

Article

A Multi-Behavior Recommendation Method for Users Based on Graph Neural Networks

Ran Li ¹, Yuxin Li ², Jingsheng Lei ² and Shengying Yang ^{2,*}¹ Guizhou Power Grid Company Limited, Guiyang 550002, China² School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China

* Correspondence: syyang@zust.edu.cn

Abstract: Most existing recommendation models only consider single user–item interaction information, which leads to serious cold-start or data sparsity problems. In practical applications, a user’s behavior is multi-type, and different types of user behavior show different semantic information. To achieve more accurate recommendations, a major challenge comes from being able to handle heterogeneous behavior data from users more finely. To address this problem, this paper proposes a multi-behavior recommendation framework based on a graph neural network, which captures personalized semantics of specific behavior and thus distinguishes the importance of different behaviors for predicting the target behavior. Meanwhile, this model establishes dependency relationships among different types of interaction behaviors under the graph-based information transfer network, and the graph convolutional network is further used to capture the high-order complexity of interaction graphs. The experimental results of three benchmark datasets show that the proposed graph-based multi-behavior recommendation model displays significant improvements in recommendation accuracy compared to the baseline method.

Keywords: multi-behavior recommendation; graph convolutional network; higher-order complexity; graph information transfer network



Citation: Li, R.; Li, Y.; Lei, J.; Yang, S. A Multi-Behavior Recommendation Method for Users Based on Graph Neural Networks. *Appl. Sci.* **2023**, *13*, 9315. <https://doi.org/10.3390/app13169315>

Academic Editor: Stefan Fischer

Received: 21 June 2023

Revised: 25 July 2023

Accepted: 4 August 2023

Published: 16 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Personalized recommendations function to provide users with appropriate products according to user preferences. Determining how to accurately capture user preferences from user behaviors is the core issue of personalized recommendations. Traditional recommendation models [1] usually only rely on a single behavior for a recommendation, which makes them insufficient when extracting complex cooperative signals from users’ multi-type behaviors [2]. Meanwhile, there are serious data sparsity [3,4] and cold-start problems [5,6], especially for certain high-cost and low-frequency behaviors. In the real world, users usually have different types of interactive behaviors. In the face of diversified user behaviors, a big challenge to achieve more accurate recommendations is whether users’ heterostructure behavior data can be processed more finely. The multi-behavior recommendation model jointly considers different types of behavioral semantics, which is of great help to predict the possibility of users adopting target behaviors [7]. For example, on an e-commerce platform, users’ page browsing, shopping-cart additions, and collection behaviors for different items can be used as auxiliary information to help predict users’ purchase intent (target behavior) tasks. Therefore, considering the complex dependencies between multiple behaviors is crucial to accurately predict user preferences.

In order to make full use of dynamic interaction information to better predict user preferences, several multi-behavioral recommendation models have emerged in recent years [2,8]. LightGCN [9] learns user/commodity embeddings via an interaction graph by propagating linearly over the interaction graph, using the weighted sum of the embeddings

learned at each layer as the final embedding. This simple, linear, and neat model is easier to implement and train but does not take into account the variability between behaviors. To distinguish the semantics of different behavior types, the KHGT [10] model assigns different learnable weights to different edges in the user–goods heterogeneous graph and clearly distinguishes which type of user–goods interaction is more important to assist in the task of predicting the target behavior. Nowadays, recommendations based on a graph neural network have been used in many real-world scenarios. The NetEase Cloud Music App introduces the graph model architecture, takes a variety of different types of songs as nodes, and constructs a graph relationship network through the multi-type behavior relationship between users and songs. Jingdong Mall also adopts the model based on a graph neural network proposed by the Jingdong platform, and more accurate recommendation results bring huge benefits to the platform.

Despite the success of these approaches in multi-behavior recommendation tasks, there are some limitations:

- (1) Different types of behaviors can characterize user preferences from different dimensions and complement each other for better learning of user preferences. User/commodity embedding is at the core of recommendation systems. Most current user/commodity embedding representations are a fusion of static features and lack the explicit encoding of a synergistic signal, which is hidden in the user–commodity interaction. Therefore, it is challenging and valuable to capture the behavioral diversity and potential dependencies in recommendations. To address this challenge, existing work models behavioral dependencies by generating specific types of behavioral embeddings through different aggregation approaches to enhance the user/goods representation. For example, MATH [11] uses self-attentiveness to encode pairwise correlations between different types of behaviors and make predictions about the target behavior.
- (2) Traditional multi-behavior recommendation models are implemented based on sequential models, which tend to focus more on the local perspective of multiple sequential behaviors of users. In contrast, graph-based multi-behavior recommendation models focus more on the global perspective of all user behaviors. In a heterogeneous graph constructed using multiple types of behavioral data, users/products are represented as nodes and different types of behaviors are represented as edges of the graph. Graph neural networks are also used to explore higher-order complexity in behavioral heterogeneous graphs due to their powerful learning capabilities. A new graph structure-based model for the novel recommender system NGCF [12] models higher-order connectivity representation in user–commodity interaction graphs by inserting collaborative signals explicitly into the embedding process of users (goods). The user–commodity correlation is well-represented in the embedding space.

