Composition-Enhanced Graph Collaborative Filtering for Multi-behavior Recommendation

Daqing Wu[1,2], Xiao Luo[1,2], Zeyu Ma[3], Chong Chen[1,2], Pengfei Wang[2], Minghua Deng[1], Jinwen Ma[1,*]
[1]*School of Mathematical Sciences, Peking University, Beijing, China*
[2]*Damo Academy, Alibaba Group, Hangzhou, China*
[3]*School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen Graduate School, Shenzhen, China*
{wudq,xiaoluo}@pku.edu.cn, zeyu.ma@stu.hit.edu.cn,
{cheung.cc, wpf228618}@alibaba-inc.com, {dengmh,jwma}@math.pku.edu.cn

***Abstract*—Rapid and accurate prediction of user preferences is the ultimate goal of today's recommender systems. More and more researchers pay attention to multi-behavior recommender systems which utilize the auxiliary types of user-item interaction data, such as page view and add-to-cart to help estimate user preferences. Recently, graph-based methods were proposed to showcase an advanced capability in representation learning and capturing collaborative signals. However, we argue that these methods ignore the intrinsic difference between the two types of nodes in the bipartite graph and aggregate information from neighboring nodes with the same functions. Besides, these models do not fully explore the collaborative signals implied by the meta-path across different types of behavior, which causes a huge loss of the potential semantic information across behaviors. To address the above limitations, we present a unified graph model named SaGCN (short for Semantic-aware Graph Convolutional Networks). Specifically, we construct separate user-user and item-item graphs by meta-path, and apply separate aggregation and transformation functions to propagate user and item information. To perform better semantic propagation, we design a relation composition function and a semantic propagation architecture for heterogeneous collaborative filtering signals learning. Extensive experiments on two real-world datasets show that SaGCN outperforms a wide range of state-of-the-art methods in multi-behavior scenarios.**

***Index Terms*—Multi-behavior Recommendation, Collaborative Filtering, Graph Embedding, Neural Network**

I. Introduction

Recently, recommender systems have been widely used in abundant real-world applications [1]–[3]. The key to recommender systems is modeling user-item interaction while capturing the user preferences and the underlying item characteristics [1], [4]. Collaborative filtering [3] is the most popular paradigm for building a recommender system, assuming that behaviorally similar users would imply similar preferences on items. Although the existing collaborative filtering models have achieved commendable results in recommender systems, they only focus on the singular type of interactions. In real-world scenarios, the interactions are multiplex and revealed with relationship diversity in nature [5]–[7]. For example, in real-world scenarios of E-commerce, users can interact with items provided in multiple manners, such as page view, add-to-cart, purchase and etc. Among various types of behaviors, purchase behavior directly determines the platform's profit. Due to the data sparsity, recommendation models built only towards the purchase behavior can hardly achieve a good performance for a user with few historical purchase records. In other words, recommender systems should have the ability to make use of auxiliary behaviors, to help predict user purchase interaction, which is the multi-behavior recommendation. Therefore, several studies have utilized multiple behavioral data to provide useful collaborative signals to obtain user preferences, which helps to build recommender systems with better performances [6], [8]–[13].

* Corresponding author.
The first two authors contribute equally to this work.

The early efforts to tackle the multi-behavior recommendation are mostly based on collective matrix factorization (CMF) [8], [14]–[16]. To deal with the challenge in modeling the dependencies across multiple behaviors, extensive researches consider the prior behavior correlations between multi-type interactions. For instance, Gao et al. [10] proposed a solution named NMTR which assumes that the behavior types have a total order and sorts them from the lowest level to the highest level. Similarly, efficient heterogeneous collaborative filtering (EHCF) [13] correlates the prediction of multiple behaviors in a transfer way. Recently, GCN-based models leverage the potential semantics to capture the collaborative signals in single behavior user-item graph [4], [17], [18]. For multi-behavior interactions, MBGCN [12] splits multiple types of behavioral interactions graph into several single behavioral interactions subgraph and builds upon a message-passing architecture on each subgraph.

Despite the effectiveness of previous studies, we perceive three important limitations. *First*, **the intrinsic difference between the two types of nodes in the bipartite graph is neglected**. For example, MBGCN [12] merely aggregates information from neighboring nodes in the same way during the embedding construction procedure. However, there is an important intrinsic difference between users and items in a real environment. This suggests that the aggregation or transformation functions should be dependent on the type of entity. *Second*, **the existing methods do not make full use of the semantic signals which is available from user-user and item-item graphs**. Although 2-hop neighborhoods in the bipartite graph can capture these to some extent, modeling user-user and item-item semantic relations directly can decrease the dis-

978-1-6654-2398-4/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDM51629.2021.00183

turbance of mediator. *Third*, **the collaborative signals across different types of behaviors are not sufficiently explored**.In the multi-behavior user-item interaction graph, two users/items are not only connected through the homogeneous behavior interactions but also the heterogeneous behavior interactions. The heterogeneous behavior interactions provide the additional potential semantics to capture the collaborative signals across different types of behavior, and further improve predicting effectiveness.

