



Multi-Behavior Enhanced Heterogeneous Graph Convolutional Networks Recommendation Algorithm based on Feature-Interaction

Yang Li, Fangtao Zhao, Zheng Chen, Yingxun Fu & Li Ma

To cite this article: Yang Li, Fangtao Zhao, Zheng Chen, Yingxun Fu & Li Ma (2023) Multi-Behavior Enhanced Heterogeneous Graph Convolutional Networks Recommendation Algorithm based on Feature-Interaction, Applied Artificial Intelligence, 37:1, 2201144, DOI: [10.1080/08839514.2023.2201144](https://doi.org/10.1080/08839514.2023.2201144)

To link to this article: <https://doi.org/10.1080/08839514.2023.2201144>



© 2023 The Author(s). Published with license by Taylor & Francis Group, LLC.



Published online: 12 Apr 2023.



Submit your article to this journal [↗](#)



Article views: 717



View related articles [↗](#)



View Crossmark data [↗](#)



Multi-Behavior Enhanced Heterogeneous Graph Convolutional Networks Recommendation Algorithm based on Feature-Interaction

Yang Li, Fangtao Zhao, Zheng Chen, Yingxun Fu, and Li Ma

School of Information science and technology, North China University of Technology, Beijing, China

ABSTRACT

Graph convolution neural networks have shown powerful ability in recommendation, thanks to extracting the user-item collaboration signal from users' historical interaction information. However, many existing studies often learn the final embedded representation of items and users through IDs of user and item, which cannot make well explanation why user choose the item. By making good use of item's attribute, the networks will gain better interpretability. In this article, we construct a heterogeneous tripartite graph consisting of user-item-feature, and propose the attention interaction graph convolutional neural network recommendation algorithm (ATGCN). We embed multi-feature fusion of users and items into the user feature interaction layer by using multi-head-attention, which explore the user's potential preference to update the user's embedded representation. Through the neighborhood aggregation of graph convolution, the feature neighbors' aggregation of items is constructed to achieve higher-order feature fusions, and the neighborhood aggregation of users and items is carried out on the historical interaction information. Then, the final embedding vector representations of user and item are obtained after many iterations. We verify the effectiveness of our proposed method on three publicly available datasets and ATGCN has improved 1.59%, 2.03%, and 1.27% in normalized discounted cumulative gain (NDCG), Precision and Recall, respectively.

ARTICLE HISTORY

Received 29 December 2022
Revised 19 March 2023
Accepted 3 April 2023

Introduction

With the explosive growth of information, recommender systems have great advantages in solving the problem of information overload. Recently, graph neural network (GNN) technology has been used more and more in recommender systems (Wu et al. 2022). The GNN-based recommendation model is classified in detail. Heterogeneous graph neural networks are developing rapidly (Cai et al. 2022; Fan et al. 2019; Ji et al. 2021; Zhang et al. 2021; Luo et al. 2020; Zhang et al. 2019). Among them, Fan et al. (2019) used rich

CONTACT Li Ma mali@ncut.edu.cn School of Information science and technology, North China University of Technology, No. 5 Jinyuanzhuang Road, Beijing, Shijingshan District 100144, China

© 2023 The Author(s). Published with license by Taylor & Francis Group, LLC.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

structural information, a heterogeneous graph neural network guided by metapath is designed to learn embeddings of recommended objects. A random walk strategy with reboot is introduced, and strongly correlated heterogeneous neighbors with a fixed size are sampled for each node and grouped according to node type (Zhang et al. 2019). Ji et al.'s (2021) model users' preferences for news based on the temporal dynamics of user interaction with news. Zhang et al. (2021) proposes a heterogeneous global graph learning framework that makes full use of user-user, user-item interaction and item-item similarity. Cai et al. (2022) utilizes users, items, and related multimodal relationship information to alleviate the sparsity of user attributes, and collected relevant neighbors of new users through multiple sampling operations. Luo et al. (2020) uses dynamically constructed heterogeneous graphs to encode the properties of events and their surroundings, learned the impact of historical behavior and surroundings on current events, and generated representations of valid events to improve model efficiency.

Recently, with the rise of graph convolution neural network, because graph neural network strong learning ability from non-Euclidean data and most of the data in real recommendation scenarios are non-Euclidean structure, graph convolutional neural network (GCN) model has also made considerable achievements in recommendation (Cao et al. 2019; He et al. 2020; Jin et al. 2020; Liu et al. 2020, 2022; Sun et al. 2020; Wang et al. 2019; Yu et al. 2020). The core idea of GCN model is to update the information of the central node by gathering the information of the neighbor nodes of the central node and using the neighbor nodes. Through this communication method, we can obtain the influence of remote nodes on the central node by simply stacking several layers of networks, so that we can learn the embedded representation of users and items more effectively to improve the accuracy of recommendations.

Among them, Yu et al. (2020) increases relational data by encoding high-order and complex connection patterns to ensure the validity of identified neighborhoods. Multi-behavior recommendation with multiple types of user item interaction solves the serious data sparsity issues or cold start faced by the traditional recommendation model (Jin et al. 2020). Liu et al. (2022) uses relational attention and neighbor aggregation to handle node heterogeneity. In this way, social information is well integrated with user project interaction. A joint learning model was built by combining recommendation and knowledge graph. Different from other knowledge graph-based recommendation methods, they pass the relationship information in knowledge graph (KG) to get the reason why users like a certain item (Cao et al. 2019). For example, if a user watches multiple movies directed by the same person. It can be inferred that when users make decisions, the director relationship plays a key role, which helps to understand users' preferences more precisely. In the work by He et al. (2020), the author's goal is to simplify the design of GCN, and to make algorithm more suitable for recommendation. They proposed a new

model called LightGCN, which only includes the most important component neighborhood aggregation in GCN for recommendation. In a word, the model updates the embedded representation of users and items through linear propagation on the user-item interaction graph, and takes the weighted sum of user and item representations learned at each level as the final embedded representation. This linear, simple and clean model is easier to train and implement.