In summary, this paper proposes multi-behavior recommendations based on the graph information transfer network method, in which a heterogeneous graph composed of users/commodities first obtains the user/commodity information of a specific type from the graph. The first-order neighborhood information of a particular type of user/goods is obtained from the graph, and the graph information transfer network is used to ensure the interaction behaviors of a particular type have their own semantic information. The above process learns the higher-order neighborhood information in the graph for user/product representation. In the target behavior prediction stage, the above process learns specific types of behavioral representations, which not only provide useful external knowledge but also serve as supervised signals for model optimization.

2. Related Work

Most previous recommendation models [13–16] have been designed for a single type of behavior, and in most cases, behaviors directly related to platform profits were selected for modeling, such as purchase behavior in e-commerce platforms. In practice, however, user behavior is inherently multi-typical (e.g., browsing, favoriting, purchasing, etc.). Different types of user behaviors may exhibit different semantic information to

characterize the diverse user–goods interactions. The existing user–commodity interactions are thus coding functions and are not sufficient to comprehensively learn complex user preferences. Moreover, using only a single behavior may lead to severe cold-start or data sparsity problems. For example, on an e-commerce website, it is difficult to construct a recommendation model based on purchase behavior alone to provide a comprehensive learning model for users without historical purchases, and new users with a purchase history can be aptly recommended.

While realizing the importance of leveraging different types of user behavior at the same time, encoding multiple types of behavioral patterns poses a significant challenge. These different types of interaction behaviors may interrelate in complex ways, providing complementary information for learning about user interests. In addition, although several multi-behavioral user modeling techniques have emerged in recent years, some multi-behavioral user modeling techniques [8,11] have emerged for recommendation, but they fail to capture higher-order information in different user–goods relationships. Inspired by this, applying graph neural networks to recommendations [17,18] is beneficial to consider user–goods interactions in the embedding space higher-order relationships between user–goods interactions are considered in the embedding space.

Recently, graph neural networks have achieved promising results in learning dependencies from graph-structured data [17]. Typically, the core of graph neural networks is to aggregate feature information of neighboring nodes on the graph under a message propagation mechanism [18]. This information dissemination mechanism aggregates the information of higher-order neighbors through nodes, which can further capture higher-order interrelationships and achieve representation learning effectively. In other words, graph neural networks can better solve relationship inference problems as an interpretable model. The most representative of these was the Graph Convolutional Network (GCN), which obtains the representation of the current node by combining the weighted values of neighboring nodes' egress and ingress. Inspired by the effectiveness of graph convolutional networks, recent studies, such as PTGCN [19] and GraphSage [20], utilize graph convolutional networks to explore the user–item interaction graph and aggregate the embeddings of neighboring nodes. These works propagate information among nodes to mine relationships between users and items. Then, graph convolutional networks became a popular research direction, and researchers have conducted a lot of work to study heterogeneous graphs. BiHGH [21] is a new bidirectional heterogeneous graph hashing method. First, it uses heterogeneous graph nodes to initialize then design an Ambigram convolution algorithm to sequentially transfer information, and finally uses Bayesian personalized sorting loss combined with dual similarity preserving regularization to achieve user preference learning. PFCM [22] created a heterogeneous graph that unifies users, items, and attributes and designed a user embedding module based on multimodal content representation to learn user representations. Finally, heterogeneous graph learning was implemented by executing meta path guidance.

3. Methodology

3.1. Problem Statement

Let U and V denote the set of users and goods, respectively, $U = \{u_1, u_2, \dots, u_i, \dots, u_I\}$, $V = \{v_1, v_2, \dots, v_j, \dots, v_J\}$, where I and J denote the number of users and goods. Considering multiple types of interactions, this paper defines a three-dimensional tensor $X \in \mathbb{R}^{I \times J \times K}$ to represent multiple types of interactions (e.g., clicks, favorites, adds, etc.) where K denotes the number of interaction behavior types. A single element $x_{i,j}^k \in X$ with a value of 1 indicates that the k th behavior category is used to interact with user u_i and product v_j , otherwise $x_{i,j}^k = 0$. In a multi-behavior recommendation scenario, the interaction category most associated with the platform benefits will be considered the target behavior (e.g., purchase). Other behaviors will be considered contextual behaviors (e.g., click, favorite, add to cart) and used to provide knowledge that aids the target behavior for prediction. Based on the above definitions, the problem studied in this paper is defined as follows:

Input: Multi-behavior interaction tensors $X \in \mathbb{R}^{I \times J \times K}$ between user set U and item set V under K interaction behavior types.