To address the aforementioned limitations, we present a unified graph model named SaGCN (short for **S**emantic-**a**ware **G**raph **C**onvolutional **N**etworks) which jointly learns desirable user/item representations integrating the collaborative signals across behaviors. We first select meaningful meta-paths $U \xrightarrow{r_k} V \xrightarrow{r'_k} U$ (k may equal k') to built user-user graph. Similarly, meta-paths $V \xrightarrow{r_k} U \xrightarrow{r'_k} V$ are adopted to form item-item graph. We design a relation composition function to encode the relation embeddings along the meta-path by the cascade manner. Last, in order to capture the effective collaborative signals, we design a semantic propagation architecture that first constructs relation-specific messages and then aggregates them by the attention mechanism. The contributions of our work can be summarized as follows: 1. Modeling the intrinsic differences between users and items: we construct separate user-user and item-item graphs by meta-path, and apply separate aggregation and transformation functions to propagate user information and item information. 2. Capturing the collaborative signals across different types of behaviors: we design a relation composition function and a semantic propagation architecture to enable semantics propagation effectively. 3. We conduct extensive experiments to evaluate the performance of our proposed model on two real-world E-commerce datasets. The results demonstrate the superiority of our proposed method over several state-of-the-art methods.

II. Related Work

A. GCN based Collaborative Filtering

GCN-based methods have been widely explored to model collaborative filtering [4], [17]–[19]. NGCF [17] proposed a embedding propagation layer to model high-order connectivity in the user-item interaction graph. BGNN [19] considered the interactions between neighbor nodes by a devised bilinear aggregator. LightGCN [4] learned user and item embeddings with light graph convolution and layer combination to simplify the design of GCN. Sun et al. [20] built user-user and item-item graphs for capturing collaborative filtering signals.

B. Multi-behavior Recommendation

The well-known early model CMF [14] simultaneously factorized multiple user-item interactions with sharing item-side embeddings across matrices and extended to leveraging multiple user behaviors for recommender systems [8]. NMTR [10] accounted for the cascading relationship among different types of behaviors and perform a joint optimization based on the multi-task learning framework. MATN [11] developed a multi-behavior dependency encoder with a transformer architecture and augmented the multi-behavior transformer network with a memory attention mechanism to model behavioral context and behavior inter-dependencies. EHCF [13] considered the translation on the behavior sequence and learned model parameters from positive-only data. MBGCN [12] built upon a message-passing architecture between user and item to learn behavior strength and designed item-to-item embedding propagation to model item-to-item similarity. Distinct from previous studies, we simultaneously conduct user-user and item-item semantic propagation across behaviors to improve the predicting effectiveness.

III. Methodology

In this section, we first give the definitions of the multi-behavior recommendation. Then, we propose SaGCN, a unified heterogeneous GCN model to learn desired user and item representations in handy and neat way, which is illustrated in Fig. 1.

A. Problem Formulation

Suppose the dataset contains users U and items V, $\boldsymbol{Y}^{(1)}$, $\boldsymbol{Y}^{(2)}$, $\cdots$, $\boldsymbol{Y}^{(K)}$ denote the user-item interaction matrices of size $|U| \times |V|$ for all K types of behaviors, in which $\boldsymbol{Y}^{(k)}$ with each entry having value 1 or 0 comes from users' implicit feedbacks of the k-th behavior:

$$y_{uv}^{(k)} = \begin{cases} 1, & \text{if user } u \text{ has observed interaction with item } v \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The task is estimating the likelihood $\hat{y}_{uv}^{(K)}$ that user u will interact with item v under the K-th behavior. The items (unobserved under the target behavior) are ranked in the descending order of the likelihood, which provides a Top-N item recommendation list for each user.