Graph convolution network is used to describe complex interactions (Liu et al. 2020). In addition, in order to learn node representation, a messaging strategy is used to aggregate messages passed from other directly linked types of nodes (for example, users or attributes). Thus, it can combine associated attributes to enhance item and user representation learning, so as to naturally solve the problem of missing attributes. However, it is also a problem how to fully consider the domain information and the user item bipartite map together to provide users with accurate, diverse, and interpretable recommendations. The previous collaborative filtering (CF) method cannot model the domain information of users and items very well, so it inevitably faces the problem of data sparsity. Sun et al. (2020) proposes a new framework, NIA-GCN. It can use the heterogeneity of user item bipartite graph to explicitly model the relationship information between adjacent nodes. That is, a new cross-depth integration (CDE) layer is proposed to capture the item-item, user-user, and user-item relationships in the adjacent regions of the graph. It allows prediction to take into account more complex relationships between graph at different depths. A new method, knowledge graph attention network for recommendation (KGAT), is proposed based on knowledge map and attention mechanism (Wang et al. 2019). The attribute information between the item and the user connects the instances of the user's item together, and explains that the user and the item are not independent of each other. So, this method combines item and user history interaction information and knowledge atlas. A new network structure is formed, and high-order link paths are extracted to represent nodes in the network.

There are few studies on heterogeneous convolution recommendations. In the work by Liang et al. (2022), not only user-item interaction is considered, but also dynamically established user-user and item-item interaction are considered. The representation of convolutional learning focuses on the heterogeneous graph of learning and project content information. Jing et al. (2022) learns the representation of new items and users in dynamic graphs by constructing multiple discrete dynamic heterogeneous maps from interactive data to mine user preferences, item dependencies, and user behavior similarities. The application of heterogeneous convolutional neural networks in other fields, node classification, combines the optimal part of PTE and text graph convolutional networks (TextGCN). The main idea (Ragesh et al. 2021) is to use

heterogeneous convolutional learning feature embedding and export document embedding to better use text classification problems. For learning the dynamic preferences of users, a new dynamic heterogeneous convolutional network is proposed (Yuan et al. 2021), and the structural characteristics of social graph and dynamic propagation graph are jointly learned. Then, the time information is encoded into the heterogeneous map. Predictive tasks that use multi-head attention to capture contextual dependencies to facilitate information dissemination. Good to learn the user's preferences.

However, most of GCN models designed by using user-item history interaction diagrams are built with user IDs and item IDs. Only the embedding of user IDs and item IDs is learned finally. In this way, the results recommended by the model lack of interpretability.

It is worth noting that most of GCN models designed based on user-item interaction diagrams only use item id and user id to build the entire model. The final learned embedding is only the embedding of user ID and item ID. The results recommended by this model are not interpretable. However, in the real data, most of the users and items in the recommended scene have very rich attribute information. For example, in the movie recommendation, we know the user's gender, age, occupation. The director of the movie, the type, which country it belongs to, etc., and in the music recommendation we obtain the style of the music, singers, etc. If the user's preference for different item attributes is modeled on the basis of considering the historical interaction record of the user and the item. In this way, we can know why the user chooses this item, and we can also indirectly filter out some unreal preference items.

Based on the above considerations, we also model the user's preference for item attribute characteristics on the premise of considering the historical interaction of users' items. Therefore, we design a heterogeneous tripartite graph composed of user-item-feature, and implement the recommended model by passing information, attention interaction graph convolution neural network (ATGCN), which models the user's historical preference with multiple features of the item, also takes into account the historical interaction between item and user, and fuses them together to implement recommendations. Specifically, we design a user-feature interaction layer that utilizes user and feature information to capture preference information between users and features. Eventually, we design a user-item-feature map convolutional neural network aggregation layer to learn the embedded representation of users and items, respectively, so that the learned user and item representations contain the attribute features of the item, which can greatly improve the effect of recommendation. In summary, our work has made the following contributions:

- We used the multi-head attention mechanism to learn the user's preference for item multi-attribute features, and modeled the user-item-feature heterogeneous tripartite graph from the real scene.
- We presented attention interaction graph convolutional neural network (ATGCN) model, which can more accurately mine the internal associations between users and multiple features of the item.
- We performed an experimental analysis on three commonly used large datasets. Compared to other baselines, our proposed model has significant improvements in NDCG, recall, and accuracy.

Methodology

Heterogeneous Tripartite Graph

As shown in [Figure 1](#), it describes the specific method of generating the graph. The heterogeneous tripartite graph shows the heterogeneity of the user and the item next-hop node. We use a heterogeneous tripartite graph composed of a user-item-feature. Use the graph convolution process to derive the representation of items and users, we plan to embed the feature information into the item and user representation. The specific construction method of the graph is that we obtain the item-feature pair $\langle I, F \rangle$ according to the attribute feature of the item, and then obtain the user-feature pair $\langle U, F \rangle$ through the item as an intermediary. Meanwhile, we can, through historical interaction information, obtain the user-item pair $\langle U, I \rangle$. Finally, we get the following adjacency matrix:

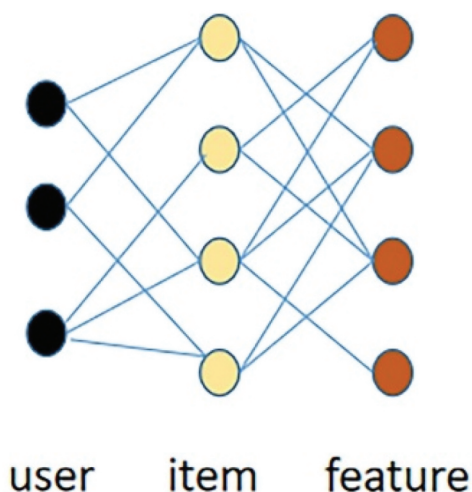


Figure 1. The user-item-feature heterogeneous tripartite graph.

$$A = \begin{bmatrix} 0 & R \\ R^T & 0 \end{bmatrix} \quad (1)$$

The obtained adjacency matrix A indicates that $A=A^T$ should be satisfied. That is, it satisfies $A = D^{\frac{1}{2}}\Lambda D^{\frac{1}{2}}$, i.e. According to the definition of the Laplace operator in GCN (Berg, Kipf, and Welling 2017), the graph volume operator in our ATGCN fits well. We map user ID, item ID, item feature into a vector. Predict each user's final preference for an item by embedding representation vectors that are eventually learned.