Output: A prediction function that estimates the likelihood that user u_i will adopt target behavior k to interact with good v_j is possible.

3.2. Model Architecture

In realistic scenarios, often users' behaviors are complex and diverse, and the model first proposes a meta knowledge learner to encode behavioral embeddings considering users' personalized feature attributes. Based on this, the graphical volume was combined with an attention mechanism to capture multiple behavioral patterns with high-order connectivity on the user–goods interaction graph. Finally, complex cross-type behavioral dependencies are captured by a prediction layer. Multiple types of user behavior can be used not only to tune the parameters of the graph neural network model but also to guide the prediction phase by injecting monitoring signals. The model architecture is shown in Figure 1.

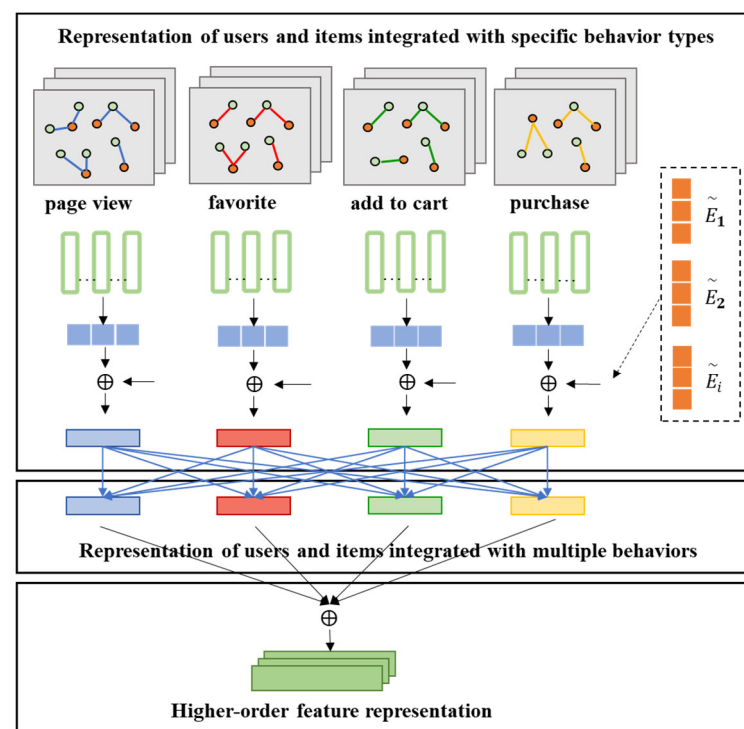


Figure 1. Model architecture diagram of multi-behavior perception.

3.3. Embedding Module Incorporating First-Order Neighborhood Information

In a realistic scenario, the behavior habits of different users are very different. For example, User A is used to collect most of the products in the process of browsing, while User B only collects the products he is most interested in, which shows that the collection behavior has little reference value for User A, while for User B, the collection behavior has little reference value for user A, but has great influence on the products collected by user B. Therefore, the design goal of this module is to capture the first-order neighborhood information of entities in the interaction graph under different behavior categories and inject their corresponding weights into the initial embedding of goods and users, so as to generate a feature representation incorporating the first-order neighborhood information. In the bipartite graph composed of user entities and commodity entities, this module learns the representations of commodity entities and user entities under different behavioral categories, respectively, by combining the initialized IDs of user u_i and v_j by aggregating

the initialized ID embedding representations E_i and E_j of user u_i and commodity v_j with the first-order neighborhood information to obtain the fused contextual feature vector.

Given the ID embedding representation E_i of the initialized user u_i , the following formula is used to learn personalized specific behavior embedding.

$$\begin{cases} P_{i,k} = \frac{E_i \parallel \sum_{j \in N_i^k} E_j}{\sqrt{|N_i^k| |N_j^k|}} \\ W_{j,k} = M \cdot P_{j,k} + Z \\ \tilde{E}_j = W_{j,k} E_j \end{cases} \quad (1)$$

where N_i^k denotes the set of goods that user u_i interacts with under k behavior types, and N_j^k denotes the set of users that interact with good v_j under k behavior types. $\parallel$ denotes the splicing operation of the vector. Here, $\sqrt{|N_i^k| |N_j^k|}$ is the normalization factor. $P_{i,k}$ is the interaction pattern of user u_i under a specific behavior type k . $W_{i,k}$ is the learned parameter matrix of user u_i . $W_{i,k}$ is the parameter matrix of the learned personalization of user u_i , which injects a specific type of behavioral context into the user u_i representation, and M and Z are transformation parameters. i is the personalized representation of the user u_i that incorporates the contexts.