B. Embedding Layer

For each type of behavior, user embedding vectors and item embedding vectors can be expressed by embedding matrix $\boldsymbol{P} \in \mathbb{R}^{|U| \times d}$ and $\boldsymbol{Q} \in \mathbb{R}^{|V| \times d}$ in d-dimension Euclidean space:

$$\boldsymbol{e}_u = \boldsymbol{P}^T \cdot \mathrm{ID}_u, \quad \boldsymbol{e}_v = \boldsymbol{Q}^T \cdot \mathrm{ID}_v \tag{2}$$

where $\mathrm{ID}_u \in \{0,1\}^{|U|}$ and $\mathrm{ID}_v \in \{0,1\}^{|V|}$ denote the one-hot feature vector for user u and item v. Similar to user and item embedding, we take considerations on edge heterogeneity and embed each type of behavior (i.e., edge type or relation) into a d-dimension vector,

$$\boldsymbol{e}_r = \boldsymbol{R}^T \cdot \mathrm{ID}_r \tag{3}$$

where $\boldsymbol{R} \in \mathbb{R}^{K \times d}$ is a embedding matrix and $\mathrm{ID}_r \in \{0,1\}^K$ denotes the one-hot feature vector for r-th behavior. To alleviate the effect of different scales, we normalize the user embeddings, item embeddings and behavior embeddings. For the sake of neat notation, we denote $\boldsymbol{e}_u^{(0)} = \boldsymbol{e}_u$ and $\boldsymbol{e}_v^{(0)} = \boldsymbol{e}_v$.

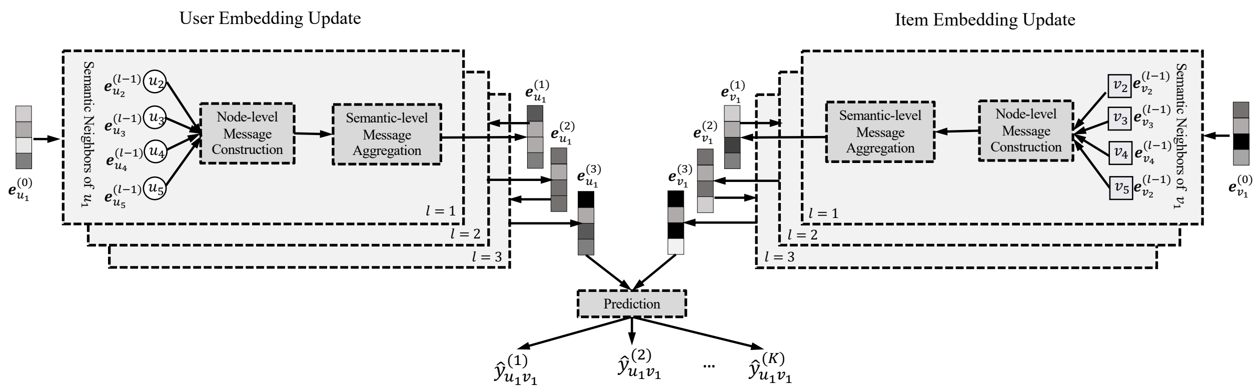

Fig. 1: The overall architecture of SaGCN, where node u_1 is the target user and node v_1 is the target item.

C. Semantic Propagation Layer

In the user-user graph or item-item graph, each semantic relation transits relation-specific information from the source node to the target node. We formulate the semantic propagation with two stages: node-level message construction and semantic-level message aggregation. Without loss of generality, we only narrate user semantic propagation, and item embedding has the same propagation style.

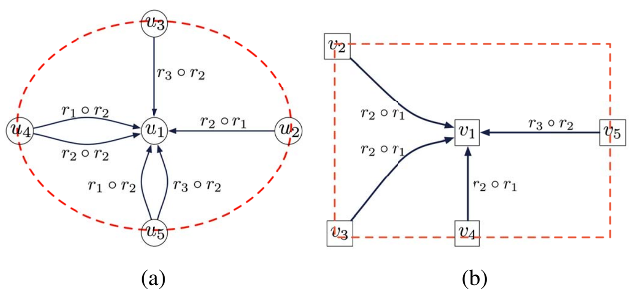

Fig. 2: An illustration of semantic neighbors, where black arrow represents semantic propagation. (a) The semantic neighbors (red dashed circle) of user u_1 are $\{u_2, u_3, u_4, u_5\}$. (b) The semantic neighbors (red dashed rectangle) of item v_1 are $\{v_2, v_3, v_4, v_5\}$.