Embedding Layer

Since features are divided into categorical and numeric features, they may not be able to interact with users and items, so a common approach is to represent them as low-dimensional spaces (for example, word embeddings). Specifically, to allow the interaction between feature and ID, we embed the user ID, item ID, and item feature into the same low-dimensional feature space.

User-Features Interaction

In the realistic recommended scene, for example product recommendation, movie recommendation, etc. We do not easily obtain attribute information of the user, but we can effectively obtain the attribute information of the item. Our purpose is to simulate user's preference for certain item features. First, we can define n^d as the attribute information for the item where d denotes the dimension of the attribute information and n denotes the number of attribute information. But, when the item features and user features are not clear, the item embedding will not be consistent with the user embedding, which may cause the model to fail to train properly. So, based on the historical interaction information between the item and user, we average the sum of all the features of the item that the user has interacted with to represent the initial feature of the user. As shown below:

$$F_u = \frac{1}{\text{sum}(R_{ui} \in E)} \left(\sum_{i \in R_{ui}} F_i \right) \quad (2)$$

Where R_{ui} denote the interaction set between item and user history. $\text{Sum}()$ denote the total count of interaction features, and F_i denote the feature of item. After that, we will $e_f \in R^{n \times d}$ it is defined as feature embedding matrix to obtain the embedding denote of the feature. For convenience, we map different features in the same low-dimensional space, and then combine different features. In order for users to learn the preferences of different combinations of features, and extract semantic information from different combinations of features. We use a multi-head attention model to build users' preferences for

items with different characteristics (Vaswani et al. 2017). In a word, we use multi-head attention to discover which item features are more consistent with user preferences. We map user and item features and the embedded representation of user features to the low-dimensional space of the same dimension, which facilitates us to model the high-level interactive representation between different features and users. Next, we will explain in detail how to define it. Firstly, we define the denote of different combinations of item features, as follows:

$$\alpha_{i,m}^h = \frac{\exp\left(\psi^{(h)}\left(F_u^i, F_u^m\right)\right)}{\sum_{l=1}^M \exp\left(\psi^{(h)}\left(F_u^i, F_l^m\right)\right)} \quad (3)$$

where m and i denote the currently computed user features, h denotes the number of attention heads, and M denote the total count of item features. $\psi^{(h)}\left(F_u^i, F_u^m\right) = \langle W_q^{(h)} F_u^i, W_k^{(h)} \rangle$ is the magnitude of the correlation coefficient between the current user features i and m . $W_k^{(h)}$ and $W_q^{(h)}$ are weights. The user feature i and m are normalized by the final correlation coefficients of the transformations on each header h , and finally the formula (3) is represented as $\alpha_{i,m}^h$. The interactive denote of the features is as follows:

$$e_{ui}^m = \sum_{h=1}^H \sum_{k=1}^M \alpha_{F_u^m, F_u^i}^{(h)} \left(W_V^{(h)} e_{F_u^i} \right) \quad (4)$$

where $W_V^{(h)}$ is weight, $e_{u,i}^{(m)}$ denote the interactive feature embedding representation of user feature m and i . We can then define user preferences for different feature interaction embeddings. It is defined as follows:

$$\alpha_u^m = \sum_{h=1}^H \frac{\exp\left(\psi^{(h)}\left(e_u, e_u^m\right)\right)}{\sum_{l=1}^M \exp\left(\psi^{(h)}\left(e_u, e_u^m\right)\right)} \quad (5)$$

Where $\psi^{(h)}\left(e_u, e_u^m\right)$ denote the user preference for each combinations feature, h denotes the count of heads, and m denotes the total count of feature combinations. We can learn the final denotes of the user preference for the current interaction features:

$$e_{F_u} = ReLU\left(\sum_{h=1}^H \sum_{l=1}^M \alpha_u^{m(h)} \left(W_V^{(h)} e_u^{m(h)} \right)\right) \quad (6)$$

Where $e_u^{w(h)}$ and $\alpha_u^{w(h)}$ denote preference vector on each head and preference coefficient, $ReLU()$ denote the nonlinear activation function.

Graph Convolutional Neural Network Aggregation Layer

Historical interaction information between items and users is a trustworthy source of user preference message. We refer to the graph convolution neural

network method. Modeling users' high-level preferences for item characteristics and items by considering the attribute feature of the item. The specific graph convolution operation is shown below. First, we aggregate the feature neighbors of the item to obtain the embedding vector of the item containing the feature:

$$e_{F_i} = \sigma \left(\sum_{n \in N_F} W F_n \right) \quad (7)$$

Where F_n represents the feature neighbor of the item, W is the weight coefficient N_F is the feature neighbor set of the item σ represents the activation function. Then, the user node denote and item node denote are obtained by aggregating neighbors on the user-item interaction graph is shown below:

$$e_u^{(k)} = \text{softmax} \left(\sigma \left(\sum_{i \in N_i} W^{(k)} e_{F_i}^{(k-1)} \right) \right) \quad (8)$$

$$e_i^{(k)} = \text{softmax} \left(\sigma \left(\sum_{i \in N_u} W^{(k)} e_{F_u}^{(k-1)} \right) \right) \quad (9)$$

Where e_{F_u} is the denote of users learned from the feature interaction layer, and it embeds the preference for multi feature combination of the item. *softmax* representation of the normalized function and k representation of the current count of layers. By aggregating the multi-hop neighbors of the node, each node will update the information containing the neighbor, making the recommendation better. Finally, our model updates the state denote of the current node according to the state of the multi-hop neighbor. After K -layer convolution, we combine the results from each layer to get the final user and item node denote:

$$e_u = e_u^{(0)} \parallel e_u^{(1)} \dots \parallel e_u^{(k)} \quad (10)$$

$$e_i = e_i^{(0)} \parallel e_i^{(1)} \dots \parallel e_i^{(k)} \quad (11)$$

Where i and u represent the final learned item embedding vector representation and the user embedding vector representation, respectively. We noticed that after initializing the feature of the item feature and the user feature. The dimensions of item embedding e_i and user embedding e_u have changed, updating from the original set $R^{n \times d}$ to $R^{n \times 2d}$.