Given the ID embedding representation E_j of the initialized good v_j , the personalized representation E_j of the good v_j of the fused context is obtained using the same method of learning as above. The specific formula is as follows:

$$\begin{cases} P_{j,k} = \frac{E_j \parallel \sum_{i \in N_j^k} E_i}{\sqrt{|N_i^k| |N_j^k|}} \\ W_{j,k} = M \cdot P_{j,k} + Z \\ \tilde{E}_j = W_{j,k} E_j \end{cases} \quad (2)$$

where $P_{j,k}$ holds information about the users who interact with commodity v_j for a specific type of behavior. $W_{j,k}$ is the learned parameter matrix of the personalization of the commodity v_j , which injects a specific type of behavioral context into the representation of the commodity v_j , and M and Z are transformation parameters. $\tilde{E}_j$ is the personalized representation of the commodity v_j incorporating the contexts.

3.4. Representation of Users and Products Based on Single Behavior

In a multi-behavior recommendation scenario, each interaction has its own features and semantic representation. For example, in an e-commerce commerce platform, users' browsing behavior is more likely to occur than purchasing behavior, and adding to cart and purchasing behavior may occur simultaneously with high probability. Therefore, the proposed module aims to capture personalized behavioral semantic signals. Based on the representation $\tilde{E}_i$ of each user u_i and the representation $\tilde{E}_j$ of each good v_j learned by the embedding module, this module designs a messaging strategy to capture the user–goods interaction graph $G_k = \{V, \varepsilon_k\}$ under a single behavior, where V denotes the set of user and goods nodes, ε_k denotes the set of interaction edges in V , and all the interactions are of type k at this point. The goal of this module is to learn different behavior-specific embedding vectors. The specific formula is as follows:

$$\begin{cases} \bar{E}_{i,k} = \tilde{E}_i + \sigma \left(\sum_{j \in N_i^k} \alpha_{i,j,k} \tilde{E}_j \right) \\ \bar{E}_{j,k} = \tilde{E}_j + \sigma \left(\sum_{i \in N_j^k} \alpha_{i,j,k} \tilde{E}_i \right) \end{cases} \quad (3)$$

where $\bar{E}_{i,k}$ and $\bar{E}_{j,k}$ are the embeddings of user u_i and item v_j at behavior type k . Define $\alpha_{i,j,k}$ as standardization factor $\sqrt{|N_i^k| |N_j^k|}$ where N_i^k denotes the set of goods that user u_i interacts with under k behavior types, and N_j^k denotes the set of users interacting with item v_j under behavior type k .

3.5. Representation of Users and Items Integrated with Multiple Behaviors

In e-commerce platforms, different types of interactions are intertwined, and they are related to each other in a complex way, which is a great challenge for modeling multi-behavioral interaction patterns of users. In order to model the potential relationships between different behavior types, this module designs a multi-behavior relationship learning function, which obtains a more accurate representation of a specific behavior type by injecting information about the interrelationships between different behaviors. The relationship learning function is based on the attention network and is represented as follows:

$$\left\{ \begin{array}{l} \hat{\alpha}_{k,k'}^h = \frac{(Q^h \bar{E}_{i,k})^T (K^h \bar{E}_{i,k'})}{\sqrt{d/H}} \\ \alpha_{k,k'}^h = \text{softmax}(\hat{\alpha}_{k,k'}^h) = \frac{\exp \hat{\alpha}_{k,k'}^h}{\sum_{k'=1}^K \exp \hat{\alpha}_{k,k'}^h} \\ B_{i,k} = ||_{h=1}^H \sum_{k'=1}^K \alpha_{k,k'}^h V^h \cdot \bar{E}_{i,k'} \\ \bar{E}_{i,K+1} = \sum_{k=1}^K B_{i,k} \end{array} \right. \quad (4)$$

The module uses multiple potential spaces ($h \in H$) to perform the embedding projection process, thereby mining the interaction behavior from different hidden dimensions to mine the degree of association between interactions k and k' from different hidden dimensions, where $\bar{E}_{i,K+1}$ denotes the global user representation considering all behavior types. $B_{i,k}$ redefines a particular type of behavioral embedding by connecting feature representations from different learning subspaces, which encodes the degree of influence of other interaction behaviors on the behavior, considering the correlation between interaction behaviors. $\hat{\alpha}_{k,k'}^h$ is the computed correlation between the interaction behavior k and k' is the degree of correlation between the computed interaction behavior k and k' . W^h is the transformation matrix that transforms the vectors into h projection space, which realizes the transformation of Q, K vector dimensions in the attention mechanism.

