are difficult to apply to the large KG, as well as the existing KG based recommendation models ignore the different importance users usually attach to different relationships of items and cannot intuitively display the attention weights. The second one is semantic matching models, such as DistMult, ComplEx and Rescal [8], [9]. The third is a model that fuses multiple sources of information, using multiple sources of information related to entities and relationships to mine the semantic relevance between entities and build more accurate knowledge representations. However, the problem of how to make efficient use of multiple sources of information is still challenging, and the approach of fusing multiple sources of information is still in its preliminary stage [10].

Recommendation models based on knowledge graphs fall into three main categories. The first is embedding-based models, which generate latent vector representations by embedding structural knowledge directly from the knowledge graph, but tend to ignore the path connectivity of the knowledge graph, making it difficult to interpret the resulting recommendation results, e.g. CKE, DKN, KSR, etc [11]. The second is path-based models, which use path connectivity to standardize entity representations, but require manual definition of the type and number of meta-paths, e.g. Hete-MF, HeteRec, PathSim, etc [12]. However, the type and number of meta-paths need to be defined manually, e.g. Hete-MF, HeteRec, PathSim, etc. The third is graph neural network-based models, which use embedding-based methods to refine the vector representation of the knowledge graph and path-based methods to provide interpretability of the recommendation results, e.g. RippleNet, KGAT, etc [13].

Neural network-based approaches can learn higher-order embedding representations from simple features layer by layer through deep network architectures, however deep networks do not model between user and item information, so using graph attention networks enables entity nodes to pass feature information to neighbouring nodes based on weight coefficients and enables entity nodes to perform embedding aggregation with propagated neighbouring nodes to generate new embedded nodes containing more information, thus enabling the learning of node uniqueness and improving the ability of nodes to embed representations [14].

In this paper, we present for the first time an improved knowledge representation model Cluster TransD (CTransD) and a recommendation model Cluster TransD-GAT (CTransD-GAT) based on KG and graph attention networks. In order to reduce the number of entity projections and improve application to large KG, the CTransD model clusters entities using a clustering algorithm, and the distance relationship between entity classes is converted into a probabilistic representation to solve the problem that the defect of no connection between entity representations after clustering. The CTransD-GAT recommendation model converts the initial embedding vector of the KG through knowledge representation learning, and then obtains the different weight coefficients of the different relationships between users of the items in the KG through the graph attention networks. Then entity nodes pass feature information to neighboring nodes according to the weight coefficients, and make entity nodes and propagated neighboring nodes perform embedding aggregation to generate new embedding nodes containing more information, and get a better graph attention networks recommendation model through scoring prediction and model training.

The main contributions of this paper are as follows:

(1) The knowledge representation learning model CTransD is improved to replace the TransD model for better representation of entity vectors and further more accurate prediction of user ratings, which provides a solid foundation for the CTransD-GAT recommendation model.

(2) A recommendation model CTransD-GAT based on KG and graph attention networks is proposed. The initial embedding vector is obtained from the KG through the CTransD model, and the initial embedding vector is propagated and aggregated with weights.

(3) A large number of comparison and ablation experiments are conducted on three real datasets to prove that the model proposed in this paper outperforms the existing state-of-the-art models.

The rest of this paper is organized as follows: Part II presents related works, which mainly introduces the current research status of knowledge representation models and recommendation models. Part III is the problem definition of the article. Part IV introduces the improved knowledge representation learning model CTransD. Part V presents the overall framework of the CTransD-GAT recommendation model, and details the design of each module of the model, the loss function and the training process. Part VI conducts comparison and ablation experiments as well as analyzes the experimental results. Part VII concludes and outlooks.

II. RELATED WORKS

Knowledge representation learning achieves the representation of semantic association information of entities and relations by projecting them into a low-dimensional vector space, projecting objects from different sources into the same semantic space, and constructing a unified representation space that can efficiently compute entities, relations and the complex semantic associations among them [15]. Knowledge representation learning is divided into three main categories, translation-based models, semantic matching models, and models that fuse information from multiple sources [16]. Since translation-based models have simple training and excellent results, while semantic matching models and fused multi-source information models have high computational complexity and are difficult to balance efficiency and results in large-scale KG, this paper mainly discusses translation-based models.

TABLE 1. Complexity of knowledge representation learning models.

Model	Parameters	Time complexity
TransE	$O(N_e m + N_r n)(m = n)$	$O(N_t)$
TransH	$O(N_e m + 2N_r n)(m = n)$	$O(2mN_t)$
TransR	$O(N_e m + N_r(m+1)n)$	$O(2mnN_t)$
CTransR	$O(N_e m + N_r(m+d)n)$	$O(2mnN_t)$
TransD	$O(2N_e m + 2N_r n)$	$O(2nN_t)$
CTransD	$O((N_e + k)m + 2N_r n)$	$O(2nN_t)$

Borders et al. proposed the simple and efficient TransE model inspired by word2vec language model which vectors have translation invariance in semantic space, but the performance of TransE model is unsatisfactory when dealing with complex relations [17], [18]. ZhenWang et al. proposed the TransH model, which introduces the concept of hyperplane to replace the original relationship vector and considers entities and relationships in the same space, however, entities and relationships are different objective facts [19]. YanKaiLin et al. proposed the TransR model, which projects entities under different relations into different semantic spaces for translation, as well as the Cluster-based TransR (CTransR) model, in which different head-to-tail entity pairs are clustered into groups and different relationship vectors are learned for each group to extend the TransR, based on the idea of segmented linear regression, but the projection matrix is only related to relations [20]. Guoliang Ji et al. proposed the TransD model, where entities and relations use different mapping matrices, and each entity and relation has two representations [21].