Prediction Layer

After the information dissemination of the graph convolution aggregation layer, we can clearly understand the user preferences of each item. It is also possible to infer which attribute features of the item the user likes. After that, we perform a dot product operation on the final learned user and item embedding vector representations to calculate the user's preference for each item. For users, our model is primarily designed to generate a set of items that

users might like. Therefore, given a user u , a candidate item i and its corresponding learning embedding, we calculate an estimated score y to measure user u is preferred for item i . We define the user's preference score for the item as follows:

$$y_{ui} = \sigma(e_u^T e_i) \quad (12)$$

Where $\sigma(x) = \frac{\exp(x)}{1+\exp(x)}$ is the sigmoid function, e_u and e_i are generated embeddings of user u and candidate item i respectively. Users can express their preferences for diverse types of items in various ways. For example, the distance between user embeddings and each item embedding can be calculated to get the user's degree of preference. In our work, dot product is used as the way of forecasting (He et al. 2017).

Model Training

The total framework of our model is shown in Figure 2. BPR (Rendle et al. 2012) is used to train our model. We calculate each user's preference for negative and positive samples separately, and then update the parameters that can be learned based on the difference between negative and positive samples. Below is our loss calculation function:

$$Loss = \sum_{(u,i,j) \in O} -\ln \sigma(y_{ui} - y_{uj}) + \lambda \|\Theta\|^2 \quad (13)$$

In (13), $\Theta = \{(u, i, j) | (u, i) \in R^+, (u, j) \in R^-\}$ denotes the paired training samples, where R^- denotes the set of negative samples and R^+ represents the set of positive samples. $\sigma()$ denotes the nonlinear activation function, and λ controls the L_2 regularization strength to prevent over-fitting and Θ denotes the trainable parameters in the model. We use Adam (Kingma and Ba 2014) as our optimizer.

Complexity Analysis

The first is the mapping of nodes. That is, we map vectors e of dimensional d to dimensional d^\sim space, W is the parameter matrix of the dd^\sim dimension. Therefore, for all nodes that need to be mapped, the computational sophistication is $O(Ndd^\sim)$, n is the count of all nodes. Second, the mapping in the process of calculating the attention score of the bulls. In the process of calculating the attention coefficient, the user-item graph needs to be calculated as many times as there are edges, and its calculation complexity is $O(h(E \times d^\sim))$, where e is how many edges there are in the user-item graph, h is the count of heads of the multi-head attention. The subsequent aggregation links are mainly weighted summation operations, and no longer involve

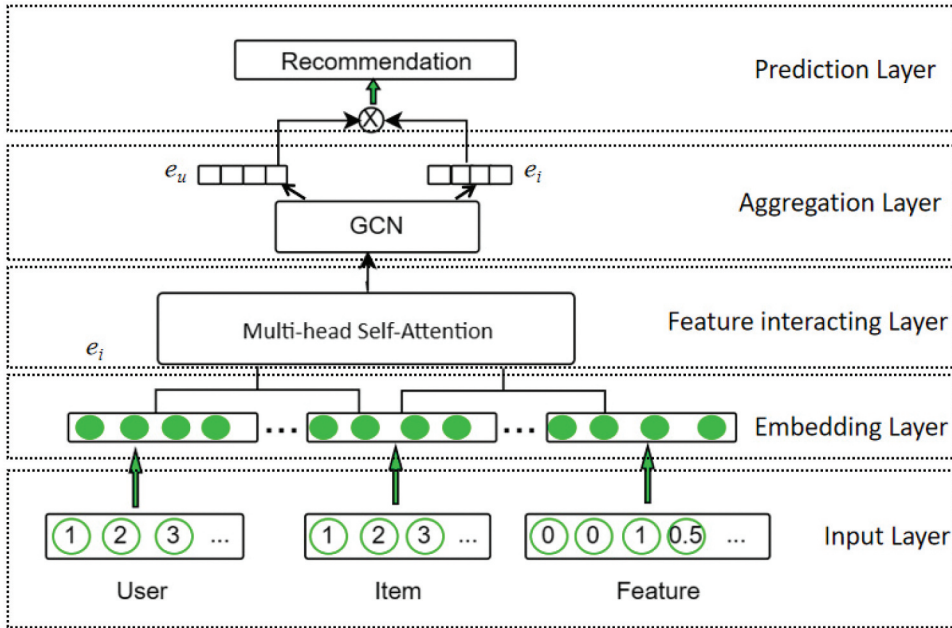


Figure 2. Overall model architecture graph of ATGCN.

high-complexity multiplication operations. Therefore, the computational complexity of the ATGCN model is $O(Ndd^{\sim} + h(E \times d^{\sim}))$.

Experiment

Dataset Description

We evaluated the validity of our ATGCN by experimenting on three publicly large datasets: Amazon-book, MovieLens-1 m, and Amine, all of which are publicly available. The three datasets differ in size, domain, and sparsity. Table 1 shows the data for each of the three datasets.

Amine: This one is a dataset for anime recommendations. We retain at a minimum 20 interactions per user.

MovieLens-1 m: Movie recommendation dataset that is widely used in personalized recommendations (Harper and Konstan 2015).

Amazon-book: A frequently used dataset of product recommendations (Ricci, Rokach, and Shapira 2011). We also retain interaction information for at a minimum 20 items per user.

Table 1. Data set statistics.

