國立臺灣大學電機資訊學院暨中央研究院
資料科學學位學程
碩士論文
Data Science Degree Program
College of Electrical Engineering and Computer Science
National Taiwan University and Academia Sinica
Master Thesis

使用個性化多互動偏好排名的多行為推薦系統
Personalized Multi-interaction Preference Ranking for Multi-behavior Recommendation

吳偉樂
Wei Luok Ngu

指導教授：張智星博士、王釧茹博士
Advisor: Jyh-Shing Jang Ph.D., Chuan-Ju Wang Ph.D.

中華民國 111 年 6 月
June, 2022

doi:10.6342/NTU202203790

國立臺灣大學碩士學位論文
口試委員會審定書

使用個性化多互動偏好排名的多行為推薦系統

Personalized Multi-interaction Preference Ranking for Multi-behavior Recommendation

本論文係 吳偉樂（學號 R09946023）在國立臺灣大學資料科學學學位學程研究所完成之碩士學位論文，於民國 111 年 06 月 08 日承下列考試委員審查通過及口試及格，特此證明

口試委員：張智星（簽名）

（指導教授）

[illegible]

黃瀚萱　蔡銘峰

學程主任：黃俊郎（簽名）

doi:10.6342/NTU202203790

誌謝

研究所的兩年很快就過去了，首先想感謝我的兩位指導老師，張智星老師以及王釧茹老師，除了在我研究生活中給予許多教導、幫助以及建議外，還提供我研究經費，讓我能安心的學習。此外，也感謝台大 MIR 實驗室以及中研院 CFDA & CLIP 實驗室的同學們，很開心能與大家一起上課、做研究以及聊八卦，讓我的碩士生活不在孤單。

不知不覺已經在台灣六年了，台灣是一個充滿人情味及溫暖的地方，去過了許多地方，認識了許多友善的人，發生了許多有趣的故事。不管以後我在哪裡，我都會記得這些很棒的回憶。

感謝我的家人，感謝你們無條件的支持及鼓勵，研究所期間因疫情無法回家，沒想到再次回家的時候已經畢業了，期待帶著論文和畢業證書回家一起拍照。

想說的話有很多，但我無法很好地把自己想表達的化作文字，總之最後希望大家都能身體健康，繼續朝著自己的目標前進。

doi:10.6342/NTU202203790

摘要

多行為推薦的目標是利用用戶及物品的多交互關係例如購買和加入購物車來進行建模以解決推薦中常見的資料稀疏及冷啟動問題。雖然最近一些基於多行為的推薦演算法成功地利用不同種類的用戶及物品交互行為來提升推薦效果，但這些方法還存在一些限制。第一，大多數開創性的工作將單一行為當作目標行為且只根據目標行為來優化模型；然而，這需要重新訓練模型以預測其它行為，因此對於大規模資料集和許多實際應用來說效率很低。第二，雖然近期有些研究透過結合所有種類的行為一起進行模型優化以解決上述問題，但模型學習到的行為向量是所有用戶及物品都共用的，這樣的設定非常粗糙且不足以補抓用戶在不同行為下的偏好。除此之外，雖然這些最先進的多行為推薦演算法看似能對針對不同的行為對用戶推薦商品，但其它行為的預測並沒有被明確地評估在相關論文。我們透過使用個性化多互動偏好排名（PMiPR）來解決這些限制，它是應用於多行為推薦中有效及高效向量學習框架。具體來說，PMiPR 透過學習用戶及物品在每種行為下的特定行為向量將多行為信息整合至建模過程中。這不僅以更細粒度的方式對多行為信息建模，也讓我們能透過利用為指定的用戶及物品的行為向量來對不同的行為進行推薦。在四個公開的基準資料集上進行的綜合實驗證明了 PMiPR 在多行為推薦的有效性及效能。

關鍵字：協同過濾、多行為推薦、偏好排序

doi:10.6342/NTU202203790

Abstract

The goal of multi-behavior recommendation is to leverage user-item interactions such as *purchase* and *add-to-cart* into the modeling process to address the commonly-faced data sparsity or cold start issues in recommendation. Although some recent multi-behavior-based recommendation algorithms successfully leverage different types of user-item interactions to improve recommendation performance, these methods still have limitations. First, most pioneering works treat a single behavior as the target behavior and optimize the model based on the target behavior only; this however necessitates retraining of the model to predict other behaviors and is thus inefficient for large-scale datasets and many real-world applications. Second, although recent studies address this issue by jointly optimizing the model based on all types of behaviors, the learned behavior embeddings are shared across all users and items, which is coarse-grained and insufficient to capture user preferences under different behaviors. Moreover, although such state-of-the-art multi-behavior recommendation algorithms seem able to recommend items for users w.r.t. different behaviors, they do not explicitly evaluate their methods in the reported experiments. We address these limitations with personalized multi-interaction preference ranking (PMiPR), an effective and efficient embedding learning framework for multi-behavior recommendation. Specifi-

doi:10.6342/NTU202203790

cally, the proposed PMiPR incorporates multi-behavioral information into the modeling process by learning user-specific and item-specific behavior embeddings for each type of behavior. This not only models multi-behavioral information in a more fine-grained way but enables us to make recommendations w.r.t. different behaviors by leveraging the designated behavior embeddings for users and items. Comprehensive experiments on four public benchmark datasets demonstrate the effectiveness and efficiency of PMiPR for multi-behavior recommendation.

Keywords: Collaborative filtering, Multi-Behavior Recommendation, Preference ranking

doi:10.6342/NTU202203790

Contents

doi:10.6342/NTU202203790

doi:10.6342/NTU202203790

List of Figures

doi:10.6342/NTU202203790

doi:10.6342/NTU202203790

List of Tables

doi:10.6342/NTU202203790

Chapter 1

Introduction

In this chapter, we discuss the motivation and contributions of this research by illustrating the shortcomings of different recommendation methods in Section 1.1, and provide an overview of each chapter in Section 1.2.

1.1 Research Motivation

With the rapid expansion of Internet services, recommender systems have become a convenient tool by which to reduce information overload for users and improve the user experience. Such systems have been applied on almost all e-commerce platforms to mine potential items for users [1]. Typically, these platforms collect user-item interactions such as *click*, *purchase*, *add-to-cart*, and *view*. Such multi-behavior data constitutes informative user preference signals which are helpful for building fine-grained recommender systems [2]. However, most conventional recommender systems [3–10] are limited to leveraging a single type of behavioral data (in most cases, purchase behavior) or simply assuming that different user behaviors are the same while model training, leaving other informative behavioral data unexplored or not well-utilized.

To better leverage such multi-behavior data, researchers have begun to consider dif-

doi:10.6342/NTU202203790

ferent types of behavioral data when training the models. Even so, most pioneering works treat a single behavior as the target behavior and optimize the models based on the the target behavior only [11–15]. For example, MBGCN [13] applies graph neural networks to learn user and item representations by optimizing the target behavior; upon recommendation, for each user, items are ranked based on their similarity scores calculated with their representations against the user representation. Note that such approaches focus on target behavior prediction, leaving unexplored prediction of other behaviors, which are however also important and may provide useful business insights. For example, given a user who often puts apparel in his/her shopping cart and regularly purchases daily necessities, we could surmise that this user likes apparel but is less likely to purchase it due to its high prices. In this case, if the model were able to accurately predict potential apparel that will be added to the shopping cart, then the company could provide special offers to this user based on the predicted items. Therefore, various marketing strategies could be devised to improve the user experience if the model were capable of providing recommendations w.r.t. different user behavior. However, such target-behavior-based models (e.g., MBGCN) must be re-trained to predict other behaviors and thus are inefficient for large-scale datasets and many real-world applications.

More recently, some studies have applied multi-task learning to learn user and item embeddings as well as behavior embeddings for multi-behavior recommendation [16, 17], where these different embeddings are jointly embedded into the modeling process, after which the user, item, and behavior embeddings are aggregated for prediction. Such designs enable recommendation w.r.t. different user behavior with a unified model. For example, EHCF [16] assumes strong transfer relations among different behaviors and thus uses a transfer matrix to describe such relations; GHCF [17], in turn, leverages graph neural networks to aggregate information concerning users, items, and behaviors to achieve state-of-the-art performance. However, note that in existing approaches, every user shares

doi:10.6342/NTU202203790

the same behavior embedding given a specific behavior, which lacks the granularity necessary to capture individual user preferences, since it is more intuitive that each user has his/her own behavior embedding given a certain behavior. For example, while user A tends to add all of the items of interest into the shopping cart and then clear it on payday, user B tends to purchase daily necessities directly, while only adding luxury items to the cart. Moreover, to exploit each item's characteristics, an item should have its own behavior embedding under each behavior. For example, luxury items that normal users cannot afford are usually added to the cart, whereas basic commodities are usually purchased directly.

Inspired by this idea, we propose personalized multi-interaction preference ranking (PMiPR), a lightweight, interaction-level embedding learning framework for multi-behavior recommendation. In contrast to the above studies, which treat nodes as basic training units, PMiPR treats interaction as the basic training unit, thus modeling the similarity between multi-behavior user-item interactions in a more natural way under a pairwise ranking framework. Specifically, if two users interact with the same (or a different) item under the same behavior, then these two interactions are clustered together in the embedding space. Moreover, in the proposed framework, an interaction embedding (describing the relation for a user and an item under a certain behavior) is composed of not only the associated user and item embeddings but also the behavior embeddings corresponding to the user and the item. Each user and item has a different behavior embedding under that behavior; in this way we exploit user and item preferences in a more fine-grained manner. Additionally, to deal with the sparsity of less-frequent behaviors, we further present a simple yet effective approach to incorporate global behavior information. Notably, in contrast to many end-to-end recommendation models, PMiPR generates a set of user and item embeddings as well as user and item behavior embeddings for recommendation, so that in practice, many calculations are completed offline or approximated by nearest-neighbor

doi:10.6342/NTU202203790

search to deal with large-scale data.

1.2 Research Contributions

To summarize, the main contributions of this work are as follows:

- We propose personalized multi-interaction preference ranking (PMiPR) which learns fine-grained behavior embeddings for multi-behavior recommendation.
- We conduct extensive experiments on four real-world datasets, showing the superiority of the proposed method to predict various types of user behavior using a single unified model.
- We present an effective, efficient implementation, which has faster computation times than recent state-of-the art methods.[1]

1.3 Chapter Overview

This thesis is categorized into six sections:

- The first chapter is the introduction, which introduces the research topic of this work, and outlines the research motivation and research contributions.
- The second chapter is the literature review, which introduces the related researches and methods of recommender systems based on single-behavior and multi-behavior respectively.
- The third chapter is the research method, which introduces the problem formulation and proposed method in this research in detail.

[1] The source code will be available online at a GitHub repository upon publication.

doi:10.6342/NTU202203790

- The fourth chapter is the experimental setup, which introduces the roadmap of experiments and details the experimental settings including datasets description, datasets preprocessing, baselines introduction, parameters settings and evaluation metrics.
- The fifth chapter is the experimental results, which presents the several experimental results in both table and graphical.
- The sixth chapter is the conclusion and future work, which summarizes the experiments of this research and discusses possible future improvements.

doi:10.6342/NTU202203790

Chapter 2

Literature Review

In this chapter, we introduce the background knowledge of our research in Section 2.1, after that we introduce some recommendation methods including single-behavior recommendation and multi-behavior recommendation in Sections 2.2 and 2.3 respectively.

2.1 Background of Recommender System

Recommender system is a widely used service to alleviate information overload for users, it has been using in various platforms to predict the "preference" or "rating" that a user would give to an item [1]. Generally, there are two types of user feedback data, which categorized as implicit feedback data, such as viewing, clicking, adding to cart list, and purchasing items, and explicit feedback data, such as rating, both are shown in the user-item interaction matrix in Figure 2.1. Implicit feedback data is a binary matrix that assigns 1 to the interacted items and 0 to the unobserved/not interested items, while explicit feedback data is a rating matrix that shows the rating of a user given to an item. The former aims to predict whether a user will interact to unobserved items in future while the latter aims to predict the possible rating that a user will give to unobserved items.

In real-world scenarios, most feedback is not explicit but implicit, since user implicit

doi:10.6342/NTU202203790

	i_1	i_2	i_3	i_4
u_1	1	0	0	1
u_2	1	1	0	0
u_3	0	1	1	0
u_4	1	0	0	1

(a) Implicit feedback data

	i_1	i_2	i_3	i_4
u_1	1	?	?	2
u_2	3	5	?	?
u_3	?	4	3	?
u_4	4	?	?	2

(b) Explicit feedback data

Figure 2.1: An example of implicit and explicit feedback data.

feedback can be collected easily and such feedback is already available in many platforms, such as users' behavior records in e-commerce platforms. As a result, most studies focus on recommendation models that work well with implicit feedback data.

2.2 Single-behavior Recommendation

First, we review some studies in single-behavior scenarios. Singular Value Decomposition based (SVD) model such as SVD++ [18] is one of the traditional work that factorizes the binary interaction matrix and assume that users dislike unobserved items. Recent works [19,20] utilize Bayesian Personalized Ranking (BPR) [7] to learn user and item representations, where BPR is a pairwise learning algorithm which assumes that users prefer the observed items to the unobserved items.

