Hindawi
Complexity
Volume 2020, Article ID 3857871, 10 pages
https://doi.org/10.1155/2020/3857871

Research Article

A Tri-Attention Neural Network Model-Based Recommendation

Nanxin Wang,[1] Libin Yang ,[1] Yu Zheng,[2] Xiaoyan Cai,[3] Xin Mei,[1] and Hang Dai[1]

[1]*School of Cyberspace Security, Northwestern Polytechnical University, Xi'an 7100072, China*
[2]*Faculty of Information Technology, Monash University, Wellington Road, Clayton, VIC 3800, Australia*
[3]*School of Automation, Northwestern Polytechnical University, Xi'an 710072, China*

Correspondence should be addressed to Libin Yang; libiny@nwpu.edu.cn

Received 16 June 2020; Revised 13 August 2020; Accepted 22 September 2020; Published 9 October 2020

Academic Editor: Huizhi Liang

Heterogeneous information network (HIN), which contains various types of nodes and links, has been applied in recommender systems. Although HIN-based recommendation approaches perform better than the traditional recommendation approaches, they still have the following problems: for example, meta-paths are manually selected, not automatically; meta-path representations are rarely explicitly learned; and the global and local information of each node in HIN has not been simultaneously explored. To solve the above deficiencies, we propose a tri-attention neural network (TANN) model for recommendation task. The proposed TANN model applies the stud genetic algorithm to automatically select meta-paths at first. Then, it learns global and local representations of each node, as well as the representations of meta-paths existing in HIN. After that, a tri-attention mechanism is proposed to enhance the mutual influence among users, items, and their related meta-paths. Finally, the encoded interaction information among the user, the item, and their related meta-paths, which contain more semantic information can be used for recommendation task. Extensive experiments on the Douban Movie, MovieLens, and Yelp datasets have demonstrated the outstanding performance of the proposed approach.

1. Introduction

With the increasing amount of data on the Internet, users find it difficult to obtain useful information. In recent years, recommender systems which can only retain relevant information have received increasing attention [1–3]. Researchers proposed many methods to solve the recommendation task, which can be classified into four classes: neighborhood-based methods [4–7], model-based methods [8–11], graph-based methods [12–16], and deep neural network based methods [17–21]. Neighborhood-based methods contain user-based collaborative filtering and item-based collaborative filtering, this kind of methods utilize neighbor information of users to make prediction [22]. Model-based methods first construct a descriptive model using the users' preferences, and the recommendations are generated based on the model [23]. Graph-based methods not only consider the neighborhood information of each node but also consider the network structure [16, 20]. Inspired by the great success of deep neural networks in computer vision and natural language process, recent researchers have exploited deep neural networks in recommendation task.

Besides, since multiple types of auxiliary information become available, many methods propose to use this information to improve the performance of recommendation [24, 25]. As auxiliary information has heterogeneity and complexity, it is challenging to leverage this information in recommender systems. Heterogeneous information network (HIN), containing various kinds of nodes connected by multiple types of relations, that can model rich auxiliary data has been applied in recommender systems [26, 27]. Although the existing HIN-based recommendation approaches enhanced the performance than the traditional recommendation approaches, they still have the following problems: first, the meta-paths are manually selected, not automatically; second, meta-path representations are rarely explicitly learned; third, the global and local information of each node in HIN has not been simultaneously explored.

To solve the above deficiencies, we develop a tri-attention neural network (TANN) model for recommendation task. The proposed TANN model applies the stud genetic algorithm to automatically select meta-paths at first. Then, it learns global and local representations of each node, as well as the representations of meta-paths existing in the HIN. After that, a tri-attention mechanism is proposed to enhance the mutual influence among users, items, and their related meta-paths. Finally, the encoded interaction information among the user, the item, and their related meta-paths, which contain much semantic information, can be used for the recommendation task. To summarization, our major contributions are

(1) Meta-paths are automatically selected via the stud genetic algorithm.

(2) A novel tri-attention neural network (TANN) model is developed to enhance the influence among the user, the item, and their related meta-paths in a mutually reinforced way.

(3) A TANN model-based recommendation approach is developed, and the extensive experiments on the Douban Movie, MovieLens, and Yelp datasets demonstrate the outstanding performance of the approach.

We organize the rest of this paper as follows. Section 2 reviews related work. Section 3 presents the preliminaries and notations in the paper. Section 4 illustrates the proposed tri-attention neural network (TANN) model-based recommendation approach. Section 5 illustrates the experimental results and Section 6 concludes the paper.

2. Related Work

Accurately finding useful information in many e-resources becomes more and more difficult for users, due to the development of information technology and the increasing content of information. However, a recommendation system can overcome this obstacle.

Alqadah et al. [28] utilized each user's local biclustering neighborhood and developed a collaborative filtering method. Yao et al. [29] applied a clustering analysis and latent factor model to enhance the neighborhood-based recommendation's performance. Neighborhood-based systems use the stored ratings to make a recommendation, whereas model-based approaches learn a predictive model by using the ratings. Cremonesi et al. [30] presented a PureSVD-based matrix factorization method, which uses the user-item rating matrix's most principle singular vectors to describe users and items. Pan et al. proposed a consensus factorization based framework for coclustering networked data [31]. Sindhwani et al. [32] developed a weighted nonnegative matrix factorization method. Xiong et al. propose an information propagation-based social recommendation method (SoInp) and model the implicit user influence from the perspective of information propagation [33]. Hofmann [34] utilized PLSA (Probabilistic Latent Semantic Analysis) to solve collaborative filtering and showed that the PLSA is equivalent to nonnegative matrix factorization.

Most graph-based recommendation approaches are based on a random walk [35]. Christoffel et al. [12] proposed a graph random walk based recommendation algorithm. Kang et al. [16] presented a graph-based top-n recommendation model, which not only considers the neighborhood information encoding by user graph and item graph, but also takes into account the data's hidden structure.