During the training process, to alleviate overfitting, $\bar{E}_{i,k'}$ is partitioned into H feature vectors of the size d/H dimension, corresponding to the H head, and the multi-head attention mechanism processes these segments in parallel before applying the splicing operation. $\bar{E}_{i,k'}^h$ denotes the h -th slice of $\bar{E}_{i,k'}$.

3.6. User and Item Representations Infused with Higher-Order Neighborhood Information

In order to capture the higher-order complexity of the interaction graph and study the higher-order interactions between user interaction behaviors, this module integrates the vector representation obtained from the behavioral semantic learning module to learn the higher-order embedding propagation paradigm. The higher-order information is injected into the user u_i embedding by the following equation:

$$\bar{E}_{i,k}^{(l+1)} = \left\{ \begin{array}{l} \text{GCN}(\bar{E}_{i,k}^{(l)}), k = 1, 2, \dots, K \\ \text{Att}(\bar{E}_{i,1}^{(l+1)}, \dots, \bar{E}_{i,K+1}^{(l+1)}), k = K + 1 \end{array} \right. \quad (5)$$

The higher-order feature representation of the commodity v_j is processed using the same network as the user representation above, where GCN is the graph convolutional network that defines the behavioral semantic learner. Att denotes the interconnected learning function between behaviors. By L operations, the model learns the connection

relations between nodes for L -hops. To obtain a higher-order information representation, the feature vectors of the $L + 1$ layer network are stitched to obtain the final user and commodity representations.

$$\hat{E}_{*,k} = \bar{E}_k^{(1)} \oplus \bar{E}_k^{(2)} \oplus \cdots \oplus \bar{E}_k^{(L+1)} \quad k = 1, 2, \dots, K + 1 \quad (6)$$

where $*$ denotes the final user embedding when $*$ is i and $*$ is j denotes the final product embedding.

3.7. Target Behavior Prediction

Based on the prediction sub-network learned above, the contextual behavioral information (page view, favorite, add to cart) not only provides useful external knowledge in the target behavior (purchase) prediction phase but also serves as a supervisory signal for model optimization. Based on the above learned feature representations $\hat{E}_{*,k}$ of users and goods under specific behavior types, the prediction network proposed in this model uses non-target behaviors as supervisory signals to obtain personalized meta-knowledge based on the target behavior k' . This process is defined as follows:

$$\begin{cases} P_{i,j}^k = \sigma(W_2 \cdot \oslash(\hat{E}_{i,k}, \hat{E}_{j,k})) \\ D_{i,j}^{k,k'} = \sigma(W_1 \cdot \oslash(P_{i,j}^k, P_{i,j}^{k'})) \end{cases} \quad (7)$$

Of this, $\oslash(v_1, v_2) = v_1 \odot v_2 \parallel v_1 \parallel v_2$, where $\odot$ denotes the multiplication of the corresponding elements of two vectors and $\parallel$ denotes the splicing between the elements. $D_{i,j}^{k,k'}$ encode the meta-knowledge between user u_i and commodity v_j , that is, the dependency between target behavior k' and context behavior k ($k \in (K + 1), k' \in K$). $P_{i,j}^k$ is the projective quantity under the behavior k .

Based on the above learned dependencies between interaction behaviors, the parameters of the prediction network are learned by the following equation.

$$\begin{cases} M_1 = W_1 D_{i,j}^{k,k'} + m_1 \\ p_1 = W_2 D_{i,j}^{k,k'} + m_2 \\ p_2 = W_3 D_{i,j}^{k,k'} + m_3 \end{cases} \quad (8)$$

Ultimately, the model predicts the interaction between user u_i and commodity v_j under target behavior k' , using the feature vector of non-target behavior k as a supervised signal. The specific formula is as follows:

$$\begin{cases} \eta = \sigma(M_1 \cdot \oslash(\hat{E}_{i,k}, \hat{E}_{j,k}) + p_1) \\ X_{i,j,k'}^{(k)} = \eta^\top p_2 \end{cases} \quad (9)$$

where $X_{i,j,k'}^{(k)}$ is the predicted likelihood of user u_i interacting with good v_j under target behavior k' . η is the intermediate feature vector.

3.8. Optimization Strategy

The model is optimized by using each pair of non-target and target behaviors for prediction. For user u_i and target behavior k' , the model samples S positive samples and S negative samples. In the training process, we use the Adam algorithm [23] for optimization, which is defined by the following equation:

$$\mathcal{L} = \sum_{i=1}^N \sum_{k=1}^{K+1} \sum_{k'=1}^K \sum_{s=1}^S \max(0, 1 - \hat{X}_{i,p_s,k'}^{(k)} + \hat{X}_{i,n_s,k'}^{(k)}) + \lambda \|\Theta\|_F^2 \quad (10)$$

where k denotes the non-target behavior, k' denotes the target behavior, and p_s and n_s denote positive and negative samples, respectively.