Vanilla matrix factorization assumes that the latent space and the original representation space are linearly mapped, such setting may not sufficient to capture the complex structure of user interaction data, so many recent studies start to utilize Deep Neural Network (DNNs) to better learn the complex mapping between these two spaces. Deep Matrix Factorization (DMF) [21] proposed a neural network architecture to map the users and items into a common low-dimensional space with non-linear projection. Neural network-based Collaborative Filtering (NCF) [20] replaces the inner product with a neural archi-

doi:10.6342/NTU202203790

tecture that can learn an arbitrary function from data, containing the fusion of Generalized Matrix Factorization (GMF) and Multi-Layer Perceptron (MLP). Deep Collaborative Filtering (DeepCF) [22] incorporates the collaborative filtering methods based on representation learning and matching function learning to improve the performance. Attentive Collaborative Filtering (ACF) [23] proposes an attention mechanism in CF that introduces item- and component-level attention model to assign attentive weights for inferring the underlying users' preferences encoded in the implicit user feedback.

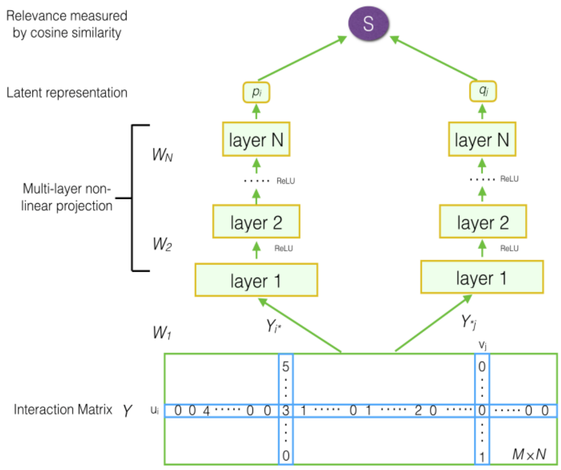


Figure 2.2: An illustration of DMF model [21]

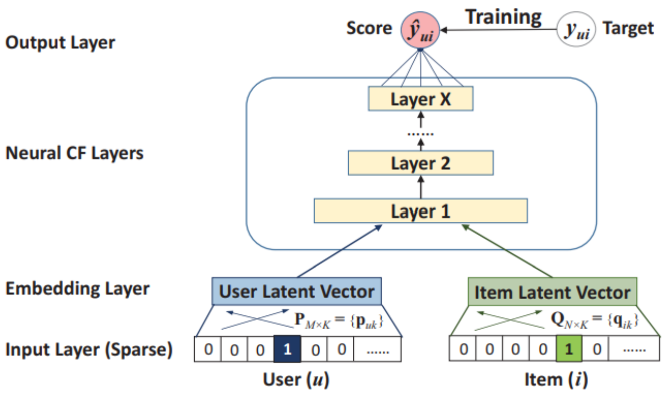


Figure 2.3: An illustration of NCF model [20].

doi:10.6342/NTU202203790

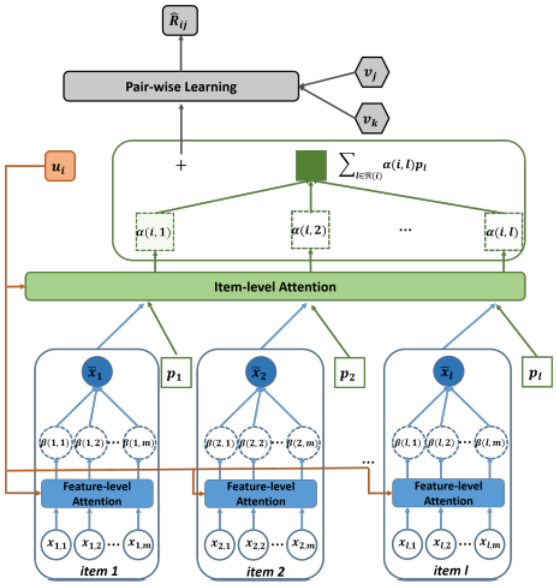


Figure 2.4: An illustration of ACF model [23].

In recent years, inspired by Graph Neural Networks (GNNs), which represents data in graph structure and apply neighborhood aggregation to learn node representations, researchers start to focus on development of graph-based recommendation. Neural Graph Collaborative Filtering (NGCF) [24] explicitly encodes the collaborative signal in the form of high-order connectivities in the user-item integration graph by performing embedding propagation. LightGCN [9] further improve NGCF by discarding two standard operations (feature transformation and nonlinear activation) in GCNs but inevitably increase the training difficulty. Disentangled Graph Collaborative Filtering (DGCF) [10] considers user-item relationships at the finer granularity of user intents and generates disentangled representations to improve the performance.

doi:10.6342/NTU202203790

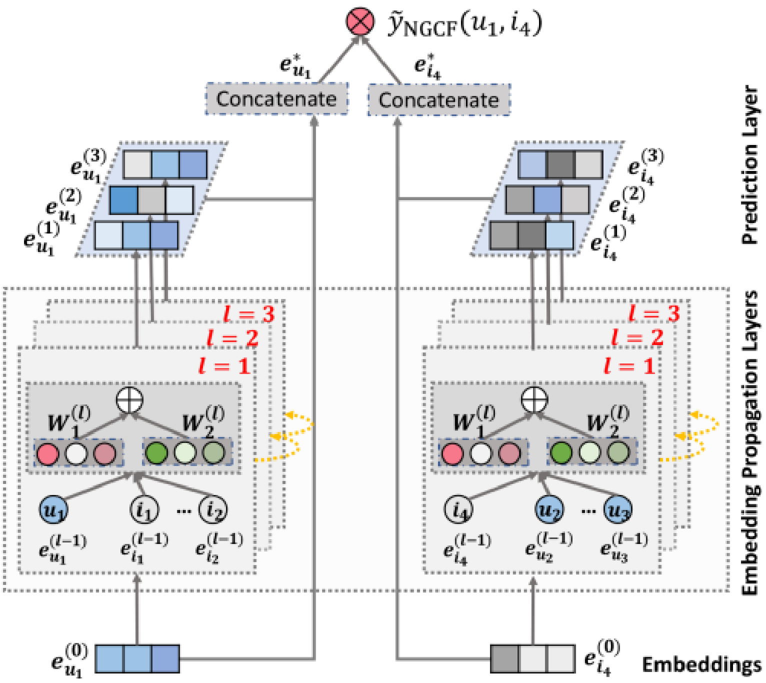


Figure 2.5: An illustration of NGCF model [24]. (the arrowed lines present the flow of information). The representations of user u_1 (left) and item i_4 (right) are refined with multiple embedding propagation layers, whose outputs are concatenated to make the final prediction.

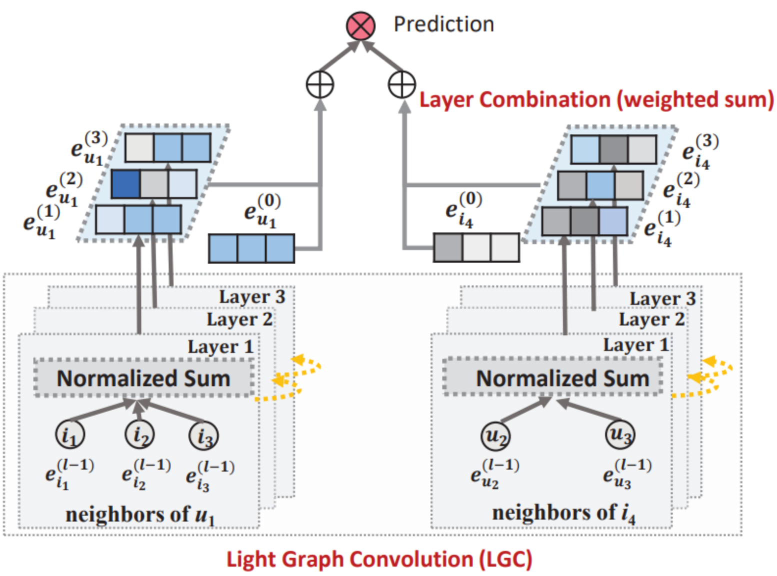


Figure 2.6: An illustration of LightGCN model [13]. In LightGCN, only the normalized sum of neighbor embeddings is performed towards next layer; other operations like self-connection, feature transformation, and nonlinear activation are all removed, which largely simplifies GCNs. In Layer Combination, we sum over the embeddings at each layer to obtain the final representations.

doi:10.6342/NTU202203790

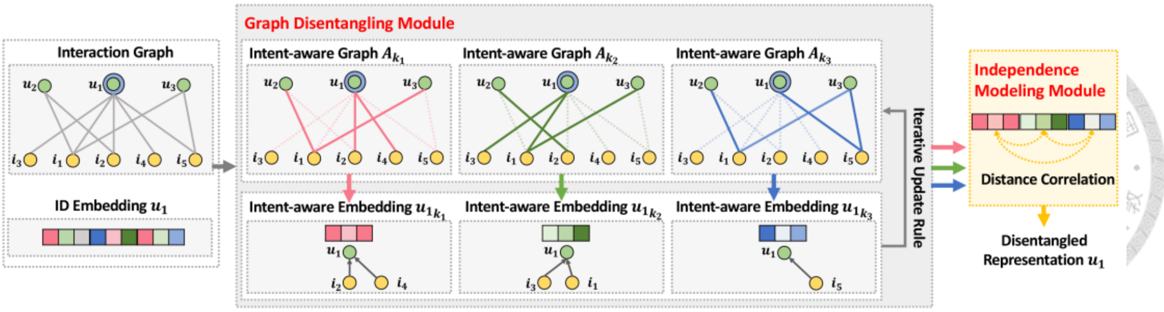


Figure 2.7: An illustration of DGCF model [10].

Existing single-behavior models have achieved great success in single-behavior recommendation. However, in real-word scenarios, the collected users' behavioral data is multi-behavior. For example, in e-commerce platform, a user can click, add-to-cart and purchase an item. So several studies start to utilize such informative behavioral data to further improve the recommendation performance, which is discussed in next section.

2.3 Multi-behavior Recommendation

Multi-behavior recommendation leverages multiple types of user-item interactions into training process to improve the recommendation performance on the target behavior, an example of multiple types of user feedback is shown in Figure 2.8. In general, existing studies of multi-behavior recommendation are divided into two categories: single-task learning and multi-task learning. We will describe each category in the following subsections.

doi:10.6342/NTU202203790

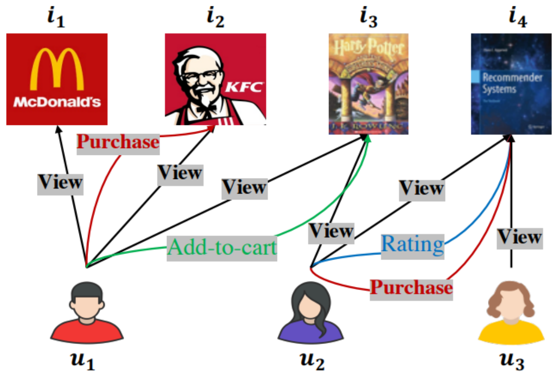


Figure 2.8: An example of multiple types of user feedback [17].

2.3.1 Single-task Learning

Single-task Learning leverages all behavioral data into training process, but only optimizes the target behavior to learn user and item representation, such setting only can predict items on target behavior. Early studies perform multiple learning of different behaviors by extending the Matrix Factorization (MF) [25–28]. For example, Collective Matrix Factorization (CMF) [27] decomposes the data matrices of multiple behavior types simultaneously to correlate the multiple factorization processes by sharing the embeddings of common entities. Apart from that, some studies [14, 15, 29] apply different negative sampling strategies to get better performance on the target behavior prediction. For example, Multi-feedback Bayesian Personalized Ranking (MFBPR) [14] extends the standard BPR sampling model by exploiting the difference in strength among user feedback 'channels' . Memory-Augmented Transformer Networks (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.

doi:10.6342/NTU202203790

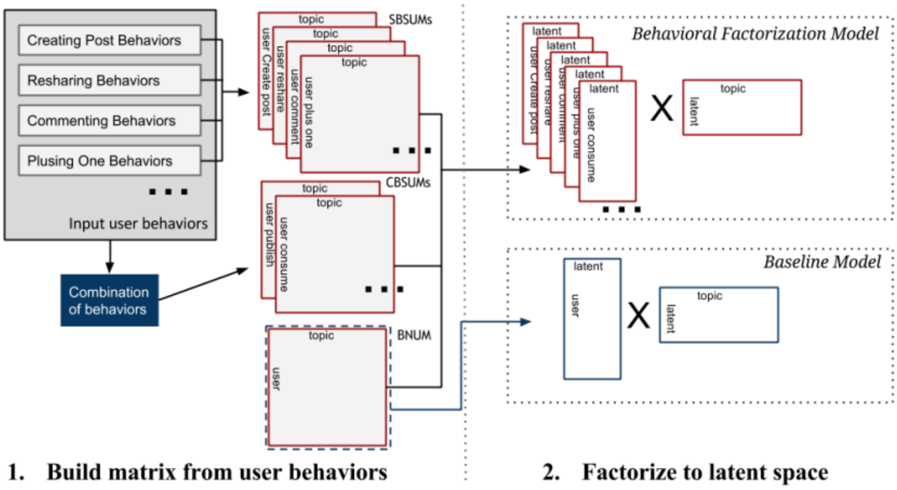


Figure 2.9: An illustration of CMF model [27]

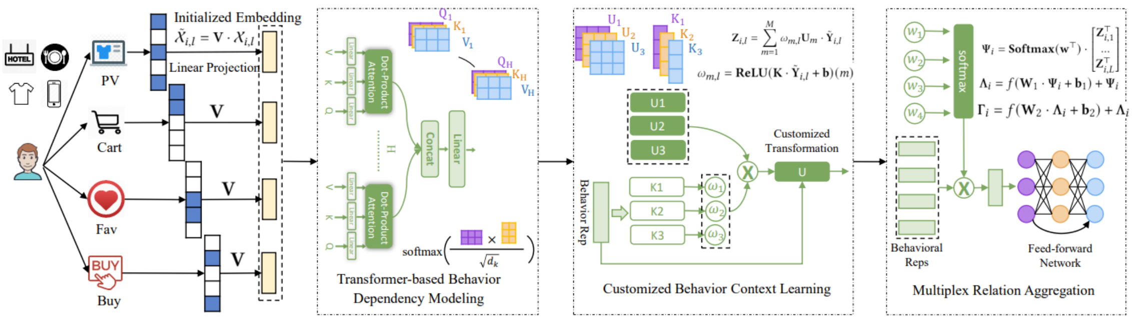


Figure 2.10: The model architecture of the proposed MATN framework [11]. The initialized embedding layer shares parameters across different behavior types. The transformer-based behavior dependency encoder takes all kinds of behavioral interaction data for dependency modeling. Different types of behaviors are individually transformed by the customized context learning with shared key and memory slots.