Dataset	#Users	#Items feature	#Interactions
Amine	15506	34325 44	2601998
MovieLens-1m	6040	5953 18	1000209
Amazon-Book	51639	84355 11	2648963

We recorded all items in the training set that the user interacted with as a positive sample and randomly sampled from items that the user had never interacted with as a negative sample.

Experimental Setting

Evaluating Indicator

Output the user's preference score for each item. We applied three widely used evaluation methods to assess the effectiveness of top-K recommendation and preference rankings: recall @K, precision @K, and NDCG @K. We make different experiments on different K values, and then the experimental results are analyzed in detail. The recall rate represents the probability of being predicted as a positive sample in the actual positive sample. The higher the value is, the more accurate it is. Precision means that we can accurately predict the results of positive samples, which represents the prediction accuracy of the results of positive samples. NDCG is used to evaluate the sorting results in the recommended tasks, and its results reflect whether the returned list is excellent. We compare our proposed ATGCN with the following baseline to demonstrate the effectiveness and efficiency of our model:

ItemCF: The algorithm learns the representation of items based on the historical interactive information of user-items, considers similar item vectors to represent closeness, and finally recommends items according to similarity.

MF: It only leverages the user-item interaction information as the target value. The matrix decomposition of BPR loss optimization.

NeuMF: This method is one of the most advanced graph neural collaborative filtering models. It uses multiple hidden layers at the top and embedded connections between items and users to capture their nonlinear feature interactions.

NGCF: neural graph collaborative filtering (NGCF) is the most advanced graph convolutional neural network model, which integrates graph neural networks into recommendation systems. Here, only historical interaction information is used to construct a bipartite diagram of user items.

PUP (Zheng et al. 2020): It is a relatively advanced graph neural network model, which models the impact of commodity prices on users. The potential relationship between users to commodities and commodities to prices is effectively established.

PinSAGE (Hamilton, Ying, and Leskovec 2017): This is an algorithm based on graph convolutional neural networks that combines random walk sampling and graph convolution operations to obtain a representation of nodes.

ATGCF (Ma et al. 2022): The algorithm uses the recommendation model of graph neural collaborative filtering, while the algorithm in this paper uses the graph convolutional neural network model.

Parameters Settings

Our ATGCN algorithm is based on the Pytorch implementation with a fixed feature embedding size of 64. Our models are all optimized using the Adam optimizer with a batch size of 4096. The hyperparameters is set to: The learning rate is between [0.0001, 0.0005, 0.001, 0.005]. For each dataset, we compose the training set with 80% of each user's historical interactions, and remaining 10% is used for the test set and 10% for the validation set. But when we trained ATGCN, we found that only 10% of the training numbers can also get very good results. This shows that our model performs well in the face of a small number of samples. L_2 the normalization coefficient is between [1e7, 1e6, 1e5, 1e4, 1e3].

Results and Discussion

We compared ATGCN performance to all other baselines. Tables 2, 3 and 4 report the results for the three datasets. The data from the table is shown as follows:

Model Effectiveness

As shown in the table above, we found that ATGCN essentially exceeded all baselines on all NDCG@K, Precision @K, and Recall @K evaluating indicator. Our models were 1.56% 1.48% and 0.98% higher than those of ATGCF for NDCG, Precision and Recall on the Amine dataset; 2.1%,1.5% and 1.47% on MovieLens-1m dataset; 1.1%, 3.1%, and 1.35% on Amazon-Book dataset. This proves the validity of our model.

Table 2. All evaluation indicators of the dataset amine and improvement of the optimal baseline.

Method	Recall@20	Recall@30	Recall@40	Recall@50	Precision@10	Precision@20	Ndcg@20
ItemCF	0.0286	0.0304	0.0401	0.0412	0.0260	0.0201	0.0115
MF	0.0354	0.0446	0.0524	0.0644	0.0311	0.0231	0.0214
NCF	0.0402	0.0542	0.0644	0.0685	0.0342	0.0256	0.0256
NGCF	0.0434	0.0580	0.0669	0.0772	0.0358	0.0294	0.0294
PUP PinSAGE	0.0482	0.0574	0.0684	0.0784	0.0336	0.0296	0.0304
	0.0429	0.0553	0.0673	0.0756	0.0329	0.0272	0.0295
ATGCF	0.0535	0.0616	0.0702	0.0806	0.0372	0.0308	0.0320
ATGCN	0.0539	0.0621	0.0711	0.0815	0.0377	0.0313	0.0325
Improve	0.7%	0.8%	1.3%	1.1%	1.4%	1.6%	1.6%

Table 3. All evaluation indicators of the dataset MovieLens-1 m and improvement of the optimal baseline.

Method	Recall@20	Recall@30	Recall@40	Recall@50	Precision@10	Precision@20	Ndcg@20
ItemCF	0.0214	0.0238	0.0446	0.0512	0.0332	0.0327	0.0396
MF	0.0384	0.0425	0.0584	0.0748	0.0559	0.0445	0.0425
NCF	0.0401	0.0554	0.0644	0.0864	0.0532	0.0496	0.0489
NGCF	0.0443	0.0647	0.0839	0.1026	0.0604	0.0584	0.0504
PUP	0.0455	0.0665	0.0856	0.1048	0.0588	0.0596	0.0548
PinSAGE	0.0431	0.0607	0.0812	0.1015	0.0496	0.0478	0.0537
ATGCF	0.0482	0.0689	0.0884	0.1070	0.0645	0.0627	0.0581
ATGCN	0.0491	0.0695	0.0892	0.1093	0.0656	0.0635	0.0593
Improve	1.9%	0.9%	0.9%	2.2%	1.7%	1.3%	2.1%

Table 4. All evaluation indicators of the dataset Amazon-book and improvement of the optimal baseline.