The complexity of the knowledge representation learning model are shown in Table 1, where N_e and N_r denote the number of entities and relationships respectively, m and n denote the dimensionality of the entity space and relationship space respectively, N_t denotes the number of triples in the knowledge graph, d denotes the average number of clusters of a relationship, and k denotes the number of clusters of entity clusters.

Recommender system is an information filtering system in the era of information overload, helping users find the information they want in the massive information and reducing the waste of time and energy caused by browsing large amount of invalid data. Recommendation models are mainly classified into collaborative filtering-based recommendation, content-based recommendation, and hybrid recommendation [22], [23].

Amazon proposes an item-based collaborative filtering model that uses the user's behavioral data and the item similarity matrix to calculate offline, which is very efficient and can produce high-quality recommendations, but the user's behavioral data is often overly sparse [24]. The matrix decomposition computational model proposed at the Netflix competition can solve the above problem very well. By matrix decomposition both users and items get their corresponding hidden vectors, but it is not suitable for dealing with large-scale sparse matrices because it's huge computational [25]. Osaka University proposed the factorial decomposition model, which can capture the second-order features well and reduce the computational complexity [26].

In recent years, many researchers have tried to use KG as auxiliary information for recommendations, which can be classified into three types: embedding-based models, path-based models, and graph neural network-based models [27], [28]. Zhang et al. proposed a collaborative knowledge base embedding (CKE) recommendation model that combines knowledge embedding on the basis of collaborative filtering by doing inner product operations of user representation learning and item representation learning to obtain the probability of a user clicking on the item, however, it is easy to ignore the path connectivity of the KG and difficult to explain the generated recommendation results [29]. For the above problem, Sun et al. proposed the PathSim path-based model to recommend for users by using the path connectivity similarity of entities in the KG, but this model requires the considerable manual design of meta-paths, which is labor-intensive and time-consuming [30]. Aiming at solving the above problem, Wang et al. proposed a RippleNet model based on graph neural network, which integrates embedding-based and path-based models to combine the semantic representation of entities, relationships and path connectivity for recommendation [31]. In order to further mine the higher-order relationships of the knowledge graph, Wang et al. proposed the knowledge graph attention networks-based recommendation KGAT model, which uses TransR techniques to obtain the initial embedding vectors of entities, obtains the weights of nodes through graph attention networks, and aggregates and updates user or item embedding vectors in multiple iterations [32].

Graph neural networks use graph convolution to learn representations of node and relationship features in graphs, providing a method for extracting features from non-regular data, extending the processing capability of deep learning for non-euclidean data, and are widely used in various aspects such as social networks, transportation networks, and recommender systems.

In general, the translation-based model in knowledge representation learning is simple and excellent to train and can balance efficiency and results in large-scale knowledge graph. The KG and graph neural network-based in recommendation model propagates and aggregates initial embedding vectors with weights by knowledge representation learning. The limitations of embedding-based and path-based models are overcome, and the potential preferences of users are automatically propagated and the hierarchical interests of users are mined in the KG. Therefore, this paper proposes an improved knowledge representation learning model CTransD and a recommendation model CTransD-GAT based on KG and graph attention networks.

III. PROBLEM DEFINITION

The problems to be solved in this paper are described as follows:

(1) Reduce the number of parameters and computational complexity of the knowledge representation learning model,

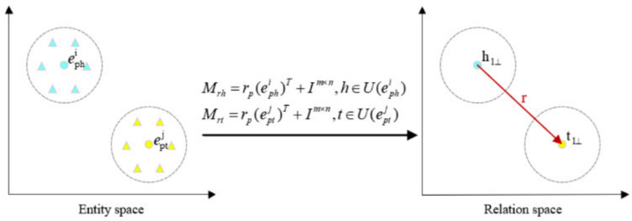

FIGURE 1. **The figure of CTranD model.**

so that it can be better applied to large knowledge graphs and solve the problem of no connection between entity representations after clustering.

(2) The problem of how to comprehensively consider the different levels of importance that users attach to different relationships of items.

(3) A large number of comparison and ablation experiments on real datasets to demonstrate the performance of the model proposed in this paper.

IV. KNOWLEDGE REPRESENTATION LEARNING

In this chapter, the knowledge representation learning model is optimised by node-embedded representations, and a probability distribution in the space of negative examples is constructed to assist the model in collaborative optimisation training to obtain a knowledge representation learning CTransD model.

A. NODE EMBEDDING REPRESENTATION

Compared with the previous translation models, the TransD model takes into account the diversity between entities and relations at the same time, and uses two vectors to dynamically reconstruct the projection matrix, and defines two vectors for each entity and relation to construct the mapping matrix. The TransD model has relatively better performance and excellent representation capability, but too many model parameters with relatively high computational complexity will cause a relative increase in model training time causing overfitting, which is difficult to apply to large KG, as well as the defect of no connection between entities and entity representations in the model. To address this problem, we optimize the TransD model and propose the CTransD model, which uses a clustering algorithm to cluster entities to reduce the number of entity projections, and uses Euclidean distance to represent the similarity between entity classes, and converts the distance relationship between entity classes into a probability representation to solve the defect of no connection between entity representations after clustering, and the CTransD model as illustrated in Figure 1.

In the CTransD model for each triplet (h, r, t), set h to be the head entity vector, r to be the relationship vector, and t to be the tail entity vector, and cluster into K classes based on the similarity between the entity vectors by the K-Means clustering algorithm, each entity vector belongs to and only belongs to a class cluster with the smallest distance to its entity vector class cluster center.