Since deep learning techniques have been successfully applied in speech recognition and computer vision, some researchers began to utilize deep learning techniques in recommendation task and found that deep learning based recommendation approaches achieve better results than the conventional recommendation approaches. Oord et al. [36] applied deep convolution neural networks to generate songs' latent factors. Wang and Yang [37] combined probabilistic graphical models and deep belief networks to simultaneously learn audio content's features and make personalized recommendations. Xue et al. [38] presented a deep matrix factorization approach to solve the top-n recommendation task. Kim et al. [39] integrated convolution neural network into probabilistic matrix factorization and proposed a context-aware hybrid model for the recommender system.

In recent years, researchers adopted the heterogeneous information network (HIN) which characterizes rich auxiliary data in recommender systems. Pham et al. [40] modeled the rich information based on the constructed heterogeneous graph to solve the recommendation task. Chen et al. [41] developed a heterogeneous information network based projected metric embedding approach for link prediction. Yu et al. [42] presented a recommendation model based on a constructed attribute-rich HIN. Jiang et al. [43] modeled the user preferences using a generalized random walk with restart model and developed a heterogeneous information network based personalized recommendation method. Hu et al. [44] incorporated meta-path based context and proposed a co-attention mechanism based deep neural network to solve the recommendation task.

Although HIN-based deep learning approaches have achieved good performance in recommendation task, they usually select meta-paths manually, ignoring how to automatically select meta-paths. Moreover, the existing approaches seldom consider the interaction between node information and meta-path information. Our work applies the stud genetic algorithm to automatically select meta-paths, proposes a tri-attention mechanism that considers interactions among user-item-meta-path triplets, and develops a recommendation approach to further enhance the recommendation performance.

3. Preliminaries and Notations

We use the definitions of heterogeneous information network (HIN) and meta-path in [45].

Definition 1. A heterogeneous information network is an information network, which contains various kinds of objects and various kinds of links. It is defined as G=(V, A, E, R, W), where V is the set of different types of vertices, A is the object type set, E is the union of different types of links, R denotes the link type set, and W is the union of the weight on each link. An edge $e \in E$ is defined as$e_{ijr} = (v_i, r, v_j)v_i, v_j \in V,\ r \in R$.

Definition 2. A meta-path P is defined as a path in the form of $A_1 \longrightarrow^{R_1} A_2 \longrightarrow^{R_2} \ldots \longrightarrow^{R_l} A_{l+1}$ (abbreviated as $A_1, A_2, , \ldots, A_{l+1}$), which describes a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between A_1andA_{l+1}, where $\circ$ denotes the composition operator on relations.

The meta-path P's length is the number of relations contained in P. Taking the user-movie network as an example, we can use a 4-length meta-path to describe the user-movie relation such as user$\xrightarrow{\text{have seen}}$movie$\xrightarrow{\text{seen by}}$user $\xrightarrow{\text{have seen}}$ movie, or short as UMUM.

Definition 3. A path instance is a sequence of entity nodes; it is an explicit path in a meta-path, that is, $p \in P$.

4. Tri-Attention Neural Network (TANN) Model-Based Recommendation

In this section, we present the proposed tri-attention neural network (TANN) model at first and then illustrate the TANN model-based recommendation approach.

4.1. TANN Model. The whole architecture of the proposed TANN is presented in Figure 1.

4.1.1. Embeddings for Users and Items. In order to make the users' and items' representations more meaningful, we propose global representation to represent coarse-grained features of users and items, and develop local representation to represent fine-grained features of users and items; then, we integrate global information and local information of each node in HIN.

(1) Global Representations of Users and Items. Following [46], we use a lookup layer to map the users' and items' one-hot representations to low-dimensional dense vectors. Given a user-item pair <**u**, **i**>, let $\mathbf{l}_u \in \mathbb{R}^{||\times 1}$and $\mathbf{z}_i \in \mathbb{R}^{|\mathrm{I}|\times 1}$denote their one-hot representations. $\mathbf{L} \in \mathbb{R}^{||d}$ and $\mathbf{Z} \in \mathbb{R}^{|\mathrm{I}|\times d}$ represent the lookup layer's corresponding parameter matrices, which preserve users' and items' information; the user embedding's and item embedding's number of dimension is denoted as d; and ||and $|\mathscr{I}|$ denote the users' number and items' number, respectively. We apply HIN2VEC algorithm [47] to obtain the matrix **L** and **Z**. The user u's and item i's global representations are represented as

$$\mathbf{x}_u = \mathbf{L}^{\mathrm{T}} \cdot \mathbf{l}_u, \tag{1}$$

$$\mathbf{y}_i = \mathbf{Z}^{\mathrm{T}} \cdot \mathbf{z}_i. \tag{2}$$

(2) Local Representations of Users and Items. As each user can be represented as a sequence of item and each item can be represented as a sequence of user, we learn local representations of users (items) according to the corresponding item (user) sequence. Here, we use $\mathbf{S}_{n(u)} \in \mathbb{R}^{|\ell_u|\times|\mathrm{I}|}$ and $\mathbf{S}_{n(i)} \in \mathbb{R}^{|\ell_i|\times||}$ to represent the sequence matrix of the user u's and the item i's neighbors, respectively. For each neighbor node in the sequence, we use one-hot representation to represent it; $|\ell_u|$ and $|\ell_i|$ are the number of u's neighbors and the number of item i's neighbors in HIN; and $n(u)$ and $n(i)$ denote the neighbor set of user u and item i, respectively. Then, we apply a lookup layer to obtain the low-dimensional vector of each node in the item (user) sequence of the user (item). After that, the local representation of the user (item) is obtained based on a neighbor attention mechanism, which can be described as follows:

$$\mathbf{x}_u' = \left(\gamma_{n(u)} \odot \mathbf{b}_{n(u)}\right)^{\mathrm{T}} \cdot 1, \tag{3}$$

$$\mathbf{y}_i' = \left(\boldsymbol{\gamma}_{n(i)} \odot \mathbf{g}_{n(i)}\right)^{\mathrm{T}} \cdot 1, \tag{4}$$

where

$$\mathbf{b}_{n(u)} = \mathbf{S}_{n(u)} \cdot \mathbf{Z}, \tag{5}$$

$$\mathbf{g}_{n(i)} = \mathbf{S}_{n(i)} \cdot \mathbf{L}, \tag{6}$$

$$\boldsymbol{\gamma}_{n(u)} = \mathrm{softmax}\left(\mathbf{b}_{n(u)}\right), \tag{7}$$

$$\boldsymbol{\gamma}_{n(i)} = \mathrm{softmax}\left(\mathbf{g}_{n(i)}\right). \tag{8}$$

We concatenate the global representation vector and local representation vector of user u and item i and feed the concatenated vectors into MLP component to get the final representation, that is,

$$\widetilde{x}_u = \mathrm{MLP}\left(\mathbf{x}_u^{\mathrm{T}} \oplus \mathbf{x}_u^{'\mathrm{T}}\right), \tag{9}$$

$$\widetilde{y}_i = \mathrm{MLP}\left(\mathbf{y}_i^{\mathrm{T}} \oplus \mathbf{y}_i^{'\mathrm{T}}\right). \tag{10}$$

4.1.2. Meta-Path Embedding

(1) Meta-Path Selection Items. Assuming there exist M meta-paths in the heterogeneous information network G, we construct a phenotype matrix **H** (the size of the matrix is $C_M^X \times X\,(X \leq M)$), representing all possible combinations of Xmeta-paths, where each row represents a meta-path. Then, we apply the stud genetic algorithm (SGA) [48] to automatically select optimal X meta-paths.

(2) Meta-Path Instance Selection. Traditional HIN embedding models mainly use simple random walk strategies to obtain path instances. However, the path instances obtained by such strategies are of low value and cannot be directly applied to the recommendation system. Therefore, we propose a weighted selecting strategy with priority. In each step, the walker considers that the next step should walk to a

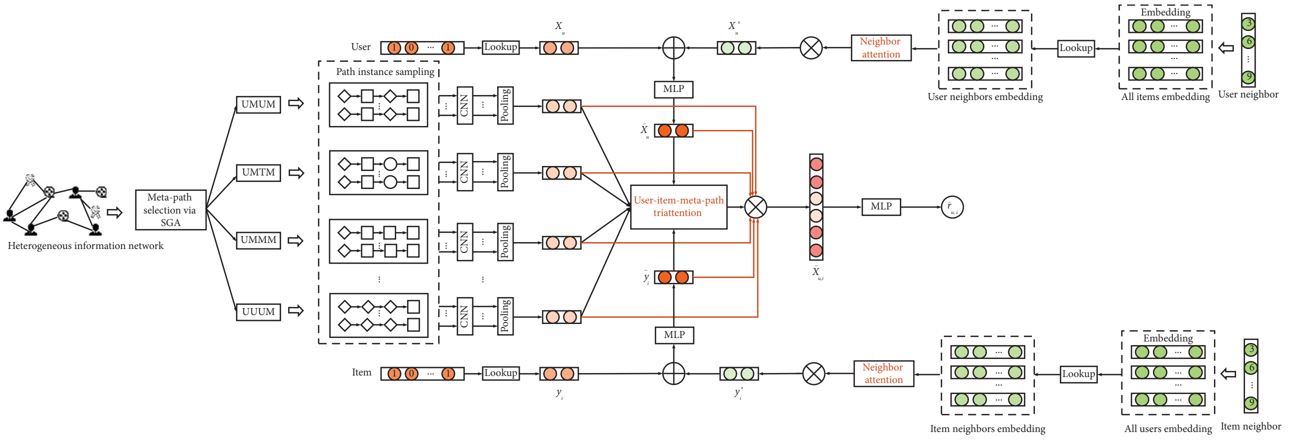


Figure 1: The whole architecture of the TANN model.

higher-priority neighbor, and using such walking strategy, a path instance which contains more semantic information can be obtained for recommendation task. Then, how to define the priority of each node in a sequence is a key problem.

Inspired by He et al. [46] and Hinton and Salakhutdinov [49], we use a similar pretraining technique to measure each candidate node's priority. The basic idea is to take the score between different nodes in the heterogeneous network as the weight allocation standard. For example, we define the score ranges from 1 to 5 in film evaluation; if the score of user u for movie i is 5, then we deem the weight value of the link between user u and movie i is the highest. For each node in HIN, it has a weight value for each node of its neighbors, and the similarity score between the node and the corresponding neighbor node can also be obtained. We measure the priority of each neighbor node by the product value between the weight value and the corresponding similarity score. Such score can reflect the correlation's degree between the two nodes. We use the above priority score strategy to construct each meta-path's instances.

Finally, for each meta-path, we obtain a different number of meta-path instances with a given length of L and then calculate scores of these meta-path instances as follows: for a path instance, we sum the product of weight and cosine similarity between adjacent nodes and then obtain that the sum value is divided by L as the score of the corresponding meta-path instance. So, we can obtain meta-path instances' scores of each meta-path and select the k path instances with high scores as the selected meta-path instance.

(3) Meta-Path Instance Embedding. Since a meta-path is a sequence of entity nodes, we apply the convolution neural network (CNN) to map a meta-path into a low-dimensional vector. For a meta-path P, we use $\mathbf{X}^P \in \mathbb{R}^{L\times d}$ to represent the path embedding matrix, where p is a path instance, L represents the path instance's length, and d is the nodes' embedding dimension. The path instance p's embedding is computed as follows:

$$\mathbf{h}_p = \mathrm{CNN}\left(\mathbf{X}^p; \Theta\right), \tag{11}$$

where Θ denotes CNN's related parameters.