In the multi-behavior pattern modeling, the model can learn the personalized semantics of specific behaviors and establish the dependency relationship between different types of behaviors, thus effectively improving the accuracy of recommendation. The model adopts lightweight graph convolutional architecture which costs only $O(L \times K \times d \times |\epsilon|)$ across L layers, K behavior types, d latent factors and $|\epsilon|$ edges. The behavior relation learning costs extra $O(L \times K \times d \times (K + d) \times (N + M))$. As $O(d \times |\epsilon|)$ is comparable with $O((K + d) \times (N + M))$ in our case, the complexity does not increase. The prediction network costs $O(S \times d^2)$ computations for each user. In conclusion, our model could achieve comparable time complexity with some graph convolution-based models.

4. Experiments

4.1. Datasets

Taobao, one of the largest e-commerce platforms in China, contains four types of user interactions, namely, page view, add to cart, favorite, and purchase. Each row of the dataset represents a user behavior, consisting of user ID, product ID, product category ID, behavior type, and timestamp, and is separated by commas.

Beibei is one of the largest online retail websites for baby products in China, and it involves three types of user interaction behaviors, including page browsing, adding to cart, and purchasing.

The JDATA dataset is from JD.com, a famous e-commerce website in China, and contains two months of user behavior data from JD.com's website. The types of actions are browse, order, follow, comment, and add to cart.

4.2. Evaluation Metrics

To verify the performance of the proposed model, we employ a variety of evaluation metrics, including the Hit Ratio ($HR@10$) and Normalized Discounted Cumulative Gain ($NDCG@10$).

$$HR@K = \frac{\text{Number of Hits@K}}{GT} \quad (11)$$

where GT is all items in the test set, and the numerator is the sum of the number of items hit in the given Top- k recommendation list.

$$NDCG@K = Z_k \sum_{i=1}^K \frac{2^{r_i} - 1}{\lg(1 + i)} \quad (12)$$

where Z_k is a normalization factor to ensure the presence of a normalized representation with a value of 1 in the list; r_i indicates the predicted relevance of the i th item, represented by 0 and 1; and $\lg(1 + i)$ is the location decay function. The larger the $NDCG$ and HR values of the user to be recommended, the more the recommendation list matches the user's preference and the better the recommendation effect of the algorithm. In order to compare the performance of different models fairly, $NDCG$ used the above calculation method in the experiment. The experimental results obtained are different from those in the references, but the trend of the experimental results is the same.

4.3. Compared Methods and Implementation Details

4.3.1. Recommendation Model Based on Graph Neural Network

ST-GCN [24]: This method is a convolution-based graph neural network model that generates user embeddings through an encoder–decoder coder framework to generate user embeddings.

SR-GNN [25]: A session-based graph neural network model is proposed, which establishes complex dependencies of the session order between interaction items, which is difficult to achieve using previous traditional sequential approaches.

NGCF [12]: This is a message-passing architecture for user commodity interaction graphs on information aggregation, thus exploiting the higher-order relationships in the interaction graph.

4.3.2. Recommendation Models for Multi-Behavioral Categories

NMTR [8]: This approach proposes a new solution for learning recommender systems from user multi-behavior data, and the model considers cascading relationships between different types of behaviors, while cascading predictions for different types of behaviors based on a multi-task learning framework.

MATN [11]: This method preserves cross-type behavioral synergy signals and type-specific behavioral contextual information by explicitly encoding multi-behavioral relational structures. The model transforms each type of behavioral feature through a designed memory unit, generating a specific behavioral representation through this type-specific transformation process.

MBGCN [2]: This approach proposes a multi-behavior graph convolutional network-based model that learns behavior intensity through the user–goods propagation layer and captures behavior semantics through the goods–goods propagation layer, which better addresses the limitations of existing work.

4.4. Experimental Results and Analysis

We evaluate the performance of all baseline methods on different datasets, and the results are shown in Table 1, which summarizes the following observations: The MK-GCN model in this article significantly improves the recommendation performance. This performance gap can be attributed to the effective personalized multi-behavior pattern modeling and the rich context information of user and item representations obtained under the meta-learning paradigm. Most studies ignore the different behavior habits of different users and simply assign different weights to different behaviors. In this paper, we learn user personalized behavior feature representations from interaction graphs according to user behavior habits.

Table 1. Overall performance of the model on the Beibei, Taobao, and JDATA datasets.