Semantic Neighbors. Given the multi-behavior user-item interaction graph $\mathcal{G} = (U, V, \mathcal{E}, \mathcal{R})$ and the pre-defined meta-path set $\{U \xrightarrow{r_k} V \xrightarrow{r_{k'}} U\}_{k,k'=1}^{K}$, $\{V \xrightarrow{r_k} U \xrightarrow{r_{k'}} V\}_{k,k'=1}^{K}$, there exists a set of meta-path based neighbors of each node which can reveal diverse structure information and rich semantics. Denote $U \xrightarrow{r_k} V \xrightarrow{r_{k'}} U$ as $\Phi_{k,k'}$ and $V \xrightarrow{r_k} U \xrightarrow{r_{k'}} V$ as $\Psi_{k,k'}$. The meta-path based neighbors $\mathcal{N}_u^{\Phi}$ of user node u are defined as the set of nodes which connect with node u via meta-path Φ. Further, we denote $\mathcal{N}_u = \bigcup_{c \in R_u} \mathcal{N}_u^{\Phi_c}$ as the semantic neighbors of user node u, where $R_u \subseteq \{r_k \circ r_{k'}\}_{k,k'=1}^{K}$ is the semantic relations set of node u. As for item node, semantic neighbors have the same definition. Based on the predefined meta-path set, we construct separate user-user graph and item-item graph.

Relation Composition. To model implicit semantic relation, we encode $\{\boldsymbol{e}_{r_k}, \boldsymbol{e}_{r_{k'}}\}$ into the composite relation embedding $\boldsymbol{e}_c$ according to

$$\boldsymbol{e}_c = \varphi(\boldsymbol{e}_{r_k}, \boldsymbol{e}_{r_{k'}}) = (\boldsymbol{W}_{comp}\boldsymbol{e}_{r_k}) \odot \boldsymbol{e}_{r_{k'}} \tag{4}$$

where $\varphi(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is relation embedding composition function, $\boldsymbol{W}_{comp} \in \mathbb{R}^{d \times d}$ is the composition parameter matrix which endows the composition with asymmetry, $\odot$ denotes element-wise product in feature space.

Node-level Message Construction. Different semantic relations propagate different semantic information from the source to target node even if the source node is the same. To enable composite relation embedding $\boldsymbol{e}_c$ to learn the semantic relevance, we introduce distance measurement $\mathcal{S}$ between target node u and source node u' according to

$$\mathcal{S}(\boldsymbol{e}_u^{(l)}, \boldsymbol{e}_{u'}^{(l-1)}, \boldsymbol{e}_c) = (\boldsymbol{e}_u^{(l)} - \boldsymbol{e}_{u'}^{(l-1)})^T \mathrm{diag}(\boldsymbol{e}_c)(\boldsymbol{e}_u^{(l)} - \boldsymbol{e}_{u'}^{(l-1)}) \tag{5}$$

According to the fact that GCN is actually a special form of Laplacian smoothing [21], the smoothness of target nodes and neighbor nodes aims to minimize the sum of the distance, i.e.,

$$\begin{aligned} \min_{\boldsymbol{e}_u^{(l)}} \quad & \frac{1}{|\mathcal{N}_u^{\Phi_c}|} \sum_{u' \in \mathcal{N}_u^{\Phi_c}} \mathcal{S}(\boldsymbol{e}_u^{(l)}, \boldsymbol{e}_{u'}^{(l-1)}, \boldsymbol{e}_c) \\ \text{s.t.} \quad & \boldsymbol{e}_u^{(l)} \mathrm{diag}(\boldsymbol{e}_c) \boldsymbol{e}_u^{(l)} = \sum_{i=1}^{d} \left(\boldsymbol{e}_u^{(l)}(i)\right)^2 \boldsymbol{e}_c(i) = 1 \end{aligned} \tag{6}$$

The objective of the above optimization problem aims to find the optimal $\boldsymbol{e}_u^{(l)}$ closed to all c-relation neighbors $\mathcal{N}_u^{\Phi_c}$. The constrained condition guarantees that $\boldsymbol{e}_u^{(l)}$ is located in a unit $\mathcal{S}$ measurement surface related with $\boldsymbol{e}_c$. It's worth noting that the positive term in $\boldsymbol{e}_c$ encourages the the smoothness of u and u' which indicates that u and u' have strong semantic relevance, vice versa. Moreover, the above constrained optimization problem is equivalent to

$$\max_{\boldsymbol{e}_u^{(l)}} \frac{1}{|\mathcal{N}_u^{\Phi_c}|} \sum_{u' \in \mathcal{N}_u^{\Phi_c}} \boldsymbol{e}_u^{(l)T} \cdot (\boldsymbol{e}_{u'}^{(l-1)} \odot \boldsymbol{e}_c) \tag{7}$$