(4) Meta-Path Embedding. As each meta-path contains many path instances, we first apply the proposed meta-path instance selection strategy to obtain each meta-path's top k path instances. Then, we use the maximum pooling operation to get important dimensional features from the selected path instances. Let $\left\{\mathbf{h}_p\right\}_{p=1}^{k}$ denote k selected path instances' embedding. The embedding of the meta-path P is computed as

$$c_P = \max - \mathrm{pooling}\left(\left\{\mathbf{h}_p\right\}_{p=1}^{k}\right). \tag{12}$$

(5) Tri-Attention Mechanism Based Interaction Embedding. As meta-paths contain rich semantic information, different users through different meta-paths show different preferences; even when the same user interacts with different items through the same meta-path, semantic information contained in the meta-path is also different. In order to better represent the semantic information existing among users, items, and meta-paths, we develop a tri-attention mechanism to assign different weights to different triplets of user-item-meta-path.

Given the user embedding $\mathbf{x}_u$, item embedding $\mathbf{y}_i$, meta-path embedding $c_P$which exists between the user u and item i, we use two full-connection layers to get the triattentive score as

$$\boldsymbol{\beta}_{1P} = f\left(\mathbf{W}_1^u \tilde{x}_u + \mathbf{W}_1^i \tilde{y}_i + \mathbf{W}_1^P \mathbf{c}_P + \mathbf{b}_1\right), \tag{13}$$

$$\beta_{2P} = f\left(\mathbf{w}_2 \boldsymbol{\beta}_1 + b_2\right), \tag{14}$$

where $\mathbf{W}_1^*$ is the first layer's weight matrix, $\mathbf{b}_1$ is the first layer's bias vector, $\mathbf{w}_2$ is the second layer's weight vector, and $\mathbf{b}_2$ is the second layer's bias. $f(\cdot)$ is set to the ReLU function.

We use softmax function to obtain the final interaction weights; that is,

$$\beta_P = \frac{\exp\left(\beta_{2P}\right)}{\sum_{\rho' \in P_{u\longrightarrow i}} \exp\left(\beta_{2P}\right)}, \tag{15}$$

where $P_{u\longrightarrow i}$ is the set of meta-paths existing between the user u and the item i. The interaction embedding among the user u, the item i, and their related meta-path P can be represented as

$$\tilde{x}_{u,i} = \sum_{p\in P_{u\longrightarrow i}} \beta_P \cdot \left(\tilde{x}_u \oplus \tilde{y}_i \oplus \mathbf{c}_p\right), \tag{16}$$

where $\oplus$ represents the vector concatenation operation.

4.2. TANN Model-Based Recommendation Approach. Once we obtain the interaction embedding, we apply an MLP component to model complicated interactions:

$$\tilde{r}_{u,i} = \mathrm{MLP}\left(\tilde{x}_{u,i}\right), \tag{17}$$

in which the MLP contains two hidden layers with ReLU nonlinear activation functions and an output layer with sigmoid functions. $\tilde{r}_{u,i}$ is interpreted as the relevance score between the user u and the item i, and $\tilde{r}_{u,i}$ is used to generate a recommendation list for the user u.

Defining an appropriate objective function is a critical step for model optimization; following [14, 17], we use negative sampling to learn the model's parameters:

$$\ell_{u,i} = -\log \tilde{r}_{u,i} - \sum_{j=1}^{c} \log\left(1 - \tilde{r}_{u,j}\right), \tag{18}$$

in which the first term is the observed interactions, and the second term indicates the negative feedback drawn from the noise distribution which is set to the uniform distribution (it can be set to other biased distributions), and c is the number of negative item sampling.

5. Experiment and Evaluation

5.1. Datasets. We use three datasets, that is, Douban Movie dataset (https://github.com/librahu), MovieLens movie dataset (https://grouplens.org/datasets/movielens), and Yelp business dataset (https://www.yelp.com/dataset). As a rating indicates whether a user has rated an item, we deem the rating as an interaction record [47, 50]. Table 1 lists the detailed description of the datasets. Each dataset's first row shows the number of users, items, and their interactions, and the other rows list the other relations' statistics. The manually selected meta-paths and SGA selected meta-paths of each dataset are listed in Table 2. Since long meta-paths may import noisy semantics [51], we set the length of meta-path to 4 and set the number of each meta-path's selected path instances to 5.

5.2. Evaluation Methods. In order to evaluate the recommendation's performance, each dataset's user implicit feedback records are divided into a training set and test set, according to a certain proportion. For example, we use 90% feedback records to predict the remaining 10% feedback records. As it is a waste of time ranking each user's items, especially for large datasets, for each user in the dataset, we randomly select 200 negative samples having no interaction records with the user at first. After that, we obtain each user's recommendation list by ranking the positive items and negative items of the list. We use Pre@K, Recall@K, and NDCG@K to evaluate the experimental results.

Table 1: Statistics of the three datasets.

Datasets	Relations (A-B)	#A	#B	#A-B
MovieLens	User-movie	943	1682	100,000
	User-user	943	943	47,150
	Movie-movie	1682	1682	82,798
	Movie-type	1682	18	2891
Douban movie	User-movie	1830	12337	697311
	User-user	1830	1830	9155
	Movie-movie	12337	12337	61690
	Movie-type	12337	38	12337
Yelp	User-business	300	450	15109
	User-user	300	300	1505
	Business-business	450	450	2255
	Business-category	450	47	450

When we apply the SGA algorithm to select the optimal 4 meta-paths, we follow the principle that "the smaller the objective function value is, the larger the fitness value is"; that is, we take negative evaluation scores. We implement the TANN model using TensorFlow with Keras (https://keras.io/). The batch size is set to 256, the regularization parameter is set to 0.0001, the learning rate is set to 0.001, and the dimension of user embedding and item embedding is set to 64. We apply Adaptive Moment Estimation (Adam) [52] to optimize the model.