	Taobao		Beibei		JDATA	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
SR-GNN	0.321	0.181	0.591	0.326	0.432	0.263
ST-GCN	0.347	0.206	0.609	0.343	0.452	0.285
NGCF	0.302	0.185	0.611	0.375	0.461	0.292
NMTR	0.332	0.179	0.613	0.349	0.481	0.304
MATN	0.354	0.209	0.626	0.385	0.489	0.309
MBGCN	0.369	0.222	0.642	0.376	0.463	0.277
MK-GCN	0.472	0.300	0.683	0.405	0.512	0.316

MK-GCN consistently achieves better performance than the baseline models, but these baseline models have different degrees of limitations. SR-GNN and ST-GCN models do not consider the specific operation behavior of users, and only model and extract features based on the products that users interact with. The NMTR model only models the cascading relationships between multiple types of interaction behaviors and cannot explore the high-order behavior dependencies in the interaction graph. The MATN model aggregates different types of behavior patterns by weighted summation, which cannot comprehensively capture the complex interdependence between different types of interaction behaviors.

MK-GCN consistently obtains better performance than the baseline models, which all have different degrees of the SR-GNN and ST-GCN models and do not consider the specific operational behavior of the user and only model and extract features based on the goods that the user interacts with. The NMTR model only models the cascading relationships between multiple types of interactions and cannot explore the higher-order behavioral

dependencies in the interaction graph. The MATN model aggregates different types of behavioral patterns through weighted summation, which cannot fully capture the complex interdependencies between different types of interactions.

Furthermore, the comparison between MK-GCN and the multi-behavior graph neural model MBGCN demonstrates the proposed method's advantages of multi-behavior dependency modeling. Among the various baseline methods, it can be observed that, compared to other single-row-for-model recommendation methods that do not distinguish between intersection types, the injection of multi-behavior information into the recommendation framework (i.e., NMTR, MATN, MBGCN) into multi-behavior information improves the performance. This result confirms the role of exploring multi-behavioral patterns for recommendation improvement.

4.5. Ablation Experiments

In order to explore the effect of each module in the model, the variant models shown in Table 2 were set up for the experiment. The result of the melting experiments is shown in Table 3. Based on the experimental results, we draw the following conclusions.

Table 2. Description of the model variants.

Model Variants	Notes
-relation	Remove the multi-behavior relational learning function
-metaEncoder	Remove the Meta knowledge encoder
-metaPred	No longer rely on the meta-knowledge encoder to learn the parameters of the prediction layer

Table 3. Results of ablation experiment.

	Taobao		Beibei		JDATA	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
-relation	0.6813	0.4049	0.3878	0.2315	0.4865	0.3138
-metaEncoder	0.6791	0.4046	0.4647	0.2852	0.5026	0.3199
-metaPred	0.6605	0.4036	0.4868	0.2968	0.5302	0.3399
MK-GCN	0.6907	0.4103	0.4906	0.2997	0.5319	0.3447

- (1) Behavioral relational learning plays an active role in capturing higher-order information during message passing in graph neural networks. This suggests that the model uses attention layers under multiple representation subspaces to capture the pairwise correlations between various interaction behavior. It is reasonable that the model uses the attention layer to capture pairwise correlations between various interaction types in multiple representation subspaces.
- (2) The results demonstrate the necessity of learning the parameters of the prediction network using the dependencies between interaction behaviors of the network. This suggests that behavioral relationships can not only provide external knowledge in the process of multi-behavior aggregation but can also serve as a supervisory signal for model optimization.
- (3) MK-GCN outperforms -metaEncoder and -metaPred because they do not incorporate a meta knowledge learner, which indicates the importance of user-specific behavior modeling through the meta-learning paradigm.

5. Conclusions and Future Work

In this paper, a multi-behavior augmented recommendation framework based on graph neural networks is studied and designed to address the heterogeneity and diversity of user interaction behaviors. The model first encodes user and product feature vectors fusing contextual information according to a custom meta-learning paradigm, explores

the dependencies between multiple behavior types by learning the semantic features of different behaviors, and uses graph convolutional networks and attention networks to obtain higher-order association information in the user–commodity interaction graph through multiple operations learning. Finally, the feature vectors of non-target behaviors are used as supervised signals to predict the likelihood of user u interacting with product j using target behavior k . Experimental validation is conducted on three large e-commerce datasets, and the results show that the model performs better compared to other baseline models. The drawback of this model is that it cannot deal with real-time user behavior data stream and can only make recommendations through the collected historical behavior data. Future work hopes to further investigate time-sensitive models that can leverage newly arrived user behavior data to facilitate real-time recommendations.

This model can be widely used in multi-behavior scenarios, such as shopping mall recommendation, music, books, movies, and so on. In a real scenario, we will model the complex relationship as a heterogeneous graph, which contains multiple types of nodes and edges. Then this model simulates the user's behavior pattern by learning the dependencies between different types of behaviors, so as to obtain more accurate recommendation results, which is more conducive to the platform to make wise decisions and adjust in time.