The mean value of the cluster is calculated using the method of arithmetic mean and this mean value is taken as the entity class center, the center of the head entity is denoted as e^i_h, the neighborhood $U(e^i_h)$ formed by the head entity, the center of the tail entity is denoted as e^j_t, the neighborhood $U(e^j_t)$ formed by the tail entity, $i, j \in \{1, 2, \ldots, k\}$ [33], [34]. The head and tail entity class centers are defined as:

$$e^i_h = \frac{\sum_{h\in(U(e^i_h))} h}{N(U(e^i_h))},\ e^j_t = \frac{\sum_{t\in(U(e^j_t))} t}{N(U(e^j_t))} \tag{1}$$

the entity class centers e^i_h and e^j_t correspond to the projection vectors e^i_{ph} and e^j_{pt}, respectively, and the entities are projected from the entity space to the relationship space, set the two mapping matrices M_{rh}, M_{rt} be:

$$\begin{aligned} M_{rh} &= r_p(e^i_{ph})^T + I^{m\cdot n}, h \in U(e^i_{ph}), M_{rt} \\ &= r_p(e^j_{pt})^T + I^{m\cdot n}, t \in U(e^j_{pt}) \end{aligned} \tag{2}$$

the projected head entity $h_\perp$ and tail entity $t_\perp$ in the relationship space are denoted as:

$$h_\perp = M_{rh}h, \quad t_\perp = M_{rt}t \tag{3}$$

the head entity $h_\perp$ and the tail entity $t_\perp$ projected into the plane of the relation r satisfy $h_\perp + r - t_\perp \approx 0$, and the triplet score function is:

$$f(h, r, t) = -\left\|h_\perp + r - t_\perp\right\|_2^2 \tag{4}$$

In the experiment, we force the constraints to be $\|h\|_2 \le 1$, $\|t\|_2 \le 1$, $\|r\|_2 \le 1$, and $\|t_\perp\|_2 \le 1$.

B. NEGATIVE TRIPLET CONSTRUCTION STRATEGY

In the optimization process of the objective function, not only the correct triplets but also the incorrect triplets are needed, and the KG already contains all the correct triplets. The negative sampling algorithm used in the TransE model is to replace the head or tail entity by a randomly selected entity from the set of all entities to obtain a new triplet, which is considered as a negative example triplet. However, in one-to-many, many-to-one and many-to-many relationships, the probability that the constructed triplet is not a negative example triplet is higher. To address the above drawbacks, Bernoulli sampling algorithm is used to select entities in classes with large class spacing to replace the head or tail entities in order to improve the differentiation of entities by the model [35].

When generating negative example triplets, different replacement strategies are set depending on the relationship type. For one-to-many relationships, a larger probability of replacing the head entity is used, for many-to-one relationships a larger probability of replacing the tail entity is used, for many-to-many relationships by considering the number of nodes of the head and tail entities in the relationship, the entity with the lower number is replaced.

In the triplets, N_{tph} denotes the average value of the number of tail entities corresponding to each head entity, N_{hpt} denotes the average value of the number of head entities corresponding to each tail entity, and the probability P is:

$$P = \frac{N_{tph}}{N_{tph} + N_{hpt}} \tag{5}$$

For a given triplet, replace the head entity with probability P and the tail entity with probability $1-P$ to generate a negative example triplet.

C. PROBABILITY DISTRIBUTION IN REAL SPACE

The closer the entities are to together, the more likely they belong to the same class, and the corresponding entity projections should be more similar. Similarly, the closer the entity class centers are to each other, the closer the corresponding projection vectors are. The similarity of entity class centers and their projection vectors are structed to obtain the similarity measure of them.

The class spacing is performed using the entity class center instead of the entire entity class, and the similarity of the entity class center is measured by the Euclidean distance, and its similarity $d(e_h^i, e_h^j)$ is [36]:

$$d(e_h^i, e_h^j) = \left\| e_h^i - e_h^j \right\|_2 \tag{6}$$

The Euclidean distance is converted into probability by the normal distribution function to represent similarity, and the probability $P(e_h^i \mid e_h^j)$ of choosing e_h^i conditionally on e_h^j, the entity class central similarity probability $P(e_h^i \mid e_h^j)$ is defined as:

$$P(e_h^i \mid e_h^j) = \frac{\exp(-d(e_h^i, e_h^j))}{\sum\limits_{n \neq i} \exp(-d(e_h^i, e_h^n))} \tag{7}$$

the entity class centers e_h^i corresponds to the projection vectors e_{ph}^i, and the probability $P(e_{ph}^i \mid e_{ph}^j)$ that e_{ph}^i is selected conditionally on e_{ph}^j, the similarity $P(e_{ph}^i \mid e_{ph}^j)$ of the entity class center projection vector is defined as:

$$P(e_h^i \mid e_h^j) = \frac{\exp(-d(e_h^i, e_h^j))}{\sum\limits_{n \neq i} \exp(-d(e_h^i, e_h^n))} \tag{8}$$

since the probabilities of the projection vectors do not have symmetry, the conditional probabilities of the above equation are symmetrized to obtain the entity class center similarity probability $P'(e_h^i|e_h^j)$ and the entity class center projection vector similarity $P'(e_{ph}^i|e_{ph}^j)$ defined respectively as:

$$P'(e_h^i|e_h^j) = \frac{p(e_h^i|e_h^j) + p(e_h^j|e_h^i)}{2k}, \quad P'(e_h^i|e_h^j) = P'(e_h^j|e_h^i) \tag{9}$$

$$P'(e_{ph}^i|e_{ph}^j) = \frac{\exp(-d(e_{ph}^i, e_{ph}^j))}{\sum\limits_{n \neq m} \exp(-d(e_{ph}^m, e_{ph}^n))},$$
$$P'(e_{ph}^i|e_{ph}^j) = P'(e_{ph}^j|e_{ph}^i) \tag{10}$$

D. EMBEDDED MODEL TRAINING

The model training iteration consists of two stages: triplet loss and Kullback-Leibler (KL) divergence loss. In each iteration, the triplet loss is firstly repeated twice, and the obtained entity vector representation is used as the input of KL divergence loss and training is continued once more to better collaborate and optimize the model by alternate learning.