5.3. Performance of TANN-Based Recommendation. To illustrate the benefits of applying the SGA algorithm, weighted random walk strategy, and integrating HIN's global and local information of TANN model, we compare TANN against the following:

(1) TANN w/o SGA, which applies manually selected meta-paths; we use the meta-paths in the second column of Table 2;

(2) TANN w/o WRWS, which applies a random walk strategy (RWS) instead of weighted random walk strategy (WRWS) to select meta-path instances;

(3) TANN w/o local, which considers global information of HIN only, ignoring local information, that is, it uses a global representation of users and items only, ignoring local representation of users and items.

Table 3 shows that the performance of TANN w/o SGA method is the poorest, which indicates that the selection of meta-paths has a great influence on the recommendation results. The performance of TANN w/o RWRS is better than the performance of TANN w/o SGA; it indicates that the meta-path selection strategy plays an important role compared to the weighted random walk strategy for recommendation task. In addition, as results illustrate, the performance of TANN w/o local is better than that of the above two methods. This can be mainly credited to the fact that the nodes and meta-paths in HIN contain rich implicit

Table 2: Manually selected meta-paths and the meta-paths selected by SGA in three datasets.

Datasets	Manually selected meta-paths	Meta-paths selected by SGA
MovieLens	UMTM, UUUM, UMUM, UMMM	UMTM, UUMM, UMUM, UMMM
Douban movie	UMMM, UMUM, UUUM, UUMM	UMTM, UMUM, UMMM, UUMM
Yelp	UUUB, UUBB, UBUB, UBCB	UUBB, UBUB, UBBB, UBCB

Table 3: Experimental results of TANN-based recommendation approach on the four datasets.

Model	MovieLens			Douban movie			Yelp		
	NDCG@10	Prec@10	Recall@10	NDCG@10	Prec@10	Recall@10	NDCG@10	Prec@10	Recall@10
TANN	**0.6968**	**0.3476**	**0.2250**	**0.9397**	**0.8082**	**0.1186**	**0.4285**	**0.1363**	**0.2554**
TANN *w/o* local	0.6957	0.3467	0.2242	0.9333	0.8042	0.1177	0.4184	0.1332	0.2488
TANN *w/o* RWRS	0.6901	0.3463	0.2235	0.9301	0.7878	0.1156	0.4120	0.1330	0.2442
TANN *w/o* SGA	0.6895	0.3451	0.2234	0.9087	0.7468	0.1094	0.4063	0.1293	0.2323

and effective information. TANN w/o local method applies SGA to automatically obtain meta-paths avoiding artificial interference, and it uses the weighted random walk strategy to get optimal meta-path instances that can represent the information of heterogeneous information network structure. TANN method, which not only considers information of users, items, and meta-paths but also considers the mutual influence among them, consistently outperforms the other three methods.

5.4. *Comparison with Other Recommendation Approaches.*

Moreover, we compare our proposed TANN method with other four recommendation methods: (1) BPR (Bayesian Personalized Ranking) [53], which is a Bayesian posterior optimization based personalized ranking algorithm; (2) LRML (Latent Relational Metric Learning) [54], which employs an augmented memory model to construct latent relations between each user-item interaction; (3) CDAE (Collaborative Denoising Autoencoders) [55], which uses a Denoising Autoencoder structure to learn users and items' distributed representations; and (4) MCRec (Meta-path based Context for RECommendation) [44], which leverages rich meta-paths and co-attention mechanism. Table 4 compares the experimental results.

From Table 4, we can see that LRML performs the poorest, as it only learns relations that describe each user-item interaction. CDAE learns the corrupted user-item preferences' latent representations that can best reconstruct the full input, so it performs better than LRML. Both LRML and CADE concentrate on explicit feedback. BPR uses not only explicit feedback but also implicit feedback; thus, it can obtain better results than LRML and CADE. MCRec learns representations of users, items, and meta-path based context, as it encodes much more information than the previous methods; its performance is better than the above three methods. MCRec selects meta-path manually and only utilizes global information of users and items, while our proposed TANN method selects meta-path automatically, uses local and global information of users and items, applies a tri-attention mechanism to enhance the users, items, and meta-paths' representations, and its performance consistently outperforms the other four recommendation methods.

5.5. *Impact of Meta-Paths Selection.*

In this set of experiments, we examine whether the automatically selected meta-paths can produce better recommendation performance than manually selected meta-paths. The experiments are conducted on the MovieLens dataset. The optimal meta-path set selected by SGA is UMTM, UUMM, UMUM, and UMMM (P1). We randomly selected three different meta-path sets: UMTM, UMUM, UUUM, and UMMM (P2); UMUM, UUMM, UUUM, and UMMM (P3); and UUMM, UUUM, UMTM, and UMMM (P4). The recommendation performance with different meta-path sets is shown in Figure 2.

We can observe in Figure 2 that the recommendation performance based on the meta-path set which is selected by SGA algorithm is better than the performance based on the three manually selected meta-path sets. The experimental results show that the interference of human factors should be avoided in the construction of heterogeneous information network.

5.6. *Impact of Users' and Items' Local Information.*

We study whether incorporating users' and items' local information can further enhance the recommendation performance. The experiments are conducted on the Yelp dataset. We randomly select a user from the user list and obtain the recommendation list with/without local information of users and items. In Figure 3, the ground-truth of the user's preference movie ids is listed in the middle column, the movie ids using TANN *w/o* local information method are listed in the left column, and the movie ids using TANN *w/i* local information method are listed in the right column. The bold ids in the left and right columns indicate the matched results with the ground-truth ids. We can see that when integrating local information, the recommendation method can find more accurate results than without local

Table 4: Comparison with other recommendation approaches on the three datasets.