Recently, inspired by several GNN-based studies on single-behavior [8–10], Multi-Behavior Graph Convolutional Network (MBGCN) [13] applies GNN to learn user and item embedding through optimizing the target behavior. Knowledge-Enhanced Hierarchical Graph Transformer Network (KHGT) [12] perform the joint information aggregation over the user-item and item-item collaborative relations in multiple knowledge-aware behavior modalities. However, these models need to be re-evaluated and re-tuned for prediction on other behaviors by changing the target behavior, which are inefficient for large-scale datasets and many real-world applications.

doi:10.6342/NTU202203790

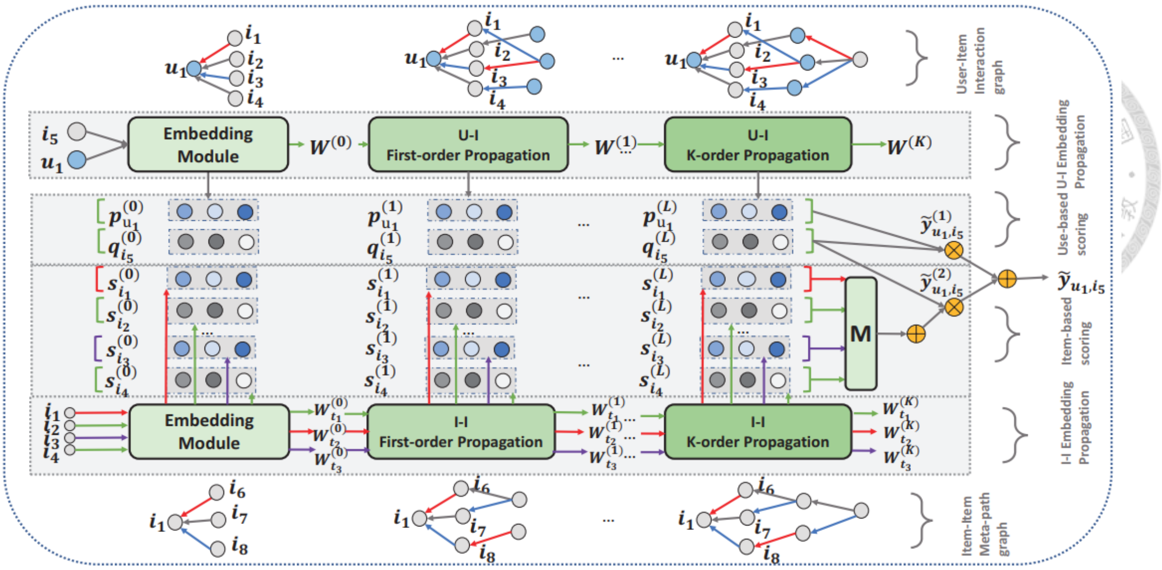


Figure 2.11: An illustration of MBGCN model [13], where node u_1 is the target user and i_5 is the target item.

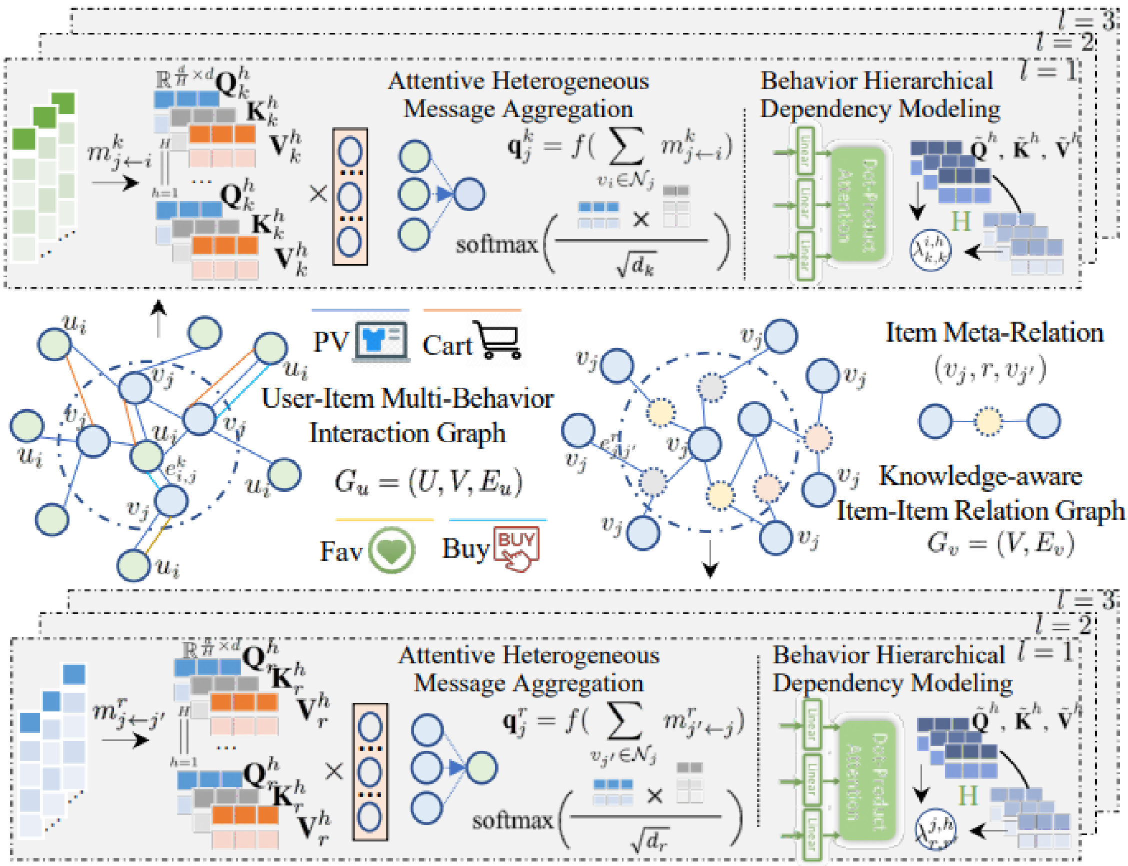


Figure 2.12: An illustration of KHGT model [12].

doi:10.6342/NTU202203790

2.3.2 Multi-task Learning

This category of works apply multi-task learning strategy into the modeling process and able to predict various types of user behaviors using one unified model. Neural Multi-Task Recommendation (NMTR) [30] assume there exists a order between behaviors, and proposed a sequential model that adopts NCF [20] for each type of behavior under multi-task learning framework, where the optimization on each behavior is treated as a task. Efficient Heterogeneous Collaborative Filtering (EHCF) [16] assumes there are strong transfer relations among different behaviors then using a transfer matrix to describe it. Contrastive Meta Learning (CML) [31] distills transferable knowledge across different types of behaviors via the constructed contrastive loss. Graph Heterogeneous Collaborative Filtering (GHCF) [17] leverages the GNN to aggregate the information of users, items and behaviors information to achieve state-of-the-art performance. Note that EHCF and GHCF jointly embeds both representations of users, items and behaviors for multi-behavior prediction and perform the advanced efficient non-sampling optimization under a multi-task learning framework.

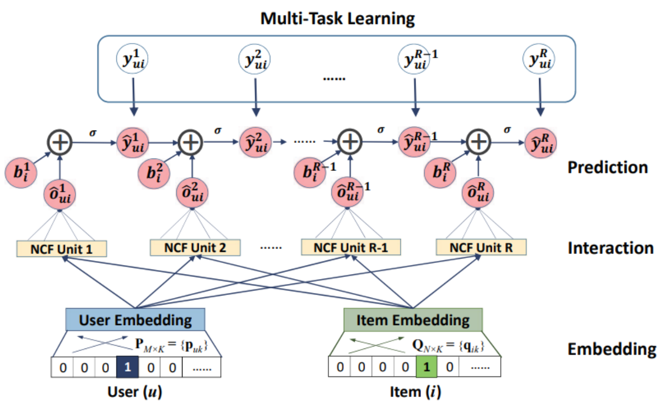


Figure 2.13: An illustration of NMTR model [30].

doi:10.6342/NTU202203790

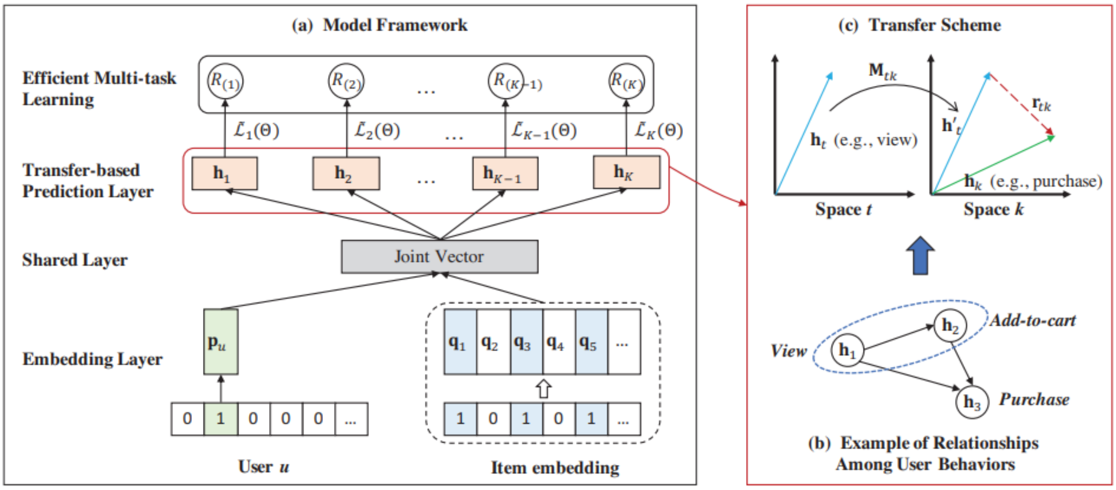


Figure 2.14: An illustration of EHCF model [16]. (a) Illustration of the model framework. (b) An example of the relationships among behaviors, where h_1, h_2, and h_3 denotes the prediction functions of behaviors: view, add-to-cart, and purchase, respectively (note that EHCF is not limited to these examples). (c) Illustration of the transfer scheme of two relational behaviors.

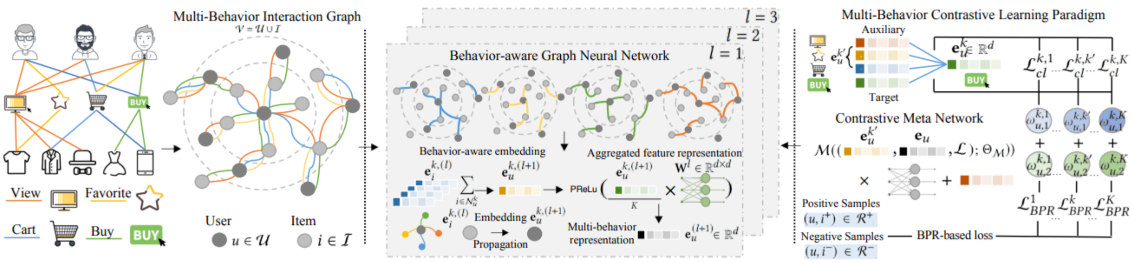


Figure 2.15: An illustration of CML model [31].

doi:10.6342/NTU202203790

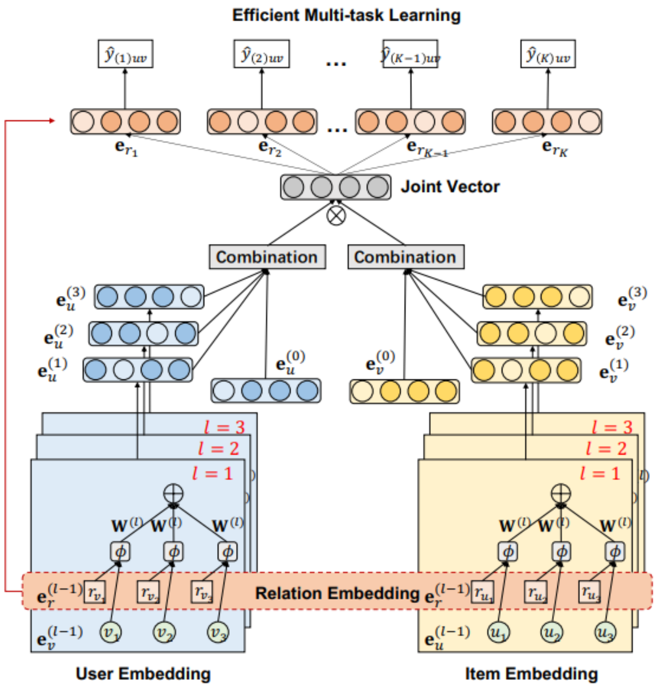


Figure 2.16: An illustration of GHCF model [17]

In our work, we build a multi-behavior user-item interactions graph based on users' and items' behavioral information, and leverage the interaction based pairwise ranking framework to learn fine-grained behavior embeddings for multi-behavior recommendation, which are introduced in next chapter.

doi:10.6342/NTU202203790

Chapter 3

Research Method

To better exploit the various types of user behaviors, in this work we propose personalized multi-interaction preference ranking (PMiPR), a unified embedding learning framework for multi-behavior recommendation (see Fig. 3.1 for an overview of the framework). In this section we first define the problem formulation in our work in Sections 3.1. After that, we detail embedding learning for the proposed PMiPR in Sections 3.2 and 3.3, after which we present a strategy to sample behavioral-based interaction triplets for optimization in Sections 3.4 and 3.5. Then we summarize the method with the procedure shown in Algorithm 1. Additionally, we propose the concept of global behavior embedding to deal with the sparsity of less-frequent behavior in Section 3.6. Finally, we detail the scoring functions used for recommending items w.r.t. different behaviors in Section 3.7.