From the perspective of optimization, the gradient of target node embedding in the above optimization problem is

$$\begin{aligned}\nabla \boldsymbol{e}_u^{(l)} &= \frac{\partial \frac{1}{|\mathcal{N}_u^{\Phi_c}|} \sum_{u' \in \mathcal{N}_u^{\Phi_c}} \boldsymbol{e}_u^{(l)T} \cdot (\boldsymbol{e}_{u'}^{(l-1)} \odot \boldsymbol{e}_c)}{\partial \boldsymbol{e}_u^{(l)}} \\ &= \frac{1}{|\mathcal{N}_u^{\Phi_c}|} \sum_{u' \in \mathcal{N}_{uc}} \boldsymbol{e}_{u'}^{(l-1)} \odot \boldsymbol{e}_c \end{aligned} \tag{8}$$

As a result, we construct relation-specific message embedding of target node u according to

$$\boldsymbol{m}_{uc}^{(l)} = \frac{1}{|\mathcal{N}_u^{\Phi_c}|} \sum_{u' \in \mathcal{N}_u^{\Phi_c}} \boldsymbol{e}_{u'}^{(l-1)} \odot \boldsymbol{e}_c \tag{9}$$

where $\boldsymbol{m}_{uc}^{(l)}$ is the message embedding of node u via composite relation c in the l-th semantic propagation layer. Element-wise product operation $\odot$ between $\boldsymbol{e}_{u'}^{(l-1)}$ and $\boldsymbol{e}_c$ enables z-th element of $\boldsymbol{e}_c$ be regarded as the scale of transformation on z-th feature dimension of $\boldsymbol{e}_{u'}^{(l-1)}$.

Semantic-level Message Aggregation. Due to the heterogeneity of semantic relations, we aggregate relation-specific message to update target node embedding. Inspired by the architecture design of Transformer [22], we map $\boldsymbol{e}_u^{(l)}$ into query vector, and $\boldsymbol{m}_{uc}^{(l)}$ into key vector, and learn the importance of each kind of semantic information for node u. Specifically, for the h-th head, we define query, key and value transformation matrices $\boldsymbol{Q}^h \in \mathbb{R}^{\frac{d}{H} \times d}$, $\boldsymbol{K}^h \in \mathbb{R}^{\frac{d}{H} \times d}$ and $\boldsymbol{V}^h \in \mathbb{R}^{\frac{d}{H} \times d}$, where H is the number of heads. The weight α_{uc}^h is determined by the dot-product of the query with all the keys as follows:

$$\alpha_{uc}^h = \frac{(\boldsymbol{Q}^h \cdot \boldsymbol{e}_u^{(l-1)})^T (\boldsymbol{K}^h \cdot \boldsymbol{m}_{uc}^{(l)})}{\sqrt{\frac{d}{H}}}, \quad \hat{\alpha}_{uc}^h = \frac{\exp \alpha_{uc}^h}{\sum_{c' \in R_u} \exp \alpha_{uc'}^h} \tag{10}$$

Based on the learned head-specific attention weights, we aggregate all semantic message with the following multi-head learning operations:

$$\boldsymbol{m}_u^{(l)} = \Big\|_{h=1}^{H} \sum_{c \in R_u} \hat{\alpha}_{uc}^h \boldsymbol{V}^h \cdot \boldsymbol{m}_{uc}^{(l)} \tag{11}$$

where $\|$ is the concatenation operator. Finally, user node u embedding is updated with the following expression

$$\boldsymbol{e}_u^{(l)} = \boldsymbol{e}_u^{(l-1)} + \boldsymbol{m}_u^{(l)} \tag{12}$$

Similarly, item node embedding is updated with the same propagation style, but the learning parameters aren't shared with user semantic propagation.

D. Prediction Layer

The representations obtained from different layers emphasize the information passed from different hops from user-user graph and item-item graph. Thus, we combine them to get the final representations:

$$\mathbf{e}_u^* = \sum_{l=0}^{L} \frac{1}{L+1} \mathbf{e}_u^{(l)}, \qquad \mathbf{e}_v^* = \sum_{l=0}^{L} \frac{1}{L+1} \mathbf{e}_v^{(l)} \tag{13}$$

Then, we incorporated different the learnt representation of each behavior as a separated prediction layer in GMF framework [2] to calculate the likelihood of users' multiple behaviours on items as follows:

$$\hat{y}_{uv}^{(k)} = \sum_{i=1}^{d} \boldsymbol{e}_u^*(i) \boldsymbol{h}^{(k)}(i) \boldsymbol{e}_v^*(i) \tag{14}$$

where $\boldsymbol{h}^{(k)} = \boldsymbol{W}_h \boldsymbol{e}_{r_k}$ is prediction vector for the k-th behavior.