Model	MovieLens			Douban movie			Yelp		
	NDCG@10	Prec@10	Recall@10	NDCG@10	Prec@10	Recall@10	NDCG@10	Prec@10	Recall@10
TANN	**0.6968**	**0.3467**	**0.2235**	**0.9397**	**0.8082**	**0.1186**	**0.4285**	**0.1363**	**0.2554**
MCRec	0.6909	0.3411	0.2192	0.9365	0.8080	0.1194	0.4212	0.1343	0.2435
BPR	0.6905	0.3386	0.2176	0.6129	0.2811	0.0393	0.4192	0.1297	0.2342
CDAE	0.6303	0.3039	0.1906	0.6008	0.2716	0.0374	0.3948	0.1167	0.2241
LRML	0.5381	0.2242	0.1418	0.4855	0.2649	0.0256	0.3934	0.1147	0.2124

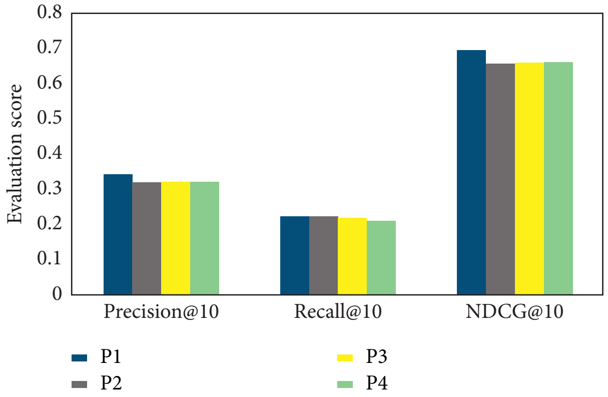


Figure 2: Recommendation performance with different meta-path sets.

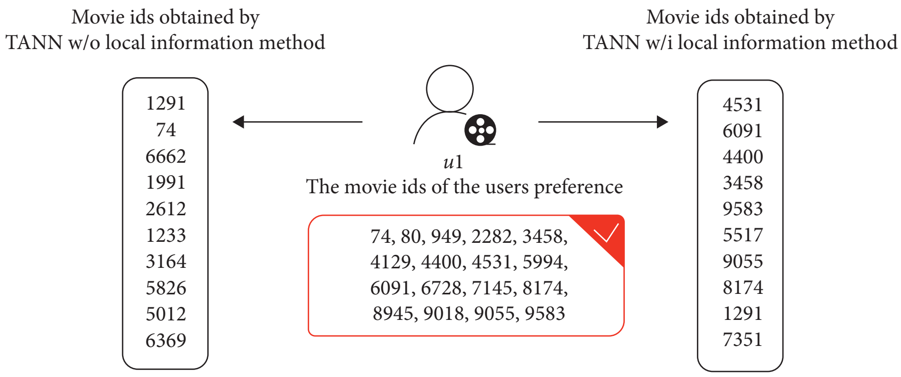
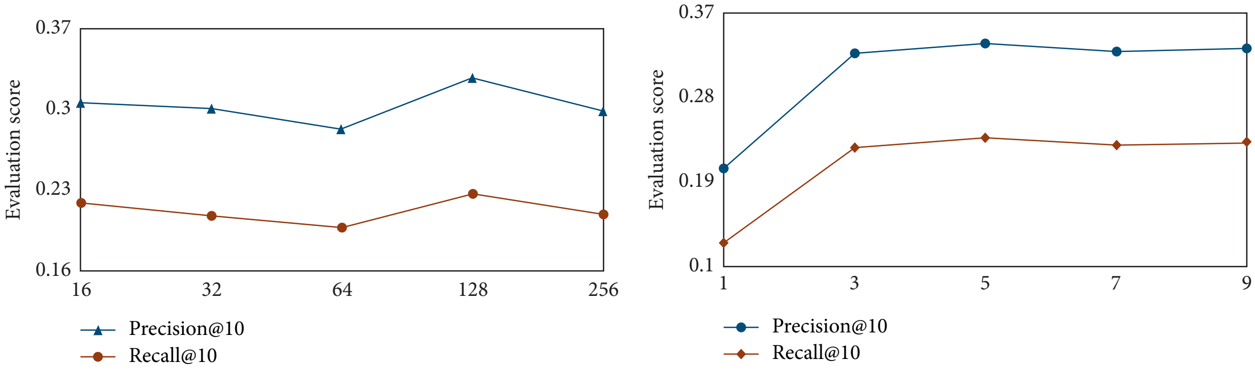


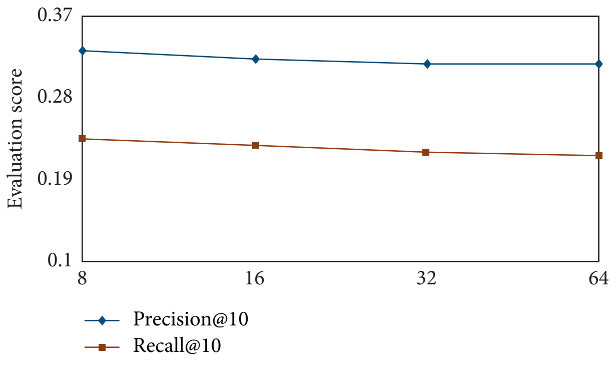
Figure 3: An example of removing local information in the TANN method.



(a)

(b)

Figure 4: Continued.