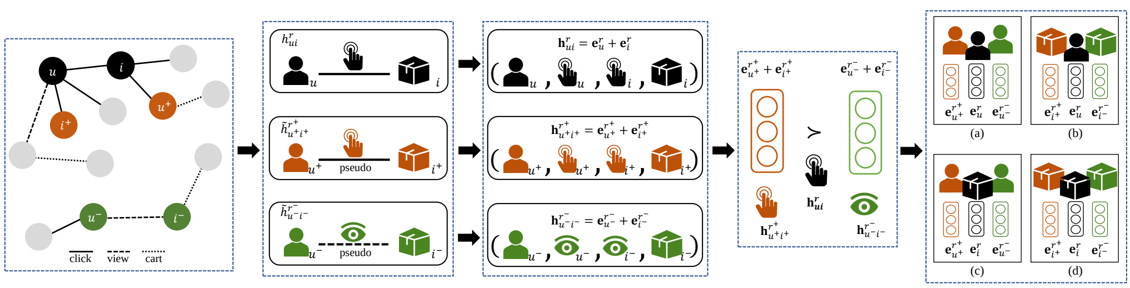


Figure 3.1: Overview of proposed PMiPR framework

doi:10.6342/NTU202203790

3.1 Problem Formulation

In this section, we formulate the problem of multi-behavior recommendation. In real-world scenarios, users of online information systems interact with items in many ways. Take the social website Reddit, for an example: a Reddit user can interact with a post by clicking, pushing, sharing, replying, and creating. Nevertheless, most conventional recommendation algorithms are designed only for a single type of user-item interaction, or simply assume that different user behaviors are the same. For instance, e-commerce recommender systems target purchase behavior and thus ignore behaviors such as clicks and views. In this study, we seek to design a unified, interaction-level embedding learning framework to better exploit different types of users and item behaviors for multi-behavior recommendation.

Definition 3.1.1 (Multi-behavior User-item Interaction Graph)**.** Let U, I, and R denote the set of users, items, and behaviors, respectively. A multi-behavior user-item interaction graph is an indirect graph defined as $G(\mathcal{V}, \mathcal{E}, \psi)$, where $\psi(\cdot)$ is an edge-type mapping function $\psi : \mathcal{E} \rightarrow 2^R$, $\mathcal{V}$ and $\mathcal{E}$ denote the sets of all nodes (i.e., $\mathcal{V} = U \cup I$) and all edges in the graph, respectively, and $(u, i) \in \mathcal{E}$ denotes an edge between a user $u \in U$ and an item $i \in I$.

Note that the co-domain of $\psi(\cdot)$ is the power set of R, as general multi-behavior recommendation scenarios sometimes involve more than one behavior between a user and an item. For example, a Reddit user $u \in U$ can interact with a post $i \in I$ both by clicking $r_i \in R$ and by sharing $r_j \in R$; in this case, for edge $(u, i) \in \mathcal{E}$ between u and i, $\psi((u, i)) = \{r_i, r_j\}$.

Definition 3.1.2 (Personalized Multi-interaction Preference Ranking for Multi-behavior Recommendation)**.** Given the multi-behavior user-item interaction graph $G(\mathcal{V}, \mathcal{E}, \psi)$ defined in Definition 3.1.1, our goal is to learn an embedding matrix $\Theta \in \mathbb{R}^{(n+m)\times(1+k)\times d}$,

doi:10.6342/NTU202203790

where $n = |U|$, $m = |I|$, $k = |R|$, and d denotes the embedding size, to better address the problem of multi-behavior recommendation. Specifically, for each user u (or each item i), the model generates its own personalized user embedding denoted as θ_u (or item embedding denoted as θ_i, respectively). Additionally, associated with each user (or item) is a behavior embedding for each behavior $r_i \in R$, denoted as $\theta_u^{r_i}$ (or $\theta_i^{r_i}$, respectively), resulting in $(n + m) \times (1 + k)$ learned embeddings in total. It is expected that with our unified learning framework, the learned embedding matrix Θ properly encodes different types of user-item interactions for recommendation. Furthermore, the proposed model enables us to make the recommendation w.r.t. different behaviors by leveraging the designated behavior embeddings for users and items.[1]

3.2 Multi-interaction Preference Ranking

The proposed PMiPR models various types of user-item interactions in a unified framework, generating a universal embedding matrix Θ for multi-behavior recommendation. We consider this to be a universal framework as the proposed model enables us to make the recommendation w.r.t. different behaviors via the learned universal embedding matrix Θ. In the mainstream literature on recommender systems, node embeddings are used to capture relations between users and items via matrix factorization and derivative techniques (e.g., [7,9,20,24,32,33]). Such methods, however, typically ignore the differences between various user behaviors; thus, they do not properly leverage the different types of interactions, nor do they offer recommendations w.r.t. different behaviors.

We address this problem with PMiPR, a pairwise interaction-level ranking algorithm for modeling the preferences of users and items under different behaviors. Inspired by [34], which changes the main idea of most ranking-based recommendation algorithms

[1] Note that in this paper, we interpret the term "multi-behavior recommendation" as making the recommendation w.r.t. different behaviors.

doi:10.6342/NTU202203790

from node-level [7] to interaction-level modeling and clusters similar user-item interactions in a self-supervised manner, we further construct behavior embeddings for all users and items with different behaviors to facilitate the various recommendations.

Let $\mathcal{H}$ be the set containing all user-item interactions from $G(\mathcal{V}, \mathcal{E}, \psi)$, where each element $h^r_{ui} \in \mathcal{H}$ denotes a user-item interaction in which user u interacts with item i with behavior r. Given an interaction h^r_{ui}, we define a basic training unit (also known as an interaction-level triplet) of the proposed PMiPR as

$$(h^r_{ui}, \tilde{h}^{r^+}_{u^+i^+}, \tilde{h}^{r^-}_{u^-i^-}), \tag{3.1}$$

along with the relation $\tilde{h}^{r^+}_{u^+i^+} \succ_{h^r_{ui}} \tilde{h}^{r^-}_{u^-i^-}$ (see the second panel from the left in Figure 3.1). This relation denotes that the *pseudo positive interaction* $\tilde{h}^{r^+}_{u^+i^+}$ is "more alike" to h^r_{ui} than the *pseudo negative interaction* $\tilde{h}^{r^-}_{u^-i^-}$. Note that in this framework, $\tilde{h}^{r^+}_{u^+i^+}$ and $\tilde{h}^{r^-}_{u^-i^-}$ are built artificially regarding h^r_{ui} and do not necessarily appear in the graph $G(\mathcal{V}, \mathcal{E}, \psi)$; thus, they can be freely defined to correspond to different application scenarios. For general multi-behavior recommendation, we propose a strategy to sample such triplets for optimization, as detailed in Section 3.4.

With the triplet definition in Eq. (3.1), we then construct training data $D_{\mathcal{H}} : \mathcal{H} \times \mathcal{H}^+_{h^r_{ui}} \times \mathcal{H}^-_{h^r_{ui}}$ as

$$\begin{aligned} D_{\mathcal{H}} := \Big\{ \left(h^r_{ui}, \tilde{h}^{r^+}_{u^+i^+}, \tilde{h}^{r^-}_{u^-i^-}\right) \Big| \, h^r_{ui} \in \mathcal{H} \wedge \\ \tilde{h}^{r^+}_{u^+i^+} \in \mathcal{H}^+_{h^r_{ui}} \wedge \tilde{h}^{r^-}_{u^-i^-} \in \mathcal{H}^-_{h^r_{ui}} \Big\}, \end{aligned} \tag{3.2}$$

where $\mathcal{H}^+_{h^r_{ui}}$ ($\mathcal{H}^-_{h^r_{ui}}$) denotes the set of pseudo positive interactions (that of negative interactions, respectively) w.r.t. h^r_{ui}.

doi:10.6342/NTU202203790

3.3 Embedding Matrix Learning

The pioneer work on interaction-level preference ranking [34] only utilizes user and item embeddings to model different user-item ratings, which is difficult to generalize for recommendations w.r.t. different behavior for multi-behavior recommendation. To this end, in addition to the original user and item embedding (i.e., θ_u and θ_i (see Definition 3.1.2), we further incorporate behavior embeddings for users and items to model different types of user-item interactions, resulting in $(\theta_u, \theta_u^r, \theta_i, \theta_i^r)$ for each user-item interaction. Thus (θ_u, θ_u^r) can be treated as a user u with his/her behavioral preference, while (θ_i, θ_i^r) denotes an item i with its behavior characteristics. For example, daily necessities are more frequently purchased than luxury goods; in our design, this is modeled properly via behavior embeddings for items.

Next, given an interaction h_{ui}^r (or a pseudo interaction $\tilde{h}_{u^+i^+}^{r^+}$ or $\tilde{h}_{u^-i^-}^{r^-}$), we define its embedding as $\mathbf{h}_{ui}^r := f(\theta_u, \theta_u^r, \theta_i, \theta_i^r)$, where $f(\cdot)$ is an arbitrary function to combine the user, item, and behavior embeddings. Note that instead of using $\tilde{\mathbf{h}}_{u^+i^+}^{r^+}$ or $\tilde{\mathbf{h}}_{u^-i^-}^{r^-}$ for denoting the embeddings of pseudo interactions, we refer to these as $\mathbf{h}_{u^+i^+}^{r^+}$ or $\mathbf{h}_{u^-i^-}^{r^-}$ to simplify the notation. In this paper, for similar reasons to those in [9], we adopt the addition operator as our aggregator, resulting in

$$\mathbf{h}_{ui}^r = \theta_u + \theta_u^r + \theta_i + \theta_i^r. \tag{3.3}$$

Other operators will be investigated in the future.

With $D_{\mathcal{H}}$, our objective is to find an embedding matrix Θ that maximizes the likelihood function from observed multi-behavior user-item interactions:

$$\mathcal{O}_{\text{PMiPR}} = \prod_{t \in D_{\mathcal{H}}} p\left(\tilde{h}_{u^-i^-}^{r^-} \prec_{h_{ui}^r} \tilde{h}_{u^+i^+}^{r^+} \middle| \Theta\right), \tag{3.4}$$

doi:10.6342/NTU202203790

where $t = (h_{ui}^r, \tilde{h}_{u^+i^+}^{r^+}, \tilde{h}_{u^-i^-}^{r^-})$. Furthermore, with the definition of the interaction embeddings in Eq. (3.3), the individual probability that an interaction $\tilde{h}_{u^+i^+}^{r^+}$ is more similar to h_{ui}^r than $\tilde{h}_{u^-i^-}^{r^-}$ is defined as

$$p\left(\tilde{h}_{u^-i^-}^{r^-} \prec_{h_{ui}^r} \tilde{h}_{u^+i^+}^{r^+} \middle| \Theta\right) = \sigma\left(\left\langle \mathbf{h}_{ui}^r, \mathbf{h}_{u^+i^+}^{r^+} - \mathbf{h}_{u^-i^-}^{r^-} \right\rangle\right), \tag{3.5}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\langle \cdot, \cdot \rangle$ denotes the dot product between two vectors.

With Eqs. (3.3)–(3.5), we formulate the maximum posterior estimator to derive the optimization criterion for the proposed PMiPR as

$$\begin{aligned}
\text{PMiPR-OPT} &:= \ln p(\Theta| \prec_{h_{ui}^r}) \propto \ln p\left(\prec_{h_{ui}^r} |\Theta\right) p\left(\Theta\right) \\
&= \ln \prod_{t \in D_{\mathcal{H}}} p\left(\tilde{h}_{u^-i^-}^{r^-} \prec_{h_{ui}^r} \tilde{h}_{u^+i^+}^{r^+}\right) p(\Theta) \\
&= \sum_{t \in D_{\mathcal{H}}} \ln \sigma\left(\left\langle \mathbf{h}_{ui}^r, \mathbf{h}_{u^+i^+}^{r^+} - \mathbf{h}_{u^-i^-}^{r^-} \right\rangle\right) - \lambda_\Theta ||\Theta||^2,
\end{aligned} \tag{3.6}$$

where λ_Θ is a model-specific regularization parameter.