E. Model Training

To learn model parameters, we utilize a weighted regression with squared loss and adapt a uniform weight w for missing entry [23]. The loss is calculated as follows:

$$\mathcal{L}_r^{(k)} = \sum_{u \in U} \Big(\sum_{v \in V_u^{(k)+}} (1 - \hat{y}_{uv}^{(k)})^2 + w \sum_{V_u^{(k)-}} \hat{y}_{uv}^{(k)2} \Big) \tag{15}$$

where $V_u^{(k)+}$, $V_u^{(k)-}$ are positive and negative items for user u under the k-th behavior. In order to better learn parameters from all the heterogeneous data, we obtain the multi-task learning [24] loss function to be minimized as

$$\mathcal{L} = \sum_{k=1}^{K} \lambda_k \mathcal{L}_r^{(k)} \tag{16}$$

where λ_k is included to control the influence of the k-th type of behavior on the joint training.

IV. Experiments

A. Datasets, Baselines and Evaluation Metrics

We experiment on two real-world E-commerce datasets Beibei and Taobao. The statistical details of datasets are summarized in Table I. We compare SaGCN with several state-of-the-art methods including CMF [8], MC-BPR [9], NMTR [10], MATN [11], EHCF [13], MBGCN [12]. We apply the widely used leave-one-out technique [10], [13], and then Hit Ratio (HR) [25] and Normalized Discounted Cumulative Gain (NDCG) [26] are adopted to evaluate the performance of each model.

TABLE I: Statistical details of the evaluation datasets.

Dataset	#User	#Item	#View	#Cart	#Purchase
Beibei	21,716	7,977	2,412,586	642,622	304576
Taobao	48,749	39,493	1,548,126	193,747	2259747

B. Parameter Settings

The parameters for all the baseline methods are initialized as in the corresponding papers. For our SaGCN, we apply a grid search for the optimal parameters and evaluate the model on test data. After tuning, the training batch size is set to 256, and the size of the latent factor dimension is set to 64. The learning rate is fixed as 0.05. The weight for negative entry is set to 0.1 for Beibei, $w = 0.01$ for Taobao. We set $[\lambda_1, \lambda_2, \lambda_3] = [1/6, 4/6, 1/6]$ for both datasets. In the semantic-level message aggregation module, we set the number of attention heads as 2. We try several layers for our SaGCN and find $L = 1$ to be good enough for the two datasets here similar to MBGCN.

TABLE II: Top$-N$ Recommendation Performance Comparisons on Beibei and Taobao. All our results have the statistical significance for $p < 0.01$ compared to the best baseline.

Models	***Beibei***							
	HR@10	HR@50	HR@100	HR@200	NDCG@10	NDCG@50	NDCG@100	NDCG@200
CMF	0.0420	0.1582	0.2843	0.4880	0.0251	0.0462	0.0661	0.0952
MC-BPR	0.0504	0.1143	0.2755	0.3822	0.0540	0.0503	0.0653	0.0996
NMTR	0.0524	0.2047	0.3189	0.4735	0.0285	0.0609	0.0764	0.0968
MATN	0.1157	0.2892	0.3998	0.5296	0.0622	0.0997	0.1176	0.1458
MBGCN	0.1564	0.3434	0.4262	0.5304	0.0828	0.1282	0.1384	0.1474
EHCF	0.1523	0.3316	0.4312	0.5460	0.0817	0.1213	0.1374	0.1535
SaGCN	**0.1700**	**0.3633**	**0.4649**	**0.5693**	**0.0898**	**0.1323**	**0.1487**	**0.1633**
Models	***Taobao***							
	HR@10	HR@50	HR@100	HR@200	NDCG@10	NDCG@50	NDCG@100	NDCG@200
CMF	0.0483	0.0774	0.1185	0.1563	0.0252	0.0293	0.0357	0.0379
MCBPR	0.0547	0.0091	0.1264	0.1599	0.0223	0.0227	0.0366	0.0997
NMTR	0.0585	0.0942	0.1368	0.1868	0.0278	0.0334	0.0394	0.0537
MATN	0.0691	0.1487	0.2149	0.2736	0.0381	0.0564	0.0652	0.0742
MBGCN	0.0701	0.1522	0.2169	0.2823	0.0390	0.0571	0.0653	0.0751
EHCF	0.0717	0.1618	0.2211	0.2921	0.0403	0.0594	0.0690	0.0789
SaGCN	**0.0768**	**0.1758**	**0.2392**	**0.3184**	**0.0428**	**0.0643**	**0.0746**	**0.0856**

TABLE III: Ablation Study of Auxiliary Data and Novel Designs on Beibei.