(c)

Figure 4: Parameter tuning of TANN on MovieLens dataset. (a) The embedding size for the weighting vector of attentive scores. (b) The number of negative samples. (c) The embedding size for the output layer.

information. It illustrates that local information has an impact on improving the recommendation result. Compared with the global information, the local information can reflect the neighborhood characteristics of the nodes centrally.

5.7. Parameter Tuning. Our model includes a few important parameters to tune. In this section, we examine three parameters' performance effect on MovieLens dataset, that is, the embedding size for the weighting vector of attentive scores (i.e., the embedding size of $\mathbf{w}_2$ in equation (14)), the negative samples' number (in equation (18)), and the output layer's embedding size.

For the embedding size of the tri-attention mechanism, we vary it in the set of {16, 32, 64, 128, 256}. As shown in Figure 4(a), our method achieves the best performance when it is 128. For the negative samples' number, we vary it in the set of {1, 3, 5, 7, 9}. We can find from Figure 4(b) that when 5 negative items are taken for each positive item, the evaluation result is the best. For the embedding size of the output layer, we vary it in the set of {8, 16, 32, 64}, and the best result can be obtained when the embedding size of the output layer is 8. The optimal performance is obtained with 128-dimension of the tri-attention mechanism, 5 negative samples, and 8-dimension of the output layer.

6. Conclusion

We present a tri-attention neural network (TANN) model for the recommendation in this paper. We first apply the stud genetic algorithm to automatically select meta-paths and propose a tri-attention mechanism to enhance the mutual influence among users, items, and their related meta-paths. Then, we encode the interaction information among the above three objects which contain more semantic information and can be used for the recommendation task. Extensive experiments on the Douban Movie, MovieLens, and Yelp datasets demonstrated the outstanding performance of the proposed approach. In the future, we will explore other auxiliary information in the heterogeneous information network for the recommendation task.

Data Availability

The authors have presented the website of the three datasets in Section 5.1. Readers can download the datasets from the corresponding link, or can contact the corresponding author.'

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (nos. 61872296 and 61772429), MOE (Ministry of Education in China) Project of Humanities and Social Sciences (18YJC870001), and the Fundamental Research Funds for the Central Universities under Grant 3102019ZDHKY04.

References

[1] L. Sharma and A. Gera, "A survey of recommendation system: Research challenges," *International Journal of Engineering Trends and Technology*, vol. 4, no. 5, pp. 1989–1992, 2013.

[2] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[3] KG. SarwarBM and RJ. KonstanJA, "Application of dimensionality reduction in recommender system — a case study," in *Proceedings of the ACM WebKDD Web Mining for E-Commerce Workshop*, vol. 82, p. 90, Boston, MA, USA, 2000.

[4] X. Ning, C. Desrosiers, and G. Karypis, *A Comprehensive Survey of Neighborhood-Based Recommendation Methods Recommender Systems Handbook*, pp. 37–76, Springer, Boston, MA, USA, 2015.

[5] Z. Li, H. Chen, F. Xiong, X. Wang, and X. Xiong, "Topological influence-aware recommendation on social networks," *Complexity*, vol. 2019, Article ID 6325654, 12 pages, 2019.

[6] X. Ning and G. K. Slim, "Sparse linear methods for top-n recommender systems," in *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, pp. 497–506, Vancouver, Canada, 2011.

[7] Y. C. Hu, "Recommendation using neighborhood methods with preference-relation-based similarity," *Information Sciences*, vol. 284, pp. 18–30, 2014.

[8] C. Paolo, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, ACM, Barcelona, Spain, pp. 39–46, 2010.

[9] S. Pan, X. Wang, H. Yang et al., "Social recommendation with evolutionary opinion dynamics," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[10] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proceedings of the International Conference on Data Mining*, pp. 263–272, Pisa, Italy, December 2008.

[11] A. N. Nikolakopoulos, V. Kalantzis, E. Gallopoulos, and J. D. Garofalakis, "EigenRec: generalizing PureSVD for effective and efficient top-N recommendations," *Knowledge and Information Systems*, vol. 58, no. 1, pp. 59–81, 2019.

[12] F. Christoffel, B. Paudel, C. Newell, and A. Bernstein, "Blockbusters and wallflowers: accurate, diverse, and scalable recommendations with random walks," in *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 163–170, Vienna, Austria, September 2015.

[13] S. Pan, R. Hu, S. Fung et al., "Learning graph embedding with adversarial training methods," *IEEE Transactions on Cybernetics*, vol. 50, 2019.

[14] C. Cooper, S. H. Lee, T. Radzik, and Y. Siantos, "Random walks in recommender systems: exact computation and simulations," in *Proceedings of the 23rd International Conference on World Wide Web*, ACM, New York, NY, USA, pp. 811–816, 2014.

[15] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: representation, acquisition and applications," 2020, http://arxiv.org/abs/2002.00388.

[16] Z. Kang, C. Peng, M. Yang, and Q. Cheng, "Top-N recommendation on graphs," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 2101–2106, Toronto, Canada, 2016.

[17] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proceedings of the Twenty-Fourth International Conference on World Wide Web*, ACM, New York, NY, USA, pp. 278–288, 2015.

[18] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proceedings of the International Conference on Learning Representations*, Vancouver, Canada, 2016.

[19] A. V. Den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," *Neural Information Processing Systems*, pp. 2643–2651, 2013.

[20] R. Hu, G. Long, C. Zhang et al., "Learning graph embedding with adversarial training methods," *IEEE Transactions On Cybernetics*, vol. 50, 2019.

[21] H. Wang, N. Wang, and D. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the Twenty-First ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, NY, USA, pp. 1235–1244, 2015.

[22] X. Yao, B. Tan, C. Hu, W. Li, Z. Xu, and Z. Zhang, "Recommend algorithm combined user-user neighborhood approach with latent factor model," in *Proceedings of the International Conference on Mechatronics and Intelligent Robotics*, Springer, Cham, Switzerland, pp. 275–280, 2017.