Method	Recall@20	Recall@30	Recall@40	Recall@50	Precision@10	Precision@20	Ndcg@20
ItemCF	0.0184	0.0196	0.0312	0.0452	0.0186	0.0195	0.0356
MF	0.0250	0.0267	0.0524	0.0624	0.0254	0.0201	0.0518
NCF	0.0265	0.0288	0.0644	0.0665	0.0262	0.0225	0.0542
NGCF	0.0348	0.0387	0.0669	0.0731	0.0328	0.0257	0.0630
PUP	0.0382	0.0415	0.0684	0.0754	0.0316	0.0285	0.0624
PinSAGE	0.0283	0.0357	0.0589	0.0710	0.0317	0.0229	0.0545
ATGCF	0.0408	0.0426	0.0713	0.0775	0.0358	0.0298	0.0651
ATGCN	0.0416	0.0433	0.0722	0.0783	0.0367	0.0309	0.0658
Improve	1.5%	1.6%	1.3%	1.0%	2.5%	3.7%	1.1%

The effect of multi feature model is well

In the table, it can be clearly seen that the prediction effect of ATGCN and PUP and ATGCF is better than other models, so adding item feature information to the model can increase performance. After the analysis of the experimental results, we find that ATGCN exceed PUP equally by 6.9% on NDCG@20, 11.8% on Recall@20, and 6.5% on Precision@20. This also shows that multi feature fusion will be better than single feature fusion

Our model is very effective in all aspects, and it learns the user's potential preference for multiple feature items. Our ATGCN model is better than those (NGCF and PUP) that can also extract users' potential preferences. This can be explained from two aspects. According to user preferences, NGCF and PUP did not mine users' deep preferences for interactive items, which may result in the loss of information about some raw data. In feature processing, PUP selects only a single feature as the impact of user selection, rather than considering the case of multi-features. However, our ATGCN fully considers the impact of multiple features on user selection, and can learn user preferences for different combinations of features. Through user feature interaction, we can mine user preferences for different mixed features, and recommend to users on the premise of understanding user preferences for item features. Better recommendations can be made to users on the premise. The user node representation after feature interaction contains a lot of item feature information, and then performs a dot product operation

on the final learned user and item embedded vector representation through the prediction layer to calculate the user's multiple feature preferences for each item. According to the experimental results, the prediction effect of ATGCN is superior to other models.

Model Analysis

Because the most important layer of our ATGCN is the user feature interaction layer, we investigated its impact on the overall model performance. So, we studied the influence of different feature embedding methods on the model. Then, the performance of ATGCN with different hyperparameter is studied.

ATGCN-F. Just embed the features of the item to get into the network for learning. This method neither understands the impact of diverse feature combinations on users, nor the preferences of users for different features.

ATGCN-IF. Use self-attention mechanisms to understand user preferences for each item feature. Be based on adding features, the user's preference for diverse features is also taken into account.

ATGCN. Model user preferences for different features and preferences for different interaction features. The multi-head attention mechanism is used to learn the user's preference for diverse feature combinations.

In order to analyze the performance of each feature extract method, we use three methods (ATGCN-F, ATGCN-IF, and ATGCN) to extract user preferences for diverse item features separately. The experimental results are displayed in [Figure 3\(a\)](#), and we can clearly see that the effect of using the multi-head attention mechanism learning interaction features is better than using the attention mechanism learning the user's preference for item characteristics. At the same time, learning using attention mechanisms makes for better performance than just adding item features. Therefore, we design models with

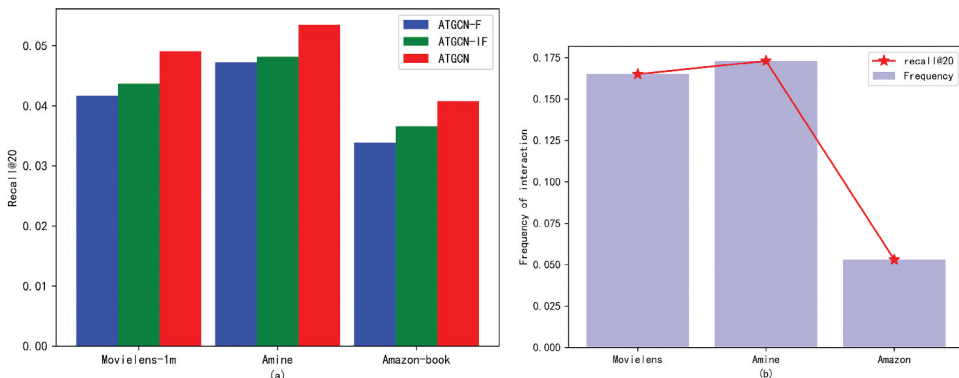


Figure 3. (a) The effect of diverse feature extraction of ATGCN on the model, (b) analytical user feature preferences by ATGCN at diverse interaction frequencies.

better performance. Because it can learn the user's preference for items with diverse combinations of features.

At the same time, we analyzed the sensitivity of the feature interaction frequency of ATGCN. We define the frequency of user interactions by dividing the total count of interactions in the dataset by the count of users (emphasizing the role of features). The results are shown in Figure 3(b). We found that the more frequently features interact, the better the performance of our model. This also justifies ATGCN sensitivity to item features and model design.

We conduct experiments on three datasets to analyze the impact of embedded dimensions of model features on the experimental results. Results of the experiment are shown in Figure 4. It is found from the research results that when the embedded dimensions of features range from 32 to 64, the results of all indicators are basically increased. However, if it is increased higher, the performance will decline steadily, possibly due to over-fitting. The embedding dimension is the d_{\sim} in complexity analysis. According to experimental results, when the embedding dimension becomes large, the efficiency of the model decreases due to the increased complexity.

We experimentally study the effect of $L2$ regularization coefficient λ on the ATGCN model. As shown in Figure 5. The optimal value for Movielens-1 m, Amine, and Amazon-Book are $1e^{-5}$, respectively. When the regularization coefficient λ is greater than $1e^{-5}$, the performance of the model gradually decreases. This shows that too large regularization coefficients will have a bad effect on normal model training, so we do not support larger regularization coefficients.