The triplet loss function and KL divergence loss function are used as the objective functions of the samples for training. The purpose of the triplet loss function is to distinguish between positive example and negative example triplets, and the purpose of the KL divergence loss function is to measure the similarity between the entity class centers and their corresponding entity projections, and the objective function is:

$$\begin{aligned}
\mathcal{L} &= \mathcal{L}_{score} + \mathcal{L}_{KL} \\
&= \sum_{(h,r,t)\in S,(h',r,t')\in S'} \xi(f(h,r,t), f(h',r,t')) + KL(P||Q) \\
&= \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} \max(f(h',r,t') + \gamma - f(h,r,t), 0) \\
&\quad + \sum_i \sum_j P'(e_h^i|e_h^j) \log\left(\frac{P'(e_h^i|e_h^j)}{P'(e_{ph}^i|e_{ph}^j)}\right)
\end{aligned} \tag{11}$$

where $(h, r, t) \in S$ denotes the set of positive example triplets and $(h', r, t') \in S'$ denotes the set of negative example triplets, generated by the negative example triplet construction strategy, P denotes the probability distribution of entity class centres, Q denotes the probability distribution of entity class centre projections, and γ denotes the distance between correct triad scores and incorrect triad scores, the objective function uses stochastic gradient descent algorithm (SGD) to update the model parameters, each time Bernoulli sampling extracts part of the triplets, generates negative example triplets to join the triplet data set, normalizes the entities and relationships in the set in their respective vector space operations, and uses the normalized vector data to train the model.

V. RECOMMENDATION MODELS BASED ON KNOWLEDGE GRAPH AND GRAPH ATTENTION NETWORKS

This chapter proposes a CTransD-GAT recommendation model based on knowledge graphs and graph attention networks, introduces the overall framework of the model, and describes in detail the processes of user weight preference layers, feature propagation and embedding aggregation, rating prediction and model training.

A. OVERALL FRAMEWORK

Since the existing KG-based recommendation systems ignore the problem of different importance arising from different relationships of users to items, they cannot intuitively display the user preference weights. Therefore, we propose the CTransD-GAT recommendation model based on KG and graph attention networks, which combines the KG and graph

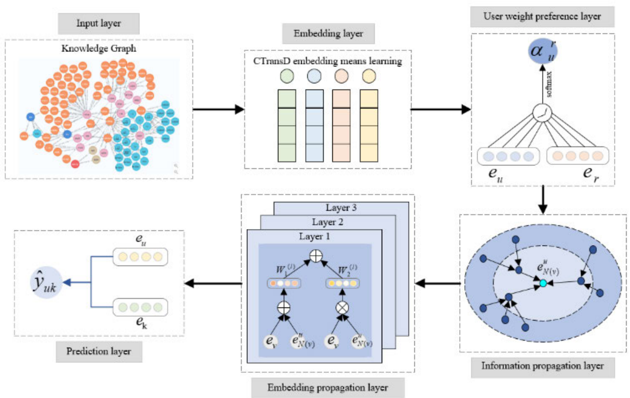

FIGURE 2. Framework diagram of CTransD-GAT model.

attention networks, allowing to mine the weight information of nodes in the KG, and then propagate and aggregate the information of nodes according to the weight coefficients to generate new node representations, so as to improve the recommendation accuracy, and the framework of CTransD-GAT model as illustrated Figure 2.

(1) The first input is the KG of users, items and relations.

(2) The third part of the knowledge representation learning CTransD model is employed to embed the nodes and relations into a low-dimensional dense vector space to form the initial embedding vectors of users, items and relations.

(3) Calculation of user weight coefficients is done by introducing a relational attention function to learn how much importance users attach to various relations of items and transforming the initial KG into a weighted graph to make it show the weight magnitude.

(4) Information propagation, where the feature information of an entity node is propagated to neighbouring nodes based on user weight coefficients.

(5) Embedding propagation aggregation, where the feature information of the entity node is embedded and aggregated with the propagated neighbouring nodes to generate a new node representation containing more information.

(6) Rating prediction and model training is the inner product operation between the user embedding vector and the aggregated item embedding vector to get the predicted rating, and the parameters in the CTransD-GAT model are trained using the Bayesian personalized ranking (BPR) learning algorithm.

B. USER WEIGHTED PREFERERENCE LAYER

In order to portray that the users generate different importance to different relationships, the weight of relations to users in the KG is calculated by the attention mechanism to judge the degree of influence of neighboring entities on the current entity. The inner product of the user's embedding vector and the relationship embedding vector is used to represent the importance of that kind of relationship to the user. The edges connected by the head and tail nodes in the KG are the weights, and the weight coefficients are expressed as follows:

$$s_u^r = e_u^T e_r \tag{12}$$

where e_u and e_r denote the embedding vectors of user u and relation r, respectively. The KG is an unweighted graph that cannot show the weight size, and the unweighted graph is transformed into a weighted graph through the user weight preference layer. To better assign the weights, we need to normalize the weight coefficients by the softmax function normalization, which is expressed as follows:

$$\alpha_u^r = soft\max(s_u^r) = \frac{\exp(e_u^T e_r)}{\sum\limits_{e\in N(\upsilon)} \exp(e_u^T e_r)} \tag{13}$$

where α_u^r is the normalized weight coefficient and $N(v)$ denotes the set of neighboring nodes of node v.

C. FEATURE PROPAGATION AND EMBEDDING AGGREGATION

Information propagation is the propagation of feature information of entity nodes to neighboring nodes based on weight coefficients along the connectivity between nodes, and information aggregation is the aggregation of embeddings of entity nodes with propagated neighboring nodes using convolutional operations and generates new node representation containing more information.