Author Contributions: Conceptualization, R.L.; methodology, R.L. and Y.L.; software, Y.L.; validation, J.L.; resources, R.L. and J.L.; Investigation, Y.L.; data curation, S.Y.; writing—original draft preparation, R.L. and Y.L.; writing—review and editing, R.L., Y.L. and S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: R.L. was employed by the company Guizhou Power Grid Company Limited. Y.L., J.L. and S.Y. were employed by Zhejiang University of Science and Technology. The authors declare that this study received funding from Guizhou Power Grid Co. Ltd. (No. 066700KK52180021) and the National Natural Science Foundation of China (No. 61972357). The funder Guizhou Power Grid Co. Ltd. (No. 066700KK52180021) had the following involvement with the study: Study design, data collection, analysis, interpretation of data, the writing of this article. The funder National Natural Science Foundation of China (No. 61972357) had the following involvement with the study: Interpretation of data, the writing of this article, decision to publish, and preparation of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available upon request.

Acknowledgments: We sincerely thank the editors and the reviewers for their valuable comments in improving this paper. The author is thankful to Liya Huang and Yuan Ji who contributed to this study.

Conflicts of Interest: The authors declare that they have no conflict of interest to report regarding the present study.

References

1. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1441–1450.
2. Jin, B.; Gao, C.; He, X.; Jin, D.; Li, Y. Multi-behavior recommendation with graph convolutional networks. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 25–30 July 2020; pp. 659–668.
3. Singh, A.P.; Gordon, G.J. Relational learning via collective matrix factorization. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 650–658.
4. Pan, W.; Xiang, E.; Liu, N.; Yang, Q. Transfer learning in collaborative filtering for sparsity reduction. In Proceedings of the AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010; Volume 24, pp. 230–235.
5. Xie, R.; Qiu, Z.; Rao, J.; Liu, Y.; Zhang, B.; Lin, L. Internal and Contextual Attention Network for Coldstart Multi-channel Matching in Recommendation. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 11–17 July 2020; pp. 2732–2738.

6. Zhu, Y.; Ge, K.; Zhuang, F.; Xie, R. Transfer-meta framework for cross-domain recommendation to cold-start users. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 11–15 July 2021; pp. 1813–1817.
7. Guo, L.; Hua, L.; Jia, R.; Zhao, B.; Wang, X.; Cui, B. Buying or browsing: Predicting real-time purchasing intent using attention-based deep network with multiple behavior. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 4–8 August 2019; pp. 1984–1992.
8. Gao, C.; He, X.; Gan, D.; Chen, X.; Feng, F.; Li, Y.; Chua, T.; Yao, L.; Song, Y.; Jin, D. Neural multi-task recommendation from multi-behavior data. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 8–11 April 2019; pp. 1554–1557.
9. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 25–30 July 2020; pp. 639–648.
10. Xia, L.; Huang, C.; Xu, Y.; Dai, P.; Zhang, X.; Yang, H.; Pei, J.; Bo, L. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 4486–4493.
11. Xia, L.; Huang, C.; Xu, Y.; Dai, P.; Zhang, B.; Bo, L. Multiplex behavioral relation learning for recommendation via memory augmented transformer network. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 25–30 July 2020; pp. 2397–2406.
12. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.-S. Neural graph collaborative filtering. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 165–174.
13. Kabbur, S.; Ning, X.; Karypis, G. Fism: Factored item similarity models for top-n recommender systems. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, Chicago, IL, USA, 11–13 August 2013; pp. 659–667.
14. Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 426–434.
15. Ning, X.; Karypis, G. Slim: Sparse linear methods for top-n recommender systems. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining, Vancouver, BC, Canada, 11–14 December 2011; pp. 497–506.
16. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
17. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [[CrossRef](#)] [[PubMed](#)]
18. Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W.L.; Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 974–983.
19. Huang, L.; Ma, Y.; Liu, Y.; Du, B.; Wang, S.; Li, D. Position-enhanced and time-aware graph convolutional network for sequential recommendations. *ACM Trans. Inf. Syst.* **2023**, *41*, 1–32. [[CrossRef](#)]
20. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
21. Guan, W.; Song, X.; Zhang, H.; Liu, M.; Yeh, C.-H.; Chang, X. Bi-directional heterogeneous graph hashing towards efficient outfit recommendation. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 268–276.
22. Guan, W.; Jiao, F.; Song, X.; Wen, H.; Yeh, C.-H.; Chang, X. Personalized fashion compatibility modeling via metapath-guided heterogeneous graph learning. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 11–15 July 2022; pp. 482–491.
23. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
24. Zheng, Y.; Gao, C.; He, X.; Li, Y.; Jin, D. Price-aware Recommendation with Graph Convolutional Networks. In Proceedings of 2020 IEEE 36th International Conference on Data Engineering, Dallas, TX, USA, 20–24 April 2020; pp. 133–144.
25. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-based recommendation with graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 346–353.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.