We use ATGCN algorithm to compare the performance of accuracy and recall analysis of diverse training ratios in ML-1 m dataset. It is found that ATGCN may achieve better performance for less training data. As Figure 6, the ordinate represents the results for each evaluation metric. The decimal point after the ATGCN in the lower right corner of each picture indicates the training ratios samples to the total dataset. We found

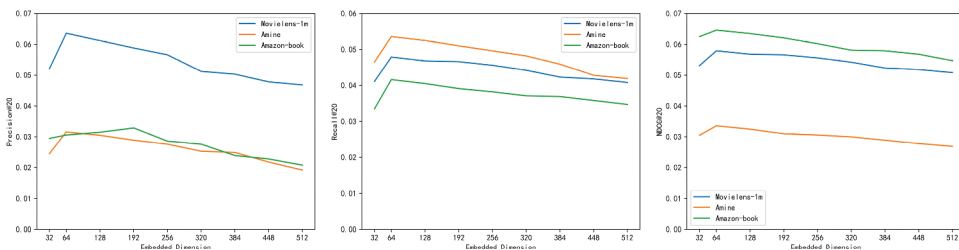


Figure 4. Different measurement results of the embedded dimensions of ATGCN key parameter characteristics on three data sets.

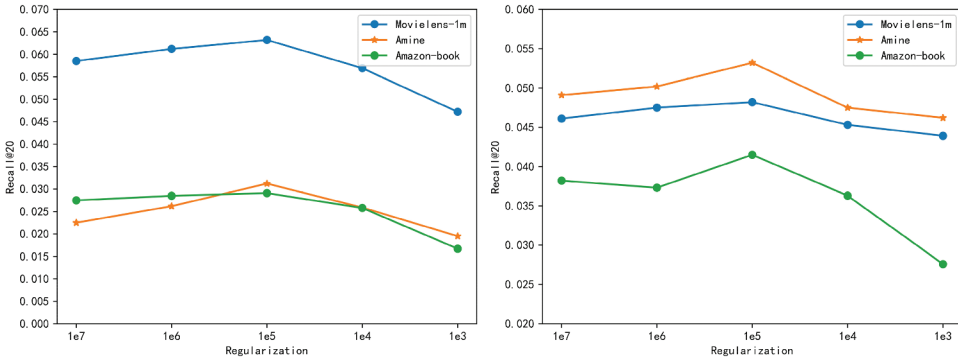


Figure 5. Different regularization coefficients of ATGCN on three data sets λ impact on results.

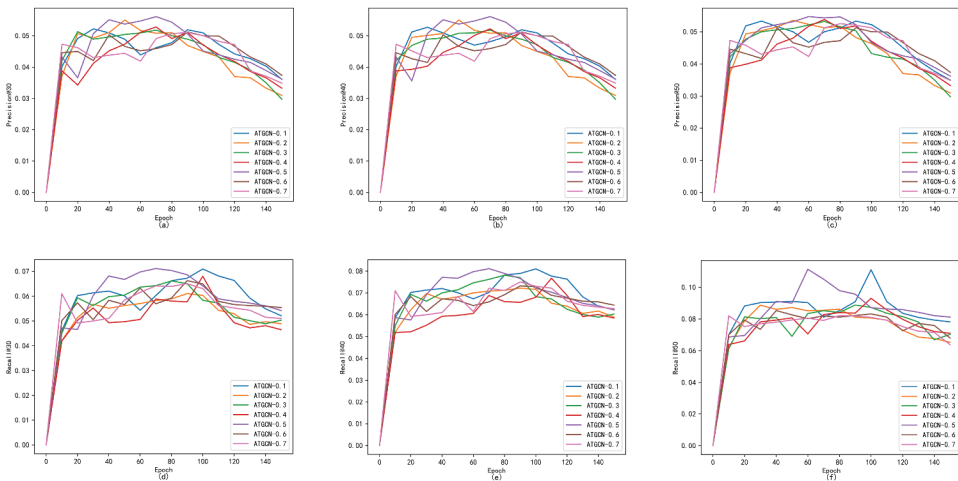


Figure 6. The recall and accuracy of ATGCN to the dataset MovieLens-1m under different training ratios.

that although the ratio of 0.5 was the best, ATGCN also achieved relatively good performance at the ratio of 0.1. Therefore, the results show that even though there are little user interaction data, it can provide users with better recommendations on the premise of understand users' preferences for item features. So, our ATGCN model has good scalability. It is good to recommend datasets with sparse data.

Conclusion

In this work, we construct a heterogeneous tripartite graph form of user-item-feature attributes. On this basis, we propose a new feature interaction graph convolutional recommendation algorithm ATGCN. We embed multi-feature fusion of users and items into the user feature interaction layer, which uses

multi-head attention for learning. It can well explore the user's potential preference for multiple features, so as to update the user's embedded representation. First, the feature neighbors of aggregate items get item embedding that contain features. Then, the neighborhood aggregation of users and items is carried out on the historical interaction information, and the final embedding vector representation is obtained. The limitation of this study is that it does not consider the user's attribute features, and only includes the item's attribute features in the feature interaction part. The future research direction is to conduct feature interaction between user attribute features and item attribute features to mine user preferences for each combination feature. Finally, experiments on three commonly used large datasets prove that our model outperforms the best baseline model in 1.59%, 2.03%, and 1.27% for NDCG, Precision, and Recall.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by National Key R&D Program of China (2018YFB1800302), Natural Science Foundation of China (62001007), Beijing Natural Science Foundation (KZ201810009011, 4202020, L192021, 4212018, 4234083), Research Start-up Fund of North China University of Technology and Jilin Province Science and Technology Development Plan Project (20190201180JC, 20200401076G×).