The neighbor nodes of all relationship types are propagated and the nodes are weighted and summed to obtain the feature vector of the entity's neighbors, denoted as follows:

$$e_{N(v)}^u = \sum_{a\in N(v)} \alpha_u^r e_a \tag{14}$$

In order to fuse more semantic association information of neighboring nodes, the model will explore the use of three different aggregation functions to aggregate the entity feature vector e_v with its neighboring feature vector $e_{N(v)}^u$ to obtain the final feature vector e_k. In this paper, three types of aggregation functions are established.

(1) Sum aggregation function, which adds the entity feature vectors and their neighbourhood feature vectors, and then performs a nonlinear transformation, is expressed as follows:

$$e_k = agg_{sum} = \sigma(W(e_v + e_{N(v)}^u) + b) \tag{15}$$

where $W \in R^{d\cdot d}$ and $b \in R^d$ are trainable weight matrices with deviations, σ is the ReLu activation function.

(2) The GraphSage aggregation function, which joins the entity feature vectors and their neighbourhood feature vectors, and then performs a nonlinear transformation, is represented as follows:

$$e_k = agg_{GraphSage} = \sigma(W(e_v || e_{N(v)}^u) + b) \tag{16}$$

where || denotes the vector join operation.

(3) The BI-Interaction aggregation function, summing the entity feature vectors and their neighbourhood feature vectors for nonlinear transformation, then dot product the entity

feature vectors and their neighbourhood feature vector elements for nonlinear transformation, and finally performs the summation operation, is represented as follows:

$$e_k = agg_{BI} = \sigma(W_1(e_v + e^u_{N(v)}) + b_1) + \sigma(W_2(e_v \odot e^u_{N(v)}) + b_2) \tag{17}$$

where $W_1, W_2 \in R^{d \cdot d}$ and $b_1, b_2 \in R^d$ are trainable weight matrices with biases, $\odot$ denoting the dot product.

D. SCORING PREDICTION AND MODEL TRAINING

After the aggregation operation in the previous step, the final item feature vector e_k is obtained. The final step of the model is to make an inner product of the user embedding vector and the aggregated item embedding vector as the probability value $\hat{y}_{uk}$ of the user clicking on the item, which is expressed as follows:

$$\hat{y}_{uk} = e_u^T e_k \tag{18}$$

where the prediction function $\hat{y}_{uk}$ is an inner product operation to describe the predicted relevance score between user u and item k. The higher the similarity between user and item, the higher the score.

The parameters in the CTransD-GAT model are trained using a Bayesian personalized ranking (BPR) learning algorithm, and the BPR algorithm calculates and ranks the relevance scores of each user for unknown items to personalize recommendations. The loss function during the training period is represented as follows:

$$loss = \sum_{(u,k^+,k^-)\in O} - \ln \sigma(\hat{y}_{uk^+} - \hat{y}_{uk^-}) + \lambda||\theta||_2^2 \tag{19}$$

where O denotes the sample training set, $O = \{(u, k^+, k^-)|(u, k^+) \in R^+, (u, k^-) \in R^-\}$, R^+ denotes the positive sample data set, R^- denotes the negative sample data set, σ is the nonlinear activation function Sigmoid, θ denotes all parameters in the model that can be trained, and $\lambda||\theta||_2^2$ denotes the L2 regularization term, which is used to prevent overfitting. The purpose of this loss function is to maximize the difference between the scores of positive and negative samples as much as possible.

VI. EXPERIMENT

The performance of the models CTransD and CTransD-GAT are evaluated by designing experiments and analyzing the experimental results. We first present the data set used for the experiments, the experimental setup, the evaluation metrics for the experimental evaluation, the baseline for comparison with other models, and the experimental results are finally given.

A. DATASET

In the experiments, to validate the performance of the model designed in this paper, public experimental datasets from three domains are used, which are from education, movies and books, respectively.

TABLE 2. Data set information.

Dataset		MoocCube	MovieLens-25M	Book-Crossing
User interaction	Users	194,195	162,541	17,860
	Items	706	62,423	14,910
	Interactions	8,002,841	25,000,095	1,039,781
Knowledge graphs	Entities	1,030,773	175,200	24,039
	Relationships	7	32	10
	Triples	7,573,244	504,875	19,793

Education dataset: MoocCube is an open source large-scale data warehouse serving MOOC-related research. The dataset contains more than 190,000 users, over 700 courses and nearly 8 million course rating data.

Movie dataset: MovieLens-25M is a stable benchmark dataset for recommender system testing, which contains more than 160,000 users, 60,000 movies, and nearly 25 million movie rating data.

Book dataset: Book-Crossing is a book rating dataset written by Cai-Nicolas Ziegler based on data from bookcrossing.com. The dataset contains more than 10,000 users, 10,000 books and 1.03 million book rating data.

Knowledge cleaning is performed on each dataset data to fill the missing values to reduce the impact of invalid data on modeling. The vertical domain KG mostly adopts top-down knowledge modeling, abstractly generalizes domain knowledge, gets the entity concept of domain KG, defines entity classes, object attributes, data attributes, value domains and constraints for each entity concept in the domain, and constructs the domain KG ontology library. We obtain data from the dataset to extract the required elements of the KG of entities, attributes and relationships, instantiate the ontology library, and form structured knowledge to deposit into the Neo4J database to complete the vertical domain KG construction, and the three datasets after completing the KG construction are illustrated in Table 2.

B. EVALUATION INDICATORS

Knowledge representation learning models assess the quality of the algorithm by common link prediction scenarios based on ranking, usually using the criteria MeanRank for the average ranking of the correct entity scoring function and Hits@10 for the odds of ranking the correct entity in the top 10 to assess the quality of the model.