[23] H. Zhang, F. Min, and S. Wang, "A random forest approach to model-based recommendation," *Journal of Information and Computational Science*, vol. 11, no. 15, pp. 5341–5348, 2014.

[24] G. mavicius and A. Tuzhilin, "Context-aware recommender systems," in *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys 2008, Lausanne, Switzerland, pp. 191–226, October 2008.

[25] R. Ronen, E. Yomtov, and G. Lavee, "Recommendations meet web browsing: enhancing collaborative filtering using internet browsing logs," in *Proceedings of the IEEE 32nd International Conference on Data Engineering*, pp. 1230–1238, Helsinki, Finland, 2016.

[26] X. Yu, X. Ren, Y. Sun et al., "Personalized entity recommendation: a heterogeneous information network approach," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pp. 283–292, New York City, NY, USA, 2014.

[27] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Metagraph based recommendation fusion over heterogeneous information networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 635–644, Halifax, Nova Scotia, Canada, 2017.

[28] F. Alqadah, C. K. Reddy, J. Hu, and H. F. Alqadah, "Biclustering neighborhood-based collaborative filtering method for top-n recommender systems," *Knowledge and Information Systems*, vol. 44, no. 2, pp. 475–491, 2015.

[29] Z. Yao, J. Wang, and Y. Han, "An improved neighborhood-based recommendation algorithm optimization with clustering analysis and latent factor model," in *Proceedings of the 38th Chinese Control Conference*, pp. 27–30, Guangzhou, China, 2019.

[30] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, ACM, New York, NY, USA, pp. 39–46, 2010.

[31] T. Guo, S. Pan, X. Zhu, and C. Zhang, "CFOND: consensus factorization for co-clustering networked data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 706–719, 2018.

[32] V. Sindhwani, S. S. Bucak, J. Hu, and A. Mojsilovic, "One-class matrix completion with low-density factorizations," in *Proceedings of the 2010 IEEE International Conference on Data Mining, Ser. ICDM '10*, IEEE Computer Society, Washington, DC, USA, pp. 1055–1060, 2010.

[33] F. Xiong, W. Shen, H. Chen et al., "Exploiting implicit influence from information propagation for social recommendation," *IEEE Transactions on Cybernetics*, 2019.

[34] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 89–115, 2004.

[35] S. Baluja, R. Seth, D. Sivakumar et al., "Video suggestion and discovery for youtube: taking random walks through the view graph," in *Proceedings of the 17th International Conference on World Wide Web*, pp. 895–904, Beijing, China, 2008.

[36] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2643–2651, Lake Tahoe, NE, USA, 2013.

[37] X. Wang and Y. Wang, *Improving Content-Based and Hybrid Music Recommendation Using Deep Learning*, pp. 627–636, ACM Multimedia, Seattle, WA, USA, 2014.

[38] H. J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3203–3209, Melbourne, Australia, August 2017.

[39] D. Kim, C. Park, J. Oh, and H. Yu, "Deep hybrid recommender systems via exploiting document context and statistics of items," *Information Sciences*, vol. 417, pp. 72–87, 2017.

[40] T.-A. N. Pham, X. Li, G. Cong, and Z. Zhang, "A general recommendation model for heterogeneous networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 3140–3153, 2016.

[41] H. Chen, H. Yin, W. Wang, H. Wang, Q. V. H. Nguyen, and X. Li, "PME: projected metric embedding on heterogeneous networks for link prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1177–1186, London, UK, August 2018.

[42] X. Yu, X. Ren, Y. Sun et al., "Recommendation in heterogeneous information networks with implicit user feedback," in *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 347–350, Hong Kong, China, 2013 October.

[43] Z. Jiang, H. Liu, B. Fu, Z. Wu, and T. Zhang, "Recommendation in heterogeneous information networks based on generalized random walk model and bayesian personalized ranking," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 288–296, Los Angeles, CA, USA, October 2018.

[44] B. Hu, C. Shi, W. Zhao, and P. Yu, "Leveraging meta-path based context for top- N recommendation with a neural co-attention model," *Knowledge Discovery and Data Mining*, pp. 1531–1540, 2018.

[45] Y. Sun, J. Han, X. Yan, S. P. Yu, and T. Wu, "PathSim: meta path-based top-K similarity search in heterogeneous information networks," in *Proceedings of the 37th International Conference on Very Large Data Base*, Seattle, Washington, September 2011.

[46] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, Perth, Australia, April 2017.

[47] C. Desrosiers and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender Systems and Book*, pp. 107–144, Springer, Boston, MA, USA, 2011.

[48] W. Khatib and P. J. Fleming, "The stud ga: a mini revolution?" in *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, Springer, New York, NY, USA, pp. 683–691, 1998.

[49] G. E. Hinton and R. Salakhutdinov, "A better way to pretrain deep Boltzmann machines," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2447–2455, Lake Tahoe, NE, USA, 2012.

[50] Y. Koren, "Factor in the neighbors," *ACsM Transactions on Knowledge Discovery from Data*, vol. 4, no. 1, pp. 1–24, 2010.

[51] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Proceedings of the 2008 ACM Conference on Recommender Systems*, pp. 23–25, Lausanne, Switzerland, October 2008.

[52] D. Kingma and J. Ba. Adam, "A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015.

[53] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: bayesian personalized ranking from implicit feedback," 2012, http://arxiv.org/abs/1205.2618.

[54] Y. Tay, L. Anh Tuan, and S. C. Hui, "Latent relational metric learning via memory-based attention for collaborative ranking," in *Proceedings of the 2018 World Wide Web Conference*, pp. 729–739, Lyon, France, April 2018.

[55] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 153–162, San Francisco, CA, USA, 2016.