References

- Berg, R., T. N. Kipf, and M. Welling. 2017. Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263. doi:10.48550/arXiv.1706.02263.
- Cai, D., S. Qian, Q. Fang, J. Hu, and C. Xu. 2022. User cold-start recommendation via inductive heterogeneous graph neural network. *ACM Transactions on Information Systems (TOIS)* 41:1–27. doi:10.1145/3560487.
- Cao, Y., X. Wang, X. He, Z. Hu, and T. -S. Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. The world wide web conference. Pp. 151–61. doi:10.1145/3308558.3313705.
- Fan, S., J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, and Y. Li. 2019. Metapath-guided heterogeneous graph neural network for intent recommendation. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. Pp. 2478–86. doi: 10.1145/3292500.3330673.
- Hamilton, W., Z. Ying, and J. Leskovec. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems* 30 .
- Harper, F. M., and J. A. Konstan. 2015. The movielens datasets: History and context. *Acm Transactions on Interactive Intelligent Systems (Tiis)* 5 (4):1–19. doi:10.1145/2827872.
- He, X., K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. Proceedings of the 43rd

- International ACM SIGIR conference on research and development in Information Retrieval. Pp. 639–48. doi:10.1145/3397271.3401063.
- He, X., L. Liao, H. Zhang, L. Nie, X. Hu, and T. -S. Chua 2017. Neural collaborative filtering. Proceedings of the 26th international conference on world wide web. Pp. 173–82. doi: 10.1145/3038912.3052569.
- Jin, B., C. Gao, X. He, D. Jin, and Y. Li 2020. Multi-behavior recommendation with graph convolutional networks. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Pp. 659–68. doi:10.1145/3397271.3401072.
- Jing, M., Y. Zhu, Y. Xu, H. Liu, T. Zang, C. Wang, and J. Yu. 2022. Learning shared representations for recommendation with dynamic heterogeneous graph convolutional networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 17:1–23. doi:10.1145/3565575.
- Ji, Z., M. Wu, H. Yang, and J. E. A. Íñigo. 2021. Temporal sensitive heterogeneous graph neural network for news recommendation. *Future Generation Computer Systems* 125:324–33. doi:10.1016/j.future.2021.06.007.
- Kingma, D. P., and J. Ba. 2014. *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980. doi:10.48550/arXiv.1412.6980.
- Liang, T., L. Ma, W. Zhang, H. Xu, C. Xia, and Y. Yin. 2022. Content-aware recommendation via dynamic heterogeneous graph convolutional network. *Knowledge-Based Systems* 109185. doi:10.1016/j.knosys.2022.109185.
- Liu, F., Z. Cheng, L. Zhu, C. Liu, and L. Nie 2020. A2-GCN: An attribute-aware attentive GCN model for recommendation. *IEEE Transactions on Knowledge and Data Engineering*. doi:10.1109/TKDE.2020.3040772.
- Liu, Z., L. Yang, Z. Fan, H. Peng, and P. S. Yu. 2022. Federated social recommendation with graph neural network. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13 (4):1–24. doi:10.1145/3501815.
- Luo, W., H. Zhang, X. Yang, L. Bo, X. Yang, Z. Li, X. Qie, and J. Ye 2020. Dynamic heterogeneous graph neural network for real-time event prediction. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Pp. 3213–23. doi: 10.1145/3394486.3403373.
- Ma, L., Z. Chen, Y. Fu, and Y. Li 2022. Heterogeneous graph neural network for multi-behavior feature-interaction recommendation. International Conference on Artificial Neural Networks. Pp. 101–12. doi: 10.1007/978-3-031-15937-4_9.
- Ragesh, R., S. Sellamanickam, A. Iyer, R. Bairi, and V. Lingam 2021. Hetegcn: Heterogeneous graph convolutional networks for text classification. Proceedings of the 14th ACM International Conference on Web Search and Data Mining. Pp. 860–68. doi:10.1145/3437963.3441746.
- Rendle, S., C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618. doi:10.48550/arXiv.1205.2618.
- Ricci, F., L. Rokach, and B. Shapira. 2011. *Introduction to recommender systems handbook. Recommender systems handbook* 1–35. Springer. 10.1007/978-0-387-85820-3_1
- Sun, J., Y. Zhang, W. Guo, H. Guo, R. Tang, X. He, C. Ma, and M. Coates 2020. Neighbor interaction aware graph convolution networks for recommendation. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Pp. 1289–98. doi:10.1145/3397271.3401123.
- Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 .

- Wang, X., X. He, Y. Cao, M. Liu, and T. -S. Chua 2019. Kgat: Knowledge graph attention network for recommendation. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. Pp. 950–58. doi:[10.1145/3292500.3330989](https://doi.org/10.1145/3292500.3330989).
- Wu, S., F. Sun, W. Zhang, X. Xie, and B. Cui. 2022. Graph neural networks in recommender systems: A survey. *ACM Computing Surveys* 55 (5):1–37. doi:[10.1145/3535101](https://doi.org/10.1145/3535101).
- Yuan, C., J. Li, W. Zhou, Y. Lu, X. Zhang, and S. Hu 2021. Dyhgcn: A dynamic heterogeneous graph convolutional network to learn users' dynamic preferences for information diffusion prediction. Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Pp. 347–63. doi:[10.1007/978-3-030-67664-3_21](https://doi.org/10.1007/978-3-030-67664-3_21).
- Yu, J., H. Yin, J. Li, M. Gao, Z. Huang, and L. Cui 2020. Enhance social recommendation with adversarial graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering*. doi:[10.1109/TKDE.2020.3033673](https://doi.org/10.1109/TKDE.2020.3033673).
- Zhang, C., D. Song, C. Huang, A. Swami, and N. V. Chawla 2019. Heterogeneous graph neural network. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. Pp. 793–803. doi:[10.1145/3292500.3330961](https://doi.org/10.1145/3292500.3330961).
- Zhang, Y., L. Wu, Q. Shen, Y. Pang, Z. Wei, F. Xu, E. Chang, and B. Long. 2021. Graph learning augmented heterogeneous graph neural network for social recommendation. arXiv preprint arXiv:2109.11898. doi:[10.48550/arXiv.2109.11898](https://doi.org/10.48550/arXiv.2109.11898).
- Zheng, Y., C. Gao, X. He, Y. Li, and D. Jin 2020. Price-aware recommendation with graph convolutional networks. 2020 IEEE 36th International Conference on Data Engineering (ICDE). Pp. 133–44. doi: [10.1109/ICDE48307.2020.00019](https://doi.org/10.1109/ICDE48307.2020.00019).