To explore the advantages of such behavioral-based interaction triplets, we further decompose the interaction embedding $\mathbf{h}_{ui}^r$ into two components $\mathbf{e}_u^r$ and $\mathbf{e}_i^r$ for analysis, where $\mathbf{e}_u^r := \theta_u + \theta_u^r$ and $\mathbf{e}_i^r := \theta_i + \theta_i^r$ (see Eq. (3.3)). Then, the likelihood in Eq. (3.5) can be rewritten as

$$\begin{aligned}
\sigma\left(\left\langle \mathbf{h}_{ui}^r, \mathbf{h}_{u^+i^+}^{r^+} - \mathbf{h}_{u^-i^-}^{r^-} \right\rangle\right) &= \sigma\left(\left\langle \mathbf{e}_u^r + \mathbf{e}_i^r, \left(\mathbf{e}_{u^+}^{r^+} + \mathbf{e}_{i^+}^{r^+}\right) - \left(\mathbf{e}_{u^-}^{r^-} + \mathbf{e}_{i^-}^{r^-}\right) \right\rangle\right) \\
&= \sigma\left(\left\langle \mathbf{e}_u^r, (\mathbf{e}_{u^+}^{r^+} - \mathbf{e}_{u^-}^{r^-}) + \left(\mathbf{e}_{i^+}^{r^+} - \mathbf{e}_{i^-}^{r^-}\right) \right\rangle\right. \\
&\qquad \left. + \left\langle \mathbf{e}_i^r, \left(\mathbf{e}_{u^+}^{r^+} - \mathbf{e}_{u^-}^{r^-}\right) + \left(\mathbf{e}_{i^+}^{r^+} - \mathbf{e}_{i^-}^{r^-}\right) \right\rangle\right).
\end{aligned} \tag{3.7}$$

The above likelihood in Eq. (3.7) can be decomposed into the following four components:

doi:10.6342/NTU202203790

1. $\langle \mathbf{e}_u^r, \mathbf{e}_{u^+}^{r^+} - \mathbf{e}_{u^-}^{r^-} \rangle$: Models the user interaction similarity to user u with behavior r regarding users u^+ with behavior r^+ and u^- with behavior r^-.

2. $\langle \mathbf{e}_u^r, \mathbf{e}_{i^+}^{r^+} - \mathbf{e}_{i^-}^{r^-} \rangle$: Models the item interaction preference ranking between item i^+ with behavior r^+ and i^- with behavior r^- for user u with behavior r.

3. $\langle \mathbf{e}_i^r, \mathbf{e}_{u^+}^{r^+} - \mathbf{e}_{u^-}^{r^-} \rangle$: Models the user interaction preference ranking between user u^+ with behavior r^+ and u^- with behavior r^- for item i with behavior r.

4. $\langle \mathbf{e}_i^r, \mathbf{e}_{i^+}^{r^+} - \mathbf{e}_{i^-}^{r^-} \rangle$: Models the item interaction similarity to item i with behavior r regarding items i^+ with behavior r^+ and i^- with behavior r^-.

Note that each of the above components corresponds to (a)–(d) in the rightmost panel of Figure 3.1. Moreover, for (a) (or d)), the model tends to cluster users (items, respectively) that involve similar interactions with items (users, respectively) in the embedding space; as for (b) and (c), the model tends to cluster users and items that involve similar interactions with each other in the embedding space. Such a design not only enables fine-grained modeling for different types of user-item interactions but also naturally yields a powerful representation matrix Θ that is suitable for various behavioral recommendation tasks.

3.4 Sampling Strategy

Recall that the pseudo positive interaction $\tilde{h}_{u^+i^+}^{r^+}$ and the negative equivalent $\tilde{h}_{u^-i^-}^{r^-}$ in a triplet given h_{ui}^r defined in Eq. (3.1) can be freely defined to correspond to different application scenarios. For general multi-behavior recommendation, we propose a strategy to sample such triplets to construct the training data $D_{\mathcal{H}}$ in Eq. (3.2) for optimization. Specif-

doi:10.6342/NTU202203790

ically, for an interaction h^r_{ui} in $G(\mathcal{V}, \mathcal{E}, \psi)$, this is the set of pseudo positive interactions:

$$\mathcal{H}^+_{h^r_{ui}} := \Big\{ \tilde{h}^{r^+}_{u^+i^+} \,\Big|\, (u, i^+), (u^+, i) \in \mathcal{E} \,\wedge \\ r^+ = r \wedge r^+ \in \psi(u, i^+) \wedge r^+ \in \psi(u^+, i) \Big\}. \tag{3.8}$$

We illustrate this sampling strategy with Fig. 3.1 (see the leftmost panel of the figure). Given an interaction between user u and item i with a specific behavior r (the solid line between two black nodes), i.e., h^r_{ui}, we sample a "pseudo" positive interaction $\tilde{h}^{r^+}_{u^+i^+}$ constructed by a sampled positive item i^+ (a neighbor of u) and a sampled positive user u^+ (a neighbor of i) with behavior r (both nodes are orange). For the pseudo negative interaction $\tilde{h}^{r^-}_{u^-i^-} \in \mathcal{H}^-_{h^r_{ui}}$, we randomly select an interaction from all interactions in $G(\mathcal{V}, \mathcal{E}, \psi)$ (i.e., $\mathcal{H}$) to construct $\mathcal{H}^-_{h^r_{ui}}$ for simplicity. We leave more complicated settings for future work.

3.5 Optimization

With the training data $D_{\mathcal{H}}$ in Eq. (3.2) and the objective function in Eq. (3.6), we optimize the embedding matrix as

$$\Theta \leftarrow \Theta + \alpha \left(\frac{\partial \text{PMiPR-OPT}}{\partial \Theta} \right), \tag{3.9}$$

where α is the learning rate. Specifically, for each given interaction $h^r_{ui} \in \mathcal{H}$, we randomly sample a pseudo positive interaction $\tilde{h}^{r^+}_{u^+i^+} \in \mathcal{H}^+_{h^r_{ui}}$ defined in Eq. (3.8) and a negative interaction $\tilde{h}^{r^-}_{u^-i^-} \in \mathcal{H}$. The resulting interaction-level triplet $(h^r_{ui}, \tilde{h}^{r^+}_{u^+i^+}, \tilde{h}^{r^-}_{u^-i^-}) \in D_{\mathcal{H}}$ is

doi:10.6342/NTU202203790

adopted to update the model parameter matrix Θ with this gradient:

$$\begin{aligned}\frac{\partial \text{PMiPR-OPT}}{\partial \Theta} &= \frac{\partial}{\partial \Theta} \ln \sigma(\hat{x}) - \lambda_{\Theta} \frac{\partial}{\partial \Theta} ||\Theta||^2 \\ &\propto \frac{e^{-\hat{x}}}{1+e^{-\hat{x}}} \frac{\partial}{\partial \Theta} \hat{x} - \lambda_{\Theta} \Theta,\end{aligned} \tag{3.10}$$

where

$$\hat{x} := \left\langle \mathbf{h}^{r}_{ui}, \mathbf{h}^{r^+}_{u^+i^+} - \mathbf{h}^{r^-}_{u^-i^-} \right\rangle = \left\langle \mathbf{e}^{r}_{u} + \mathbf{e}^{r}_{i}, \left(\mathbf{e}^{r^+}_{u^+} + \mathbf{e}^{r^+}_{i^+}\right) - \left(\mathbf{e}^{r^-}_{u^-} + \mathbf{e}^{r^-}_{i^-}\right) \right\rangle.$$

Additionally, we follow [32, 33, 35] by utilizing asynchronous stochastic gradient ascent (ASGD) [36] to efficiently update parameter matrix Θ in a parallel manner. Algorithm 1 details the complete model training procedure.

Algorithm 1 Training with proposed PMiPR algorithm

Inputs: $G(\mathcal{V}, \mathcal{E}, \psi)$, N iterations
Output: Θ

```
Randomly initialize Θ
𝓗 ← all user-item interactions from G(𝓥, 𝓔, ψ)
for e = 1 to N do
    Draw a user-item interaction h^r_ui from 𝓗
    Construct positive interaction set 𝓗^+_{h^r_ui}
    Construct negative interaction set 𝓗^-_{h^r_ui}
    Draw a positive interaction h^{r+}_{u+i+} ∈ 𝓗^+_{h^r_ui}
    Draw a negative interaction h^{r-}_{u-i-} ∈ 𝓗^-_{h^r_ui}
    Update Θ with Eqs. (3.9)–(3.10)
end for
```

3.6 Global Behavior Embedding

To fully leverage the multi-behavior information and account for the sparsity of less-frequent behaviors, we additionally incorporate global behavior information in the multi-behavior user-item interaction graph $G(\mathcal{V}, \mathcal{E}, \psi)$. That is, if there exist ℓ different types

doi:10.6342/NTU202203790

of behavior in a multi-behavior recommendation dataset, we additionally create a pseudo behavior, namely the global behavior r_g, resulting in $|R| = \ell + 1 = k$ types of behaviors in the graph. Specifically, for any edge (u, i) between a user u and an item i in $\mathcal{E}$, $r_g \in \psi((u, i))$. In the other words, there exists a global relation between user u and item i if u has interacted with i with any of the relations $r \in \{r | r \in R \wedge r \neq r_g\}$. Experimental results show the effectiveness of such a design in Section 5.2.

3.7 Scoring Function for Multi-behavior Recommendation

Algorithm 1 yields the embedding matrix $\Theta \in \mathbb{R}^{(n+m)\times(1+k)\times d}$. The learned Θ enables us to make recommendations w.r.t. different behaviors. Specifically, for any target behavior $r \in \{r | r \in R \wedge r \neq r_g\}$, at the inference stage, for each user u, we calculate the scores considering all items as

$$\hat{y}^r_{ui} = \left(\theta_u + \theta^r_u \,\middle\|\, \theta_u + \theta^{r_g}_u\right) \cdot \left(\theta_i + \theta^r_i \,\middle\|\, \theta_i + \theta^{r_g}_i\right), \tag{3.11}$$

where $a \| b$ denotes the concatenation of the embedding vectors a and b, and $r_g \in R$ is the global behavior (see Section 3.6). We then rank the items according to the score in Eq. (3.11) for each user to obtain the recommended items for the target behavior.

doi:10.6342/NTU202203790

Chapter 4

Experimental Setup

In this section, we first introduce the roadmap of experiments in Section 4.1. After that, we detail the experimental settings including dataset description, datasets preprocessing, baselines introduction, parameters settings and evaluation metrics in Section 4.2.

4.1 Roadmap for experiments

- **Experiment 1**: Overall performance comparison on various behavior.
- **Experiment 2**: Ablation studies on global behavior embeddings.
- **Experiment 3**: Sensitivity analysis on hyper-parameters.
- **Experiment 4**: Computational efficiency comparison.

The experiments seek to answer the following research questions (RQs):

- **RQ1**: How does the proposed framework perform compared to other state-of-the-art baseline models for the prediction of different behaviors?
- **RQ2**: How does the global behavior embedding influence our model's performance?

doi:10.6342/NTU202203790

- **RQ3**: How does the sensitivity of hyperparameters (α, λ and n) affect the performance of our model?
- **RQ4**: How does the computational time of our model compare with that of the compared models?

4.2 Experimental Settings

4.2.1 Datasets Description

We conducted experiments on four public recommendation datasets to evaluate PMiPR. All datasets contain three common types of e-commerce behaviors, as summarized in Table 4.1 and Fig. 4.1.

- **Beibei**[1]. This is the dataset obtained from Beibei, the largest infant product e-commerce platform in China. There are 21716 users and 7977 items with three types of behaviors, including purchase, cart and view collected in this dataset.
- **Taobao**[2]. This is the dataset collected from Tmall, the largest e-commerce platform in China. There are 48749 users and 39493 items with three types of behavior, including purchase, cart and view collected in this dataset.
- **Ecommerce**[3]. This is the dataset collected from a medium cosmetics online store. There are 55608 users and 48547 items with three types of behavior, including purchase, cart and view collected in this dataset.
- **Rees46**[4]. This is the dataset collected from a large multi-category online store.

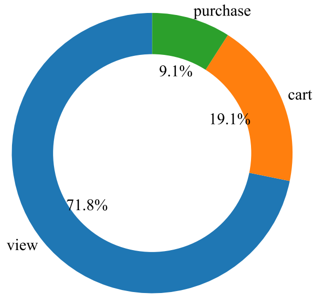
[1] https://www.beibei.com/
[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
[3] https://www.kaggle.com/datasets/mkechinov/ecommerce-events-history-in-cosmetics-shop
[4] https://www.kaggle.com/datasets/mkechinov/ecommerce-behavior-data-from-multi-category-store

doi:10.6342/NTU202203790

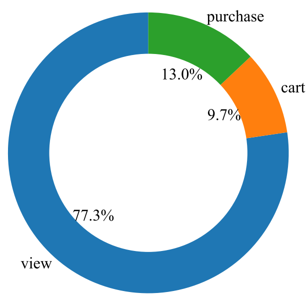
There are 20399 users and 31972 items with three types of behavior, including purchase, cart and view collected in this dataset.