	HR@10	HR@50	HR@100	HR@200	NDCG@10	NDCG@50	NDCG@100	NDCG@200
SaGCN-V	0.1690	0.3426	0.4366	0.5449	0.0890	0.1307	0.1460	0.1611
SaGCN-C	0.0528	0.1824	0.3020	0.4576	0.0267	0.0540	0.0733	0.0950
SaGCN-H	0.1522	0.3273	0.4317	0.5465	0.0827	0.1209	0.1378	0.1538
SaGCN-A	0.1457	0.2983	0.3963	0.5186	0.0817	0.1148	0.1306	0.1477
SaGCN	**0.1700**	**0.3633**	**0.4649**	**0.5693**	**0.0898**	**0.1323**	**0.1487**	**0.1633**

C. Performance Comparison

The results of the comparison of different methods on both two datasets are shown in Table II. We have the following observations:

- Neural network-based methods perform better than traditional methods (CMF, MC-BPR), which demonstrates that the strong representation learning ability of deep learning helps to improve the performance of recommendation.
- GCN-based methods (MBGCN, SaGCN) outperform other methods generally, which indicates that GCN is more capable of encoding the collaborative signals through embedding propagation.
- Our proposed method SaGCN consistently outperforms other competitive methods in terms of all metrics on both datasets. The average HR and NDCG improvement of our model over the best baseline method are respectively 8.32% and 8.40% on Beibei, 8.24% and 7.77% on Taobao, which justifies the superiority of SaGCN which sufficiently explores the collaborative signals across different types of behaviors.

D. Ablation Study

To understand the effectiveness of the special designs and multiple auxiliary behaviors, we conduct experiments with several variants of SaGCN. Particularly, we introduce the following model variants: SaGCN-C uses view data as the only auxiliary behavior data. SaGCN-V uses cart data as the only auxiliary behavior data. SaGCN-H removes the heterogeneous composite relations. SaGCN-A replaces the attention mechanism with simple average operation. The results on Beibei are recorded in Table III, and the results on Taobao are similar. We summary the following findings: 1. SaGCN using all types of interaction behaviors consistently outperforms SaGCN-V and SaGCN-C under all settings, which validates that integrating multi-behavior relations can improve purchase forecasting. 2. From the comparison of SaGCN-A with SaGCN, our model is able to investigate the semantics of heterogeneous composite relations. 3. From the comparison of SaGCN-H with SaGCN, we find that the performances of SaGCN-H are inferior, verifying that different composite relations propagate different strengths of semantics. By integrating attention operation, the aggregation contains the consistent semantic information with the target node, which benefits from distilling the effective collaborative signals.

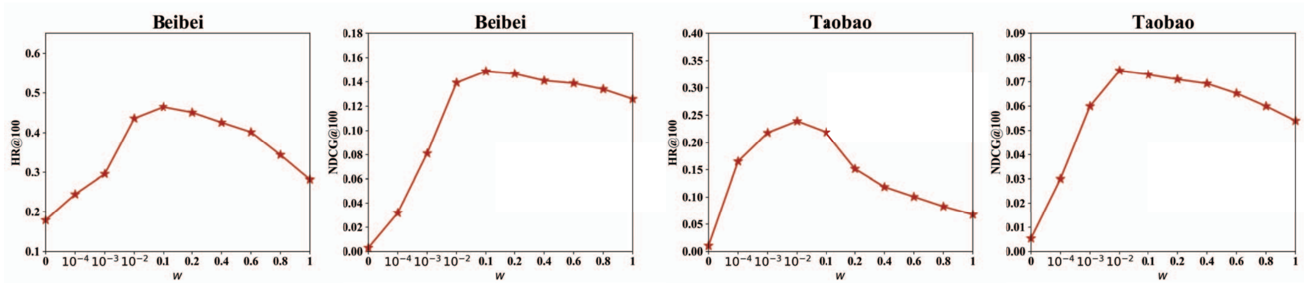

Fig. 3: Performance of SaGCN on Beibei and Taobao w.r.t. different weight for negative entry w.

E. Parameter Sensitivity

We evaluate weight for negative entry w in $[0, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ and the result of HR@100 and NDCG@100 is presented in Figure 3. We can see that when w increases from 0 to 1, the performance of our model first increases and then decreases. The best performance is reached at $w = 0.1$ for Beibei and $w = 0.01$ for Taobao, respectively, which shows that the appropriate weight for negative entry is specific for different datasets. The potential reason is that sparsity is various for different datasets.