The recommendation model of graph attention networks for TopN recommendation task uses Precision@K and Recall@K to measure the ability of the recommendation system to correctly predict users' preferences or not preferences of an item, assuming that $N_{TP}, N_{FP}, N_{FN}, N_{TN}$ denote that the system recommends to the user and the user preferences, the system recommends to the user but the user not preferences, the user preferences but the system does not recommend and the user not preferences and the system does not recommend, respectively. The definition of precision rate is expressed as follows:

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{20}$$

the definition of recall is expressed as follows:

$$Recall = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{21}$$

graph attention networks recommendation model for click-through prediction, AUC and F1 are used to measure the performance of the recommendation system. AUC is the area under the ROC curve, which indicates how well the recommendation system is able to distinguish items that user preferences from those that they do not like. The AUC definition is expressed as follows:

$$AUC = \int_0^1 f(ROC)dx \tag{22}$$

F1 is a comprehensive reflection of system performance by considering both accuracy and recall, and F1 is defined and expressed as follows:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{23}$$

C. BASELINE MODELS

In order to verify the effectiveness of the CTransD and CTransD-GAT models proposed in this paper, five baseline models with outstanding effects in the knowledge representation learning domain and five baseline models with outstanding effects in the recommendation model domain are compared, respectively.

The five models in the knowledge representation learning domain are TransE, TransH, TransR, CTransR, and TransD, respectively.

The five baseline models with outstanding effectiveness in the recommendation model domain are CKE, LibFM, RippleNet, NGCF, respectively, which are specifically represented as:

CKE: A model that fuses knowledge mapping and collaborative filtering, while employing TransR heterogeneous network embedding.

LibFM: A model based on feature matrix decomposition, which employs stacked gradient descent (SGD) and alternating least squares (ALS) for optimization.

RippleNet: a water wave network model that introduces preference propagation into a recommendation model that fuses KG.

NGCF: Neural graph collaborative filtering model, which embeds a bipartite graph coded representation of user items.

KGAT: An end-to-end approach to explicit modelling of higher-order connections in knowledge graph attention networks based on a knowledge graph attention networks recommendation model.

D. EXPERIMENTAL PARAMETER SETTING

For each dataset, 70% was randomly selected as the training set for model training, and the remaining 30% was used as the test set for testing data. The model was optimised using the Adam optimiser in Python 3.7, TensorFlow 2.10, and Numpy 1.19 environments. To ensure a fair comparison, for the baseline model, the parameter settings are first based on its original text, and the comparison model is optimized as much as possible on this basis. For the model CTransD, the parameter learning rate ε is set to take values in the range $\{0.01,0.001,0.0001\}$, the spacing γ is chosen among $\{0.25,0.5,1,2\}$, the embedding dimensions m and n of entities and relations are chosen among $\{20,50,80,100\}$, the size of single batch data B is chosen among $\{100,200,1000,1400\}$, and the number of clusters K is chosen among $\{20,50,100,200\}$. For the CTransD-GAT model, the model is trained in batch mode with a fixed batch size of 512 and a learning rate chosen from $\{0.01,0.001,0.0001\}$, and its aggregator uses the BI-Interaction aggregator by default. the number of neighbors and the number of hops for the nodes are 4 and 2, respectively. 10 experiments are done for each combination configuration, and the maximum number of iterations in the experiments is 500, and then the experimental results are averaged to determine the optimal parameter configuration.

TABLE 3. Experimental results of model about CtansD.

Dataset	MoocCube		MovieLens-25M		Book-Crossing	
Evaluation index	Mean Rank	Hits@10	Mean Rank	Hits@10	Mean Rank	Hits@10
TransE	263	80.3	168	67.7	245	61.2
TransH	369	86	155	73.1	238	69.4
TransR	225	86.9	140	74.5	244	72.1
CTransR	221	88.2	142	74.1	238	73.5
TransD	218	91.7	133	78.2	231	77.8
CTransD	212	92.5	128	81.5	228	79.6

TABLE 4. Click-through rate prediction experiment results.

Model	MoocCube		MovieLens-25M		Book-Crossing	
	AUC	F1	AUC	F1	AUC	F1
CKE	0.922	0.869	0.895	0.812	0.671	0.594
LibFM	0.913	0.851	0.824	0.793	0.682	0.635
RippleNet	0.946	0.904	0.936	0.901	0.693	0.655
NGCF	0.962	0.921	0.947	0.915	0.704	0.663
KGAT	0.973	0.920	0.955	0.919	0.716	0.669
CTransD-GAT	0.976	0.932	0.963	0.922	0.739	0.687

E. EXPERIMENTAL RESULTS

The results of the experimental prediction of the domain knowledge graph entity links in each of the three datasets are illustrated in Table 3. The results show that the model CTransD has an improvement in both metrics relative to the original model TransD, and the model CTransD represents learning better ability relative to the other models.

Top-K recommendation and click-through prediction experiments are performed on the baseline and CTransD-GAT models, and the accuracy and recall of each model are derived on each of the three datasets as illustrated in Figure 3, as well as the AUC and F1 of each model as illustrated in Table 4.

The AUC and F1 values of CTransD-GAT model are higher than other models in all three datasets, and the accuracy rate of all models in Top-K recommendation shows a decreasing trend and the recall rate shows an increasing trend as the