Table 4.1: Dataset statistics

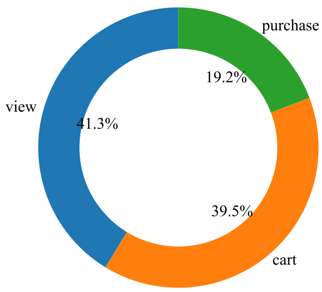
Dataset	#User	#Item	#View	#Cart	#Purchase
Beibei	21,716	7,977	2,412,586	642,622	304,576
Taobao	48,749	39,493	1,548,126	193,747	259,747
Ecommerce	55,608	48,547	1,945,122	1,860,450	905,847
Rees46	20,399	31,972	112,652	48,313	47,368



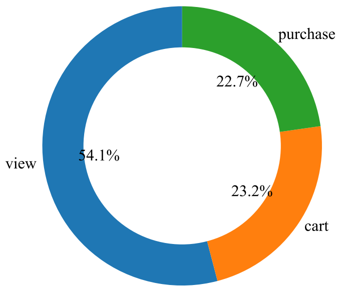
(a) Beibei



(b) Taobao



(c) Ecommerce



(d) Rees46

Figure 4.1: Dataset statistics in graphical.

4.2.2 Datasets Preprocessing

For Beibei, Taobao, and Ecommerce, we followed the settings in [17], which filters out users and items with fewer than five purchase interactions. For the smaller Rees46

doi:10.6342/NTU202203790

dataset, we filtered out users with fewer than two purchase interactions. For each dataset, we took each user's last *purchase* record as the test data for the purchase recommendation evaluation. We also took each user's last *cart* and *view* records as the test data for the cart and view behavior evaluations, respectively, for both Rees46 and Ecommerce. As Beibei and Taobao did not include timestamps for each interaction, we could not split the dataset for other behaviors like *view* and *cart*; therefore, for these, we evaluated the recommendation performance only for the purchase behavior, which is available in the original datasets provided and used in previous studies [13, 16, 17].

4.2.3 Baselines

To demonstrate the effectiveness of our method, we compared it with seven baseline methods. These baselines can be categorized into two groups: 1) single-behavior models that utilize only single-behavior data and neglect other behavioral data in the training process, and 2) multi-behavior models that consider all types of behavioral data in the training process.

Single-behavior models

- **BPR** [7]: a widely used pairwise learning framework that considers node-level triplets for model training.

- **LightGCN** [9]: a simplified architecture of the graph neural network (GNN) from NGCF [24] that usually achieves state-of-the-art performance for single-behavior recommendation.

Multi-behavior models

- **MC-BPR** [14]: a multi-behavior recommendation algorithm that assumes an importance order between different behaviors and extends BPR [7] by building sampling pairs with a type of positive behavior and a type of weaker behavior.

doi:10.6342/NTU202203790

- **NMTR** [30]: a neural model that involves joint optimization based on the multi-task learning framework, where the optimization on each behavior is treated as a task.
- **MBGCN** [13]: a graph convolutional network (GCN) that learns the strengths of different behaviors by the user-item propagation layer and item-item propagation layer.
- **EHCF** [16]: a non-sampling transfer learning solution model good for modeling both single- and multi-behavior data.
- **GHCF** [17]: a GCN-based model that jointly embeds user, item, and behavior representations for multi-behavior modeling, which also utilizes non-sampling optimization as in [16] to improve performance.

We re-trained single-behavior models for each type of behavior and evaluated the performance based on the corresponding model. For multi-behavior models, note the following: 1) as MC-BPR only models importance order between different behaviors and does not provide behavior-dependent recommendations, we use the same recommended list for different behavior evaluation; 2) as MBGCN utilizes the target behavior to optimize the loss function, we had to re-train the model for different behaviors by changing the target behavior for different behavior evaluation; 3) as NMTR, EHCF, and GHCF predict for each behavior via their multi-task learning frameworks, there was no need to re-train the model; thus evaluation for different behaviors leveraged the learned behavior embeddings.

4.2.4 Parameters Settings

We set dimension d of the embedding vectors of all the baselines and the proposed model to 128. For PMiPR, we set the L_2 regularization coefficient λ, learning rate α and negative sample n to 0.001, 0.025 and 5, respectively. For the baselines, we initialized the hyperparameters and used a grid search over different settings per the corresponding

doi:10.6342/NTU202203790

papers, selecting the hyperparameters that yielded the best performance. For each method, the final reported results were calculated by averaging the results over five repetitions.

4.2.5 Evaluation Metrics

Following previous studies [7, 37, 38], for each user in the test set under target behavior, we treated all items that the user has not interacted with as negative items. Then we used each method to generate a ranking list for each user with the user' s preference scores over all the items, except for the positive ones in the training set of target behavior. To evaluate the performance of the ranking list, we adopted two common metrics for top-N recommendation: recall (Recall@N) and normalized discount cumulative gain (NDCG@N) with $N = 10, 50, 100$ in our experiments.

- **Recall@N**. Recall measures whether the testing items is found in the top-N item ranking list (1 for yes and 0 for no).
- **NDCG@N**. Normalized Discounted Cumulative Gain (NDCG) is a position sensitive metric which hits in higher positions on the item ranking list are assigned a higher score.

Let $U_T = \{u_1, u_2, ..., u_n\}$ be the user test set. The generated recommendation item list for user $u \in U_T$ is denoted as $R_u = \{i^1, i^2, ..., i^N\}$, where N is the number of recommendation item, i^k is an item that ranked at the k-th position in R_u. T_u denotes the set of u' s interacted item in the test set. The above matrices are formulated as follows:

$$
\begin{aligned}
Recall@N &= \frac{1}{|U_T|} \sum_{u \in U_T} \frac{|T_u \cap R_u|}{|T_u|} \\
NDCG@N &= \frac{1}{Z} DCG@N = \frac{1}{Z|U|} \sum_{u \in U_T} \sum_{j=1}^{N} \frac{2^{I(|\{i^j\} \cap T_u|)} - 1}{\log_2(j+1)},
\end{aligned}
\tag{4.1}
$$

doi:10.6342/NTU202203790

in which $I(x)$ denotes an indicator function whose value is 1 when $x > 0$, and 0 otherwise, and Z is the maximum possible value of $DCG@N$ for normalization. Note that the larger value indicates better performance in both metrics.

doi:10.6342/NTU202203790

Chapter 5

Experimental Results

In this section, we describe experiments conducted on public datasets to demonstrate the effectiveness of the proposed PMiPR. We describe several experiments in Section 5.1 to Section 5.4

5.1 Experiment 1: Overall Performance Comparison on Various Behavior

In this section, we compared PMiPR with several single-behavior and multi-behavior baselines for recommendation tasks w.r.t. different behaviors, including *purchase*, *cart*, and *view* prediction, as shown in Tables 5.1–5.3. The best results are in boldface; the best-performing method among all the baselines is indicated by "†"; "Improv. (%)" indicates the percentage improvement of the proposed model w.r.t. the best-performing baselines. Below we separately discuss the results for the three prediction tasks.

doi:10.6342/NTU202203790

Table 5.1: Overall performance comparison on purchase recommendation

	Beibei					
	Recall			NDCG		
	@10	@50	@100	@10	@50	@100
BPR	0.0355	0.1276	0.2264	0.0182	0.0374	0.0533
LightGCN	0.0444	0.1339	0.1984	0.0212	0.0404	0.0501
MCBPR	0.0488	0.1969	0.3228	0.0226	0.0540	0.0743
NMTR	0.0414	0.2708	0.4534	0.0172	0.0651	0.0947
MBGCN	0.0582	0.3319	0.4823	0.0294	0.1506	0.171
EHCF	0.2424	0.4149	0.5009	0.1365	0.1748	0.1887
GHCF	†**0.2912**	†0.4595	†0.5395	†**0.1569**	†0.1947	†0.2077
PMiPR	0.2387	**0.5596**	**0.6514**	0.1173	**0.1957**	**0.2102**
Improv. (%)	-21.99%	21.78%	20.74%	-33.76%	0.51%	1.20%

	Taobao					
	Recall			NDCG		
	@10	@50	@100	@10	@50	@100
BPR	0.0342	0.0664	0.0824	0.0204	0.0274	0.0300
LightGCN	0.0438	0.0819	0.1001	0.0258	0.0342	0.0371
MCBPR	0.0713	0.1190	0.1423	0.0383	0.0488	0.0526
NMTR	0.0803	0.1308	0.1666	0.0411	0.0523	0.0581
MBGCN	0.1092	0.1854	0.2465	0.0553	0.0788	0.0802
EHCF	0.1175	0.2387	0.3108	0.0667	0.0931	0.1048
GHCF	†0.1359	†0.2833	†0.3676	†0.0768	†0.1090	†0.1226
PMiPR	**0.2187**	**0.5270**	**0.6287**	**0.1087**	**0.1765**	**0.1932**
Improv. (%)	60.93%	86.02%	71.03%	41.53%	61.93%	57.59%

doi:10.6342/NTU202203790

	Ecommerce					
	Recall			NDCG		
	@10	@50	@100	@10	@50	@100
BPR	0.0571	0.1269	0.1737	0.0320	0.0471	0.0547
LightGCN	0.0651	0.1853	0.2608	0.0332	0.0538	0.0660
MCBPR	0.0792	0.2070	0.2919	0.0407	0.0683	0.0820
NMTR	0.0691	0.2359	0.3543	0.0355	0.0630	0.0821
MBGCN	0.0657	0.1942	0.2790	0.0345	0.0591	0.0729
EHCF	0.1659	0.3881	0.5123	0.0907	0.1390	0.1592
GHCF	†0.2330	†0.4351	†0.5347	†0.1375	†0.1819	†0.1981
PMiPR	**0.2621**	**0.5532**	**0.6861**	**0.1447**	**0.2086**	**0.2302**
Improv. (%)	12.49%	27.14%	28.31%	5.24%	14.68%	16.20%

	Rees46					
	Recall			NDCG		
	@10	@50	@100	@10	@50	@100
BPR	0.0876	0.2201	0.2972	0.0473	0.0759	0.0883
LightGCN	0.1656	0.1864	0.3925	0.0914	0.1143	0.1371
MCBPR	0.1743	0.3087	0.3598	0.0837	0.1139	0.1222
NMTR	0.1724	0.3160	0.3759	0.0825	0.1141	0.1238
MBGCN	0.1783	0.3469	0.4199	0.0990	0.1365	0.1483
EHCF	0.3563	0.5624	0.6223	0.2161	0.2623	0.2720
GHCF	†0.3945	†0.5749	†0.6255	†0.2410	†0.2818	†0.2901
PMiPR	**0.4118**	**0.6070**	**0.6620**	**0.2569**	**0.3010**	**0.3099**
Improv. (%)	4.39%	5.58%	5.84%	6.60%	6.81%	6.83%

doi:10.6342/NTU202203790

Table 5.2: Overall performance comparison on cart recommendation

	Ecommerce					
	Recall			NDCG		
	@10	@50	@100	@10	@50	@100
BPR	0.0285	0.0697	0.1004	0.0153	0.0242	0.0291
LightGCN	0.0606	0.1245	0.2261	0.0324	0.0485	0.0644
MCBPR	0.0644	0.1702	0.2457	0.0407	0.0683	0.0820
NMTR	0.0744	0.1498	0.1951	0.0393	0.0558	0.0632
MBGCN	0.0708	0.1369	0.1894	0.0355	0.0521	0.0620
EHCF	0.0247	0.0743	0.1126	0.0121	0.0227	0.0289
GHCF	†0.0889	†0.2129	†0.2938	†0.0479	†0.0747	†0.0878
PMiPR	**0.1494**	**0.3592**	**0.4780**	**0.0803**	**0.1259**	**0.1451**
Improv. (%)	68.05%	68.72%	62.70%	67.64%	68.54%	65.26%

	Rees46					
	Recall			NDCG		
	@10	@50	@100	@10	@50	@100
BPR	0.1058	0.2587	0.3291	0.0577	0.0913	0.1027
LightGCN	0.0606	0.1245	0.2261	0.0324	0.0485	0.0644
MCBPR	0.1431	0.2631	0.3148	0.0685	0.0951	0.1035
NMTR	0.1718	0.3143	0.3736	0.0823	0.1137	0.1233
MBGCN	0.2257	0.4001	0.4846	0.1363	0.2029	0.2253
EHCF	0.3217	0.5339	0.5945	0.1896	0.2370	0.2469
GHCF	†0.3616	†0.5459	†0.5977	†0.2092	†0.2510	†0.2594
PMiPR	**0.4089**	**0.6004**	**0.6565**	**0.2516**	**0.2948**	**0.3039**
Improv. (%)	13.08%	9.98%	9.84%	20.27%	17.45%	17.15%

doi:10.6342/NTU202203790

Table 5.3: Overall performance comparison on view recommendation

	Ecommerce					
	Recall			NDCG		
	@10	@50	@100	@10	@50	@100
BPR	0.0257	0.0630	0.0890	0.0152	0.0232	0.0274
LightGCN	0.0560	0.1432	0.2025	0.0230	0.0487	0.0583
MCBPR	0.0433	0.1141	0.1649	0.0225	0.0378	0.0460
NMTR	0.0368	0.1161	0.1881	0.0189	0.0357	0.0473
MBGCN	0.0425	0.1298	0.1930	0.0296	0.0438	0.0531
EHCF	0.0314	0.0975	0.1482	0.0155	0.0296	0.0378
GHCF	†0.0766	†0.1796	†0.2465	†0.0411	†0.0634	†0.0742
PMiPR	**0.1015**	**0.2481**	**0.3337**	**0.0547**	**0.0866**	**0.1005**
Improv. (%)	32.51%	38.14%	35.38%	33.09%	36.59%	35.44%