V. Conclusion

In this paper, we propose SaGCN for recommendation with heterogeneous user feedbacks. SaGCN has two key characteristics: 1) SaGCN embeds the composite behaviors into vectors to represent heterogeneous semantic relations. 2) SaGCN encodes the effective collaborative signals across different types of behaviors into user/item representations. Extensive experiments on two real-world datasets show that SaGCN outperforms the state-of-the-art recommendation models. This work opens up a new avenue of research by introducing composite behavior embedding for the multi-behavior recommendation. Future work includes exploring our model in complex situations such as cold start problems and knowledge-based recommendation problems. We will also try to extend our method to make it applicable in other recommendation tasks such as sequence-based recommendation tasks.

VI. Acknowledgements

This work was supported by the National Key Research and Development Program of China under grant 2018AAA0100205, and Alibaba Group through Alibaba Research Intern Program.

References

[1] C. Liu, T. Lu, X. Wang, Z. Cheng, J. Sun, and S. C. Hoi, "Compositional coding for collaborative filtering," in *SIGIR*, 2019, pp. 145–154.
[2] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*, 2017, pp. 173–182.
[3] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *ICDM*, 2008, pp. 263–272.
[4] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: simplifying and powering graph convolution network for recommendation," in *SIGIR*, 2020, pp. 639–648.
[5] M. Zhou, Z. Ding, J. Tang, and D. Yin, "Micro behaviors: A new perspective in e-commerce recommender systems," in *WSDM*, 2018, pp. 727–735.
[6] D. Wu, X. Luo, Z. Ma, C. Chen, M. Deng, and J. Ma, "Argo: Modeling heterogeneity in e-commerce recommendation," in *IJCNN*, 2021, pp. 1–8.
[7] X. Luo, D. Wu, C. Chen, J. Ma, M. Deng, C. Shen, J. Huang, and X.-S. Hua, "Criterion-based heterogeneous collaborative filtering for multi-behavior implicit recommendation," *arXiv preprint arXiv:2105.11876*, 2021.
[8] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi, "Improving user topic interest profiles by behavior factorization," in *WWW*, 2015, pp. 1406–1416.
[9] B. Loni, R. Pagano, M. Larson, and A. Hanjalic, "Bayesian personalized ranking with multi-channel user feedback," in *RecSys*, 2016, pp. 361–364.
[10] C. Gao, X. He, D. Gan, X. Chen, F. Feng, Y. Li, T.-S. Chua, and D. Jin, "Neural multi-task recommendation from multi-behavior data," in *ICDE*, 2019, pp. 1554–1557.
[11] L. Xia, C. Huang, Y. Xu, P. Dai, B. Zhang, and L. Bo, "Multiplex behavioral relation learning for recommendation via memory augmented transformer network," in *SIGIR*, 2020, pp. 2397–2406.
[12] B. Jin, C. Gao, X. He, D. Jin, and Y. Li, "Multi-behavior recommendation with graph convolutional networks," in *SIGIR*, 2020, pp. 659–668.
[13] C. Chen, M. Zhang, Y. Zhang, W. Ma, Y. Liu, and S. Ma, "Efficient heterogeneous collaborative filtering without negative sampling for recommendation," in *AAAI*, 2020, pp. 19–26.
[14] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *KDD*, 2008, pp. 650–658.
[15] C. Park, D. Kim, J. Oh, and H. Yu, "Do" also-viewed" products help user rating prediction?" in *WWW*, 2017, pp. 1113–1122.
[16] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using bayesian personalized ranking for social network data," in *WSDM*, 2012, pp. 173–182.
[17] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *SIGIR*, 2019, pp. 165–174.
[18] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T.-S. Chua, "Disentangled graph collaborative filtering," in *SIGIR*, 2020, pp. 1001–1010.
[19] H. Zhu, F. Feng, X. He, X. Wang, Y. Li, K. Zheng, and Y. Zhang, "Bilinear graph neural network with neighbor interactions," in *IJCAI*, 2020, pp. 1452–1458.
[20] J. Sun, Y. Zhang, C. Ma, M. Coates, H. Guo, R. Tang, and X. He, "Multi-graph convolution collaborative filtering," in *ICDM*, 2019, pp. 1306–1311.
[21] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *AAAI*, 2018.
[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.
[23] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *ICDM*, 2008, pp. 502–511.
[24] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *NeurIPS*, 2007, pp. 41–48.
[25] G. Karypis, "Evaluation of item-based top-n recommendation algorithms," in *CIKM*, 2001, pp. 247–254.
[26] K. Järvelin and J. Kekäläinen, "Ir evaluation methods for retrieving highly relevant documents," in *SIGIR*, 2000, pp. 41–48.