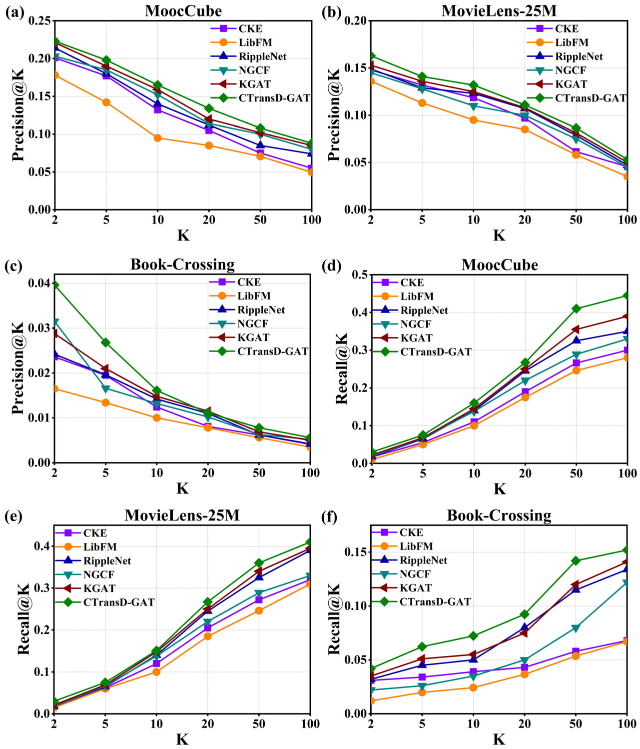

FIGURE 3. Accuracy rate of the model are about (a) MoocCube, (b) MovieLens-25M, (c) Book-Crossing, respectively. Recall Rate of the model about (d) MoocCube, (e) MovieLens-25M, (f) Book-Crossing, respectively.

TABLE 5. Experimental results of different aggregators.

Model	MoocCube		MovieLens-25M		Book-Crossing	
	AUC	F1	AUC	F1	AUC	F1
Sum	0.975	0.936	0.964	0.924	0.739	0.691
GraphSage	0.971	0.921	0.957	0.917	0.737	0.677
BI-Interaction	0.982	0.939	0.968	0.925	0.741	0.693

K value increases. In the three datasets, the LibFM model has a slightly lower accuracy rate compared with the other models incorporating KG, indicating that the introduction of KG as auxiliary information in the recommendations effectively extracts the feature vectors of items and improves the accuracy rate of the recommendations. All the models based on graph neural networks performed better, which also proved that the iterative aggregation method of graph neural networks helps to mine the higher-order relationships in the knowledge graph. The CTransD-GAT model performs significantly better than other models that only fuse KG, indicating that the incorporation of graph attention networks effectively propagates and aggregates the initial embedding vectors with weights, which improves the recommendation accuracy and indicates the effectiveness of the model, and also confirms that the recommendation model based on KG and graph neural networks is better than the recommendation model based on KG embedding.

The effects of different aggregators on the model performance are illustrated in Table 5, and the effects of different aggregation depths on the model performance are illustrated in Table 6.

TABLE 6. Experimental results with different polymerization depths.

Jump Count	MoocCube		MovieLens-25M		Book-Crossing	
	AUC	F1	AUC	F1	AUC	F1
1	0.968	0.917	0.951	0.917	0.739	0.689
2	0.976	0.932	0.963	0.922	0.735	0.674
3	0.954	0.915	0.944	0.903	0.716	0.658

A large number of ablation experiments are conducted on the model, and different aggregators had different effects on the model performance. Sum aggregator, GraphSage aggregator and BI-Interaction aggregator are compared, and BI-Interaction aggregator had better results in each index. The impact of different aggregation depths on model performance is produced. The best performance in MoocCube and MovieLens-25M when the number of hops is 2, and the better performance in Book-Crossings when the number of hops is 1. Therefore, the model performance is better when the number of hops is 1 or 2.

VII. CONCLUSION

In this paper, we propose an improved knowledge representation learning model CTransD and a recommendation model CTransD-GAT based on KG and graph attention networks. The CTransD model reduces the number of entity projections and makes the entity representations related to each other, which reduces the computational pressure and makes it better to be applied to large KG. The CTransD-GAT model captures the information of different users on different relations of items through graph attention networks captures the weight information of different users on different relationships of items, and then propagates and aggregates the information of nodes according to the weight coefficients, which improves the recommendation performance of the model. A large number of comparison and ablation experiments are conducted on three real datasets, and the experimental results show the rationality and effectiveness of the model proposed in this paper, and it outperforms the existing excellent models.

REFERENCES

[1] L. Wu, "Collaborative filtering recommendation algorithm for MOOC resources based on deep learning," *Complexity*, vol. 2021, pp. 1–11, Apr. 2021.

[2] X. Wu, Q. Chen, H. Liu, and C. He, "Collaborative filtering recommendation algorithm based on knowledge graph," *Math. Problems Eng.*, vol. 2019, pp. 1–13, Jul. 2019.

[3] X. Liu, "An improved clustering-based collaborative filtering recommendation algorithm," *Cluster Comput.*, vol. 20, no. 2, pp. 1281–1288, 2017.

[4] A. Hogan, "Knowledge graphs," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–37, 2021.

[5] M. He, X. Du, and B. Wang, "Representation learning of knowledge graphs via fine-grained relation description combinations," *IEEE Access*, vol. 7, pp. 26466–26473, 2019.

[6] J. Dong, J. Wang, and S. Chen, "Knowledge graph construction based on knowledge enhanced word embedding model in manufacturing domain," *J. Intell. Fuzzy Syst.*, vol. 41, no. 2, pp. 3603–3613, Sep. 2021.

[7] Z. Pei, J. Zhang, W. Xiong, L. Li, and Z. Luo, "A general framework for Chinese domain knowledge graph question answering based on TransE," *J. Phys., Conf. Ser.*, vol. 1693, pp. 12136–12144, Dec. 2020.

[8] L. Li, P. Wang, Y. Wang, S. Wang, J. Yan, J. Jiang, B. Tang, C. Wang, and Y. Liu, "A method to learn embedding of a probabilistic medical knowledge graph: Algorithm development," *JMIR Med. Inform.*, vol. 8, pp. 17645–17657, May 2020.

[9] M. M. Alam, M. R. A. H. Rony, M. Nayyeri, K. Mohiuddin, M. S. T. M. Akter, S. Vahdati, and J. Lehmann, "Language model guided knowledge graph embeddings," *IEEE Access*, vol. 10, pp. 76008–76020, 2022.