	Rees46					
	Recall			NDCG		
	@10	@50	@100	@10	@50	@100
BPR	0.1964	0.3775	0.4647	0.1094	0.1491	0.1632
LightGCN	0.2294	0.3845	0.4510	0.1365	0.1708	0.1816
MCBPR	0.1449	0.2707	0.3234	0.0849	0.1125	0.1211
NMTR	0.1625	0.3007	0.3575	0.0779	0.1085	0.1176
MBGCN	0.1960	0.3953	†0.4804	0.1035	0.1472	0.1610
EHCF	†0.2137	†0.4101	0.4773	0.1132	†0.1567	†0.1676
GHCF	0.2120	0.3939	0.4553	†0.1137	0.1543	0.1643
PMiPR	**0.2863**	**0.4722**	**0.5366**	**0.1806**	**0.2220**	**0.2325**
Improv. (%)	33.97%	12.14%	11.70%	58.31%	41.67%	38.72%

doi:10.6342/NTU202203790

5.1.1 Purchase Recommendation

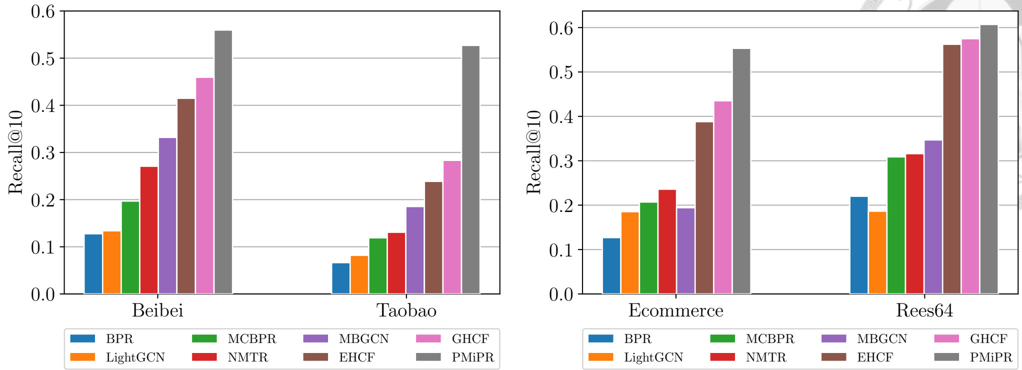


Figure 5.1: Overall performance comparison on purchase recommendation.

This is the typical recommendation task evaluated in most studies on multi-behavior recommender systems. Table 5.1 and Fig. 5.1 show the results on the four datasets; below are the findings.

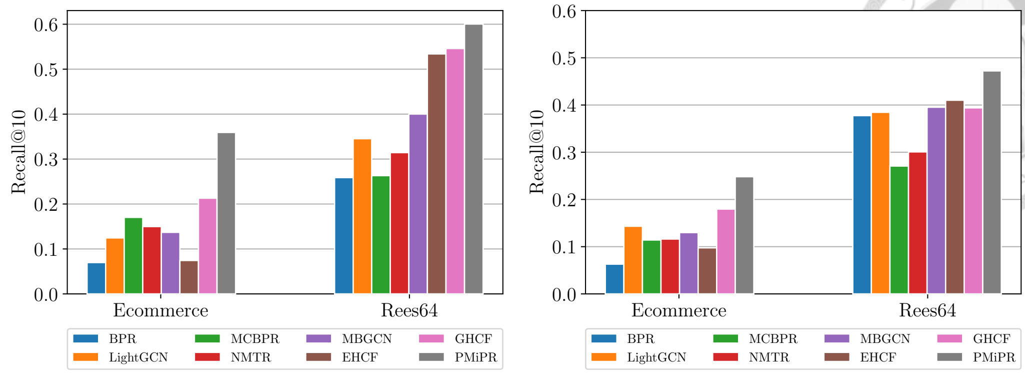
- PMiPR consistently outperforms all baselines over the four datasets except for Recall@10 and NDCG@10 on the Beibei dataset, which justifies the effectiveness of our model.

- All multi-behavior models outperform the two single-behavior models, which attests the effectiveness of leveraging multiple types of behavioral data for recommendation. This is consistent with previous findings [16, 17].

- GHCF is the strongest baseline of the compared methods. Nevertheless, the proposed PMiPR still yields significant improvements on all four datasets (e.g., ranging from 5.58% to 86.02% improvement in terms of Recall@50).

doi:10.6342/NTU202203790

5.1.2 Cart and View Recommendation



(a) cart

(b) view

Figure 5.2: Overall performance comparison on cart and view recommendation.

To verify the effectiveness of our model for recommendation w.r.t. different user behaviors, we further evaluated on *cart* and *view* recommendation; such tasks are however overlooked and thus not evaluated in most of the literature. We report the results conducted on the Ecommerce and Rees46 datasets, as shown in Tables 5.2 and 5.3, we also show the graphical results in Fig. 5.2, observed from which we itemize the findings as follows.

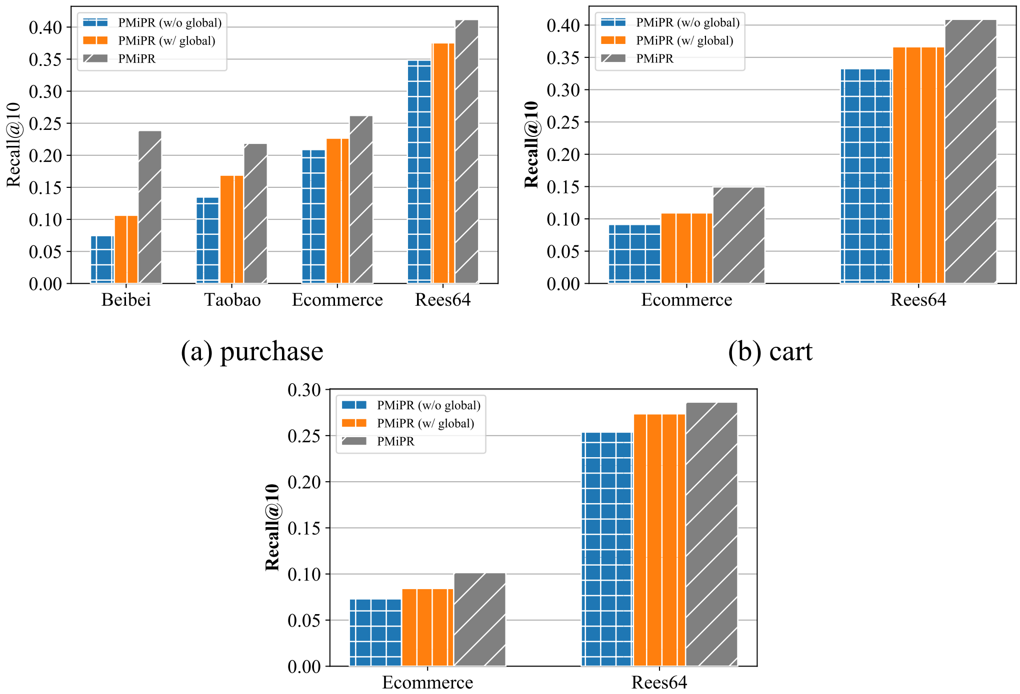
- PMiPR significantly outperforms all baselines over the two datasets in terms of all metrics, demonstrating that it better predicts not only the "target behavior" but also other behaviors compared to the state-of-the-art methods.
- GHCF still remains the strongest baseline, demonstrating the superiority of this state-of-the-art GNN-based model for multi-behavior recommendation.

Overall, PMiPR shows the effectiveness of leveraging interaction as training units and incorporating fine-grained behavior embeddings of users and items to learn a unified embedding matrix for multi-behavior recommendation. Multi-behavior prediction tasks are completed successfully with the proposed unified framework in a straightforward manner.

doi:10.6342/NTU202203790

These results demonstrate notable performance improvements over all three prediction tasks compared to state-of-the-art multi-behavior recommendation approaches.

5.2 Experiment 2: Ablation Studies on Global Behavior Embeddings



(a) purchase

(b) cart

(c) view

Figure 5.3: Ablation studies on global behavior embeddings.

To understand the impact of our mechanism for global behavior embeddings, we additionally considered two variants of the proposed model: PMiPR (w/o global) and PMiPR (w/ global), which disables and enables the global behavior r_g in $G(\mathcal{V}, \mathcal{E}, \psi)$, respectively (see Section 3.6). Moreover, both variants apply $\hat{y}^r_{ui} = (\theta_u + \theta^r_u) \cdot (\theta_i + \theta^r_i)$ as the scoring function to estimate the likelihood that a user u will interact with an item i under behavior r. (Note that the above score function is different from that used in the original PMiPR in Eq. (3.11), which additionally concatenates the global behavior embeddings when cal-

doi:10.6342/NTU202203790

culating the scores.)

Figure 5.3 illustrates the results of the three PMiPR versions: PMiPR (w/o global), PMiPR (w/ global), and PMiPR. As shown, adding global behavioral information indeed yields better recommendation performance for all types of recommendation tasks (see the bars representing the results of PMiPR (w/ global) and PMiPR). Moreover, PMiPR is shown to consistently outperform PMiPR (w/ global), which demonstrates that including global behavior embeddings in the scoring function further benefits performance and thus yields superior results.

5.3 Experiment 3: Sensitivity Analysis on Hyperparameters

In this section, we only report the hyperparameter analysis on the *purchase* prediction results; predictions on other behaviors exhibit similar phenomena. Figure 5.4 plots the effect of the L_2 regularization parameter λ_Θ, where all datasets perform relatively poorly when $\lambda_\Theta = 0$ and show the best performance when $\lambda_\Theta = 0.001$. In addition, we investigate the effect of the learning rate α, as shown in Fig. 5.5, and $\alpha = 0.025$ leads the best results for all datasets. Furthermore, we investigate the effect of number of negative sample n w.r.t each dataset, where the number of negative sample that lead the best result in each dataset is different due to different size of each dataset, as shown in Fig. 5.6.

doi:10.6342/NTU202203790

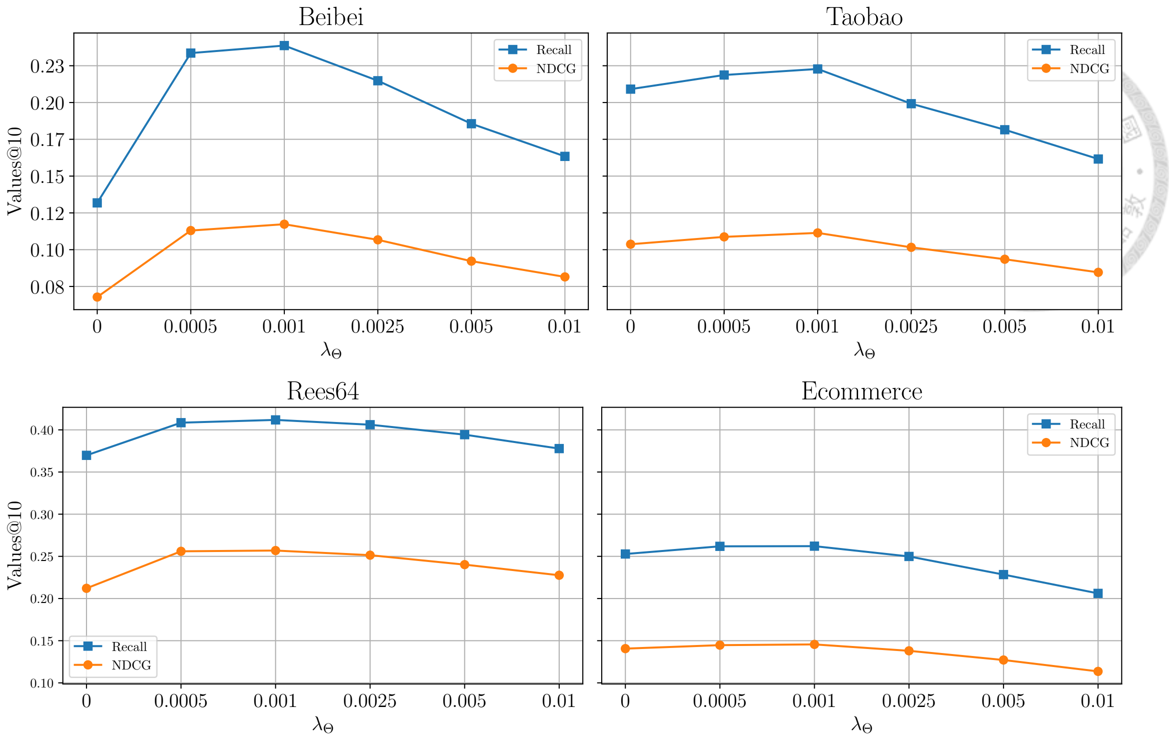


Figure 5.4: Sensitivity on Regularization.