[10] Y. Zhu, Q. Lin, H. Lu, K. Shi, P. Qiu, and Z. Niu, "Recommending scientific paper via heterogeneous knowledge embedding based attentive recurrent neural networks," *Knowl.-Based Syst.*, vol. 215, pp. 106744–106755, Jan. 2021.

[11] M. Abulaish and M. Fazil, "Modeling topic evolution in Twitter: An embedding-based approach," *IEEE Access*, vol. 6, pp. 64847–64857, 2018.

[12] L. Tang, Y. Zhao, Z. Duan, and J. Chen, "Efficient similarity search for travel behavior," *IEEE Access*, vol. 6, pp. 68760–68772, 2018.

[13] Y. Wang, L. Dong, Y. Li, and H. Zhang, "Multitask feature learning approach for knowledge graph enhanced recommendations with RippleNet," *PLoS ONE*, vol. 16, no. 5, May 2021, Art. no. e0251162.

[14] S. Marton, S. Lüdtke, and C. Bartelt, "Explanations for neural networks by neural networks," *Appl. Sci.*, vol. 12, no. 3, p. 980, Jan. 2022.

[15] X. Kang, H. Yao, Q. Li, X. Li, C. Liu, and L. Dong, "TDN: An integrated representation learning model of knowledge graphs," *IEEE Access*, vol. 7, pp. 55199–55205, 2019.

[16] P. Cui, S. Liu, and W. Zhu, "General knowledge embedded image representation learning," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 198–207, Jan. 2018.

[17] S. Ji, N. Satish, S. Li, and P. K. Dubey, "Parallelizing Word2Vec in shared and distributed memory," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 2090–2100, Sep. 2019.

[18] B. Antoine, U. Nicolas, and G.-D. Alberto, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2013, pp. 1–15.

[19] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. 28th AAAI Conf. Artif. Intell.*, Québec City, QC, Canada, Jul. 2014, pp. 1–8.

[20] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin, TX, USA, Jan. 2015, pp. 1–7.

[21] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 7th Int. Joint Conf. Natural Lang. Process.*, Beijing, China, Jul. 2015, pp. 687–696.

[22] X. Guan, C. T. Li, and Y. Guan, "Matrix factorization with rating completion: An enhanced SVD model for collaborative filtering recommender systems," *IEEE Access*, vol. 5, pp. 27668–27678, 2017.

[23] Z. Huang, C. Yu, J. Ni, H. Liu, C. Zeng, and Y. Tang, "An efficient hybrid recommendation model with deep neural networks," *IEEE Access*, vol. 7, pp. 137900–137912, 2019.

[24] F. Zhao, F. Yan, H. Jin, T. Yang, and C. Yu, "Personalized mobile searching approach based on combining content-based filtering and collaborative filtering," *IEEE Syst. J.*, vol. 11, no. 1, pp. 324–332, Mar. 2017.

[25] M. Bahrkazemi and M. Mohammadi, "A strategy to estimate the optimal low-rank in incremental SVD-based algorithms for recommender systems," *Intell. Data Anal.*, vol. 26, no. 2, pp. 447–467, Mar. 2022.

[26] M. Belford, B. M. Namee, and D. Greene, "Stability of topic modeling via matrix factorization," *Expert Syst. Appl.*, vol. 91, pp. 159–169, Jan. 2018.

[27] F. Lu, P. Cong, and X. Huang, "Utilizing textual information in knowledge graph embedding: A survey of methods and applications," *IEEE Access*, vol. 8, pp. 92072–92099, 2020.

[28] Y. Lan, S. He, K. Liu, X. Zeng, S. Liu, and J. Zhao, "Path-based knowledge reasoning with textual semantic information for medical knowledge graph completion," *BMC Med. Informat. Decis. Making*, vol. 21, no. S9, pp. 335–345, Nov. 2021.

[29] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 353–362.

[30] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-k similarity search in heterogeneous information networks," in *Proc. 37th Int. Conf. Very Large Data Bases*, Seattle, WA, USA, 2011, pp. 1–12.

[31] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 417–426.

[32] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Anchorage, AK, USA, Aug. 2019, pp. 950–958.

[33] S. Wu, B. Li, D. Xie, C. Teng, and D. Ji, "Neural transition model for aspect-based sentiment triplet extraction with triplet memory," *Neurocomputing*, vol. 463, pp. 45–58, Nov. 2021.

[34] M.-S. Yang and K. P. Sinaga, "A feature-reduction multi-view k-means clustering algorithm," *IEEE Access*, vol. 7, pp. 114472–114486, 2019.

[35] R. Fontana and P. Semeraro, "Exchangeable Bernoulli distributions: High dimensional simulation, estimation, and testing," *J. Stat. Planning Inference*, vol. 225, pp. 52–70, Jul. 2023.

[36] R. B. Bapat and H. Kurata, "On Cartesian product of Euclidean distance matrices," *Linear Algebra Appl.*, vol. 562, pp. 135–153, Feb. 2019.

QING HE was born in Xinxiang, Henan, China, in 1997. She is currently pursuing the master's degree in electronic information with Heilongjiang University, Harbin, China.

Her research interests include knowledge representation learning, knowledge graph, and recommender systems.

SONGYAN LIU was born in Harbin, Heilongjiang, China, in 1969. He received the M.S. and Ph.D. degrees in computer organization and system architecture from the Harbin Institute of Technology, Harbin, in 2000.

He is currently an Associate Professor with the School of Electronic Engineering, Heilongjiang University, Harbin. His research interests include recommender systems, data mining, and network security.

YAO LIU was born in Suqian, Jiangsu, China, in 1999. She is currently pursuing the master's degree in information and communication with Heilongjiang University, Harbin, China.

Her research interests include knowledge representation learning and recommender systems.