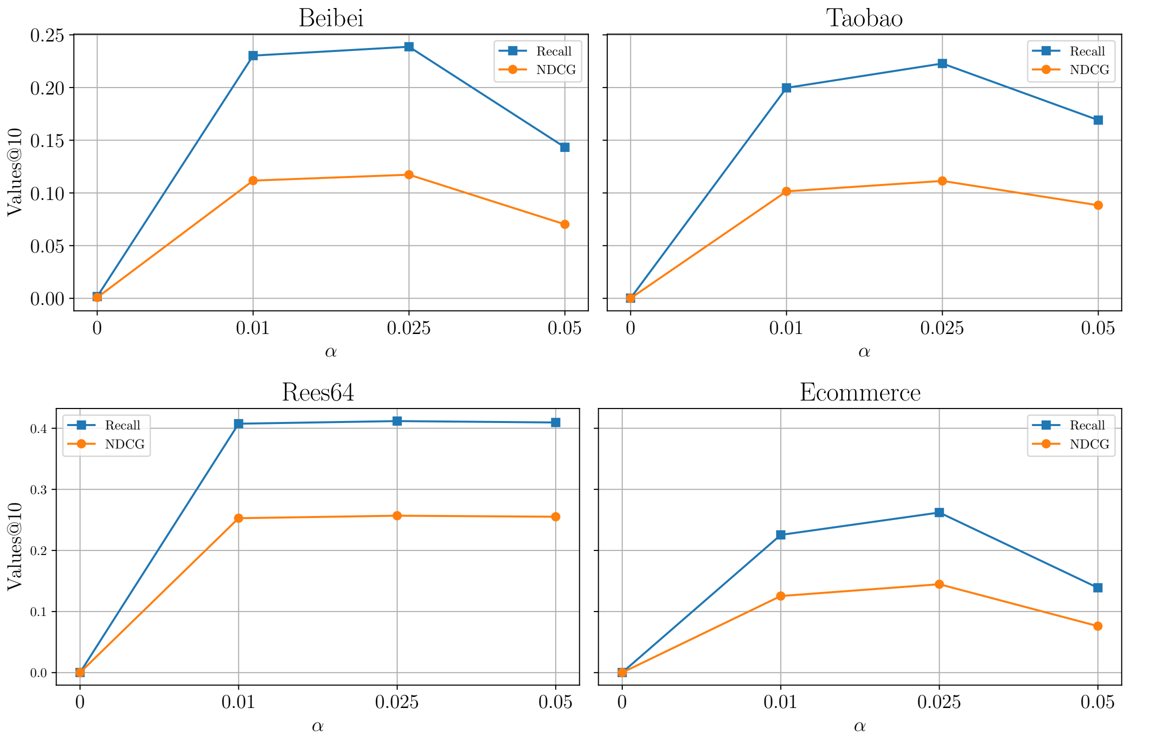


Figure 5.5: Sensitivity on Learning Rate.

doi:10.6342/NTU202203790

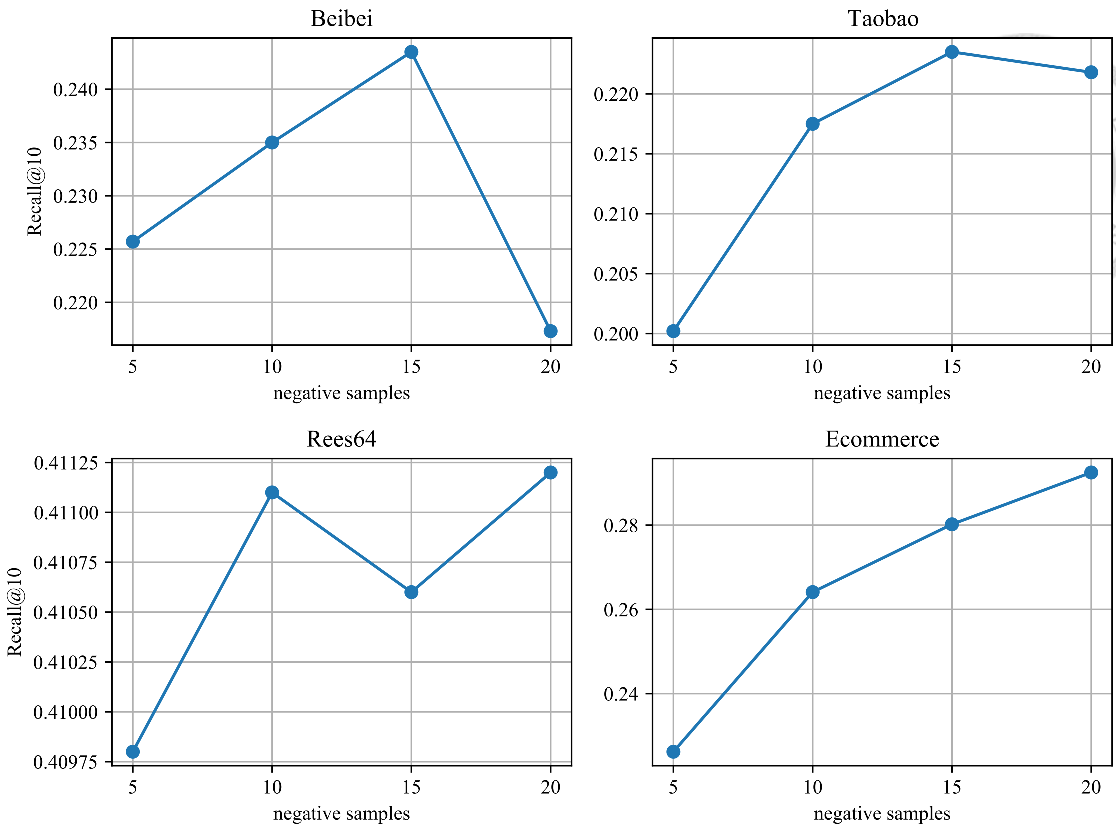


Figure 5.6: Sensitivity on Negative Sample.

5.4 Experiment 4: Computational Efficiency Comparison

We used Ecommerce, the largest dataset, to compare the computational efficiency among different models. Figure 5.7 plots the execution time of the proposed model and the seven compared methods.[1] As observed, the proposed PMiPR requires around 600 seconds to complete the whole training process, which is much faster than all models except BPR. Moreover, while BPR, MCBPR, and PMiPR use only CPU computations, other models use GPU computations. Such results demonstrate the lightweight nature and computational efficiency of the proposed embedding learning framework, which is thus more practical than other advanced methods.

[1] Values reported in the figure vary when different implementations are applied.

doi:10.6342/NTU202203790

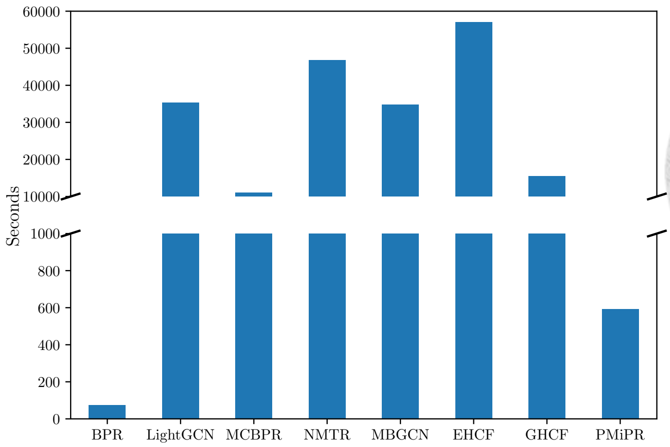


Figure 5.7: Execution Time Usage.

doi:10.6342/NTU202203790

Chapter 6

Conclusion and Future Work

We describe conclusion and future work of this research in Section 6.1 and Section 6.2 respectively.

6.1 Conclusion

In this paper, we propose personalized multi-interaction preference ranking (PMiPR), a unified interaction-based pairwise ranking embedding framework that incorporates both multi-behavior and global information for embedding learning. PMiPR samples and constructs interaction triples and then leverages a pairwise ranking algorithm to capture user and item preferences under each behavior based on interaction similarity. Its effectiveness and efficiency are demonstrated by extensive experiments and analysis.

6.2 Future Work

It is worth mentioning that although this paper follows most exiting multi-behavior literature to conduct experiments on e-commerce datasets only, we plan to investigate the performance on more diverse datasets, especially including the ones with much more

doi:10.6342/NTU202203790

types of user behaviors in the future work. Also, there are several possible future directions based on the proposed PMiPR framework. To name a few, first, we will explore the possibility of artificially defining different types of user-item interactions by leveraging the information from user profiles and item metadata. Such an attempt may further refine the behaviors and thus make the model better accommodate different usage scenarios. Second, as the so-called "positive" interaction and "negative" interaction in a triplet can be freely defined in our framework, we prepare to investigate other sampling strategies for constructing positive and negative interactions for model training. Finally, how to combine the proposed interaction-level modeling with knowledge graphs for the task of recommendation is also an interesting future research direction.

doi:10.6342/NTU202203790

Bibliography

[1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Springer, 2011, pp. 1–35.

[2] W. Pan, M. Liu, and Z. Ming, "Transfer learning for heterogeneous one-class collaborative filtering," in *Proceedings of the 8th IEEE Intelligent Systems*, 2016, pp. 43–49.

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, 2001, pp. 285–295.

[4] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proceedings of the 8th IEEE International Conference on Data Mining*, 2008, pp. 263–272.

[5] S. Kabbur, X. Ning, and G. Karypis, "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*, 2013, pp. 659–667.

[6] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 974–983.

doi:10.6342/NTU202203790

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[8] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[9] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "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*, 2020, pp. 639–648.

[10] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T.-S. Chua, "Disentangled graph collaborative filtering," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1001–1010.

[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 *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 2397–2406.

[12] L. Xia, C. Huang, Y. Xu, P. Dai, X. Zhang, H. Yang, J. Pei, and L. Bo, "Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021, pp. 4486–4493.

[13] B. Jin, C. Gao, X. He, D. Jin, and Y. Li, "Multi-behavior recommendation with graph convolutional networks," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 659–668.

doi:10.6342/NTU202203790

[14] B. Loni, R. Pagano, M. Larson, and A. Hanjalic, "Bayesian personalized ranking with multi-channel user feedback," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 361–364.

[15] H. Qiu, Y. Liu, G. Guo, Z. Sun, J. Zhang, and H. T. Nguyen, "Bprh: Bayesian personalized ranking for heterogeneous implicit feedback," in *Information Sciences*, vol. 453, 2018, pp. 80–98.

[16] C. Chen, M. Zhang, Y. Zhang, W. Ma, Y. Liu, and S. Ma, "Efficient heterogeneous collaborative filtering without negative sampling for recommendation," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020, pp. 19–26.

[17] C. Chen, W. Ma, M. Zhang, Z. Wang, X. He, C. Wang, Y. Liu, and S. Ma, "Graph heterogeneous multi-relational recommendation," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021, pp. 3958–3966.

[18] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2008, pp. 426–434.

[19] R. He and J. McAuley, "Vbpr: visual bayesian personalized ranking from implicit feedback," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016, pp. 144–150.

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

[21] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems." in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 3203–3209.

doi:10.6342/NTU202203790

[22] Z.-H. Deng, L. Huang, C.-D. Wang, J.-H. Lai, and S. Y. Philip, "Deepcf: A unified framework of representation learning and matching function learning in recommender system," in *Proceedings of the 33th AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 61–68.

[23] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 335–344.

[24] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 165–174.

[25] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using bayesian personalized ranking for social network data," in *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, 2012, pp. 173–182.

[26] L. Tang, B. Long, B.-C. Chen, and D. Agarwal, "An empirical study on recommendation with multiple types of feedback," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 283–292.

[27] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 650–658.

doi:10.6342/NTU202203790

[28] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi, "Improving user topic interest profiles by behavior factorization," in *Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 1406–1416.

[29] J. Ding, G. Yu, X. He, Y. Quan, Y. Li, T.-S. Chua, D. Jin, and J. Yu, "Improving implicit recommender systems with view data," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3343–3349.

[30] 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 *Proceedings of the 35th IEEE International Conference on Data Engineering*, 2019, pp. 1554–1557.

[31] W. Wei, C. Huang, L. Xia, Y. Xu, J. Zhao, and D. Yin, "Contrastive meta learning with behavior multiplicity for recommendation," in *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, 2022, pp. 1120–1128.

[32] J.-H. Yang, C.-M. Chen, C.-J. Wang, and M.-F. Tsai, "Hop-rec: High-order proximity for implicit recommendation," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 140–144.

[33] C.-M. Chen, C.-J. Wang, M.-F. Tsai, and Y.-H. Yang, "Collaborative similarity embedding for recommender systems," in *In Proceedings of the 28th International Conference on World Wide Web*, 2019, pp. 2637–2643.

[34] S.-Y. Liu, H. H. Chen, C.-M. Chen, M.-F. Tsai, and C.-J. Wang, "Ipr: Interaction-level preference ranking for explicit feedback," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1912–1916.

[35] Y.-N. Chuang, C.-M. Chen, C.-J. Wang, M.-F. Tsai, Y. Fang, and E.-P. Lim, "Tpr: Text-aware preference ranking for recommender systems," in *Proceedings of the*

doi:10.6342/NTU202203790

29th ACM International Conference on Information and Knowledge Management, 2020, pp. 215–224.

[36] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild!: A lock-free approach to parallelizing stochastic gradient descent," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, 2011, pp. 693–701.

[37] W. Krichene and S. Rendle, "On sampled metrics for item recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 1748–1757.

[38] 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.

doi:10.6342/NTU202203790