

多行为超图增强序列推荐系统

汇报

演讲人 林思灵 日期 2023.5.12

目录

01

简介

02

MBHT

03

实验

04

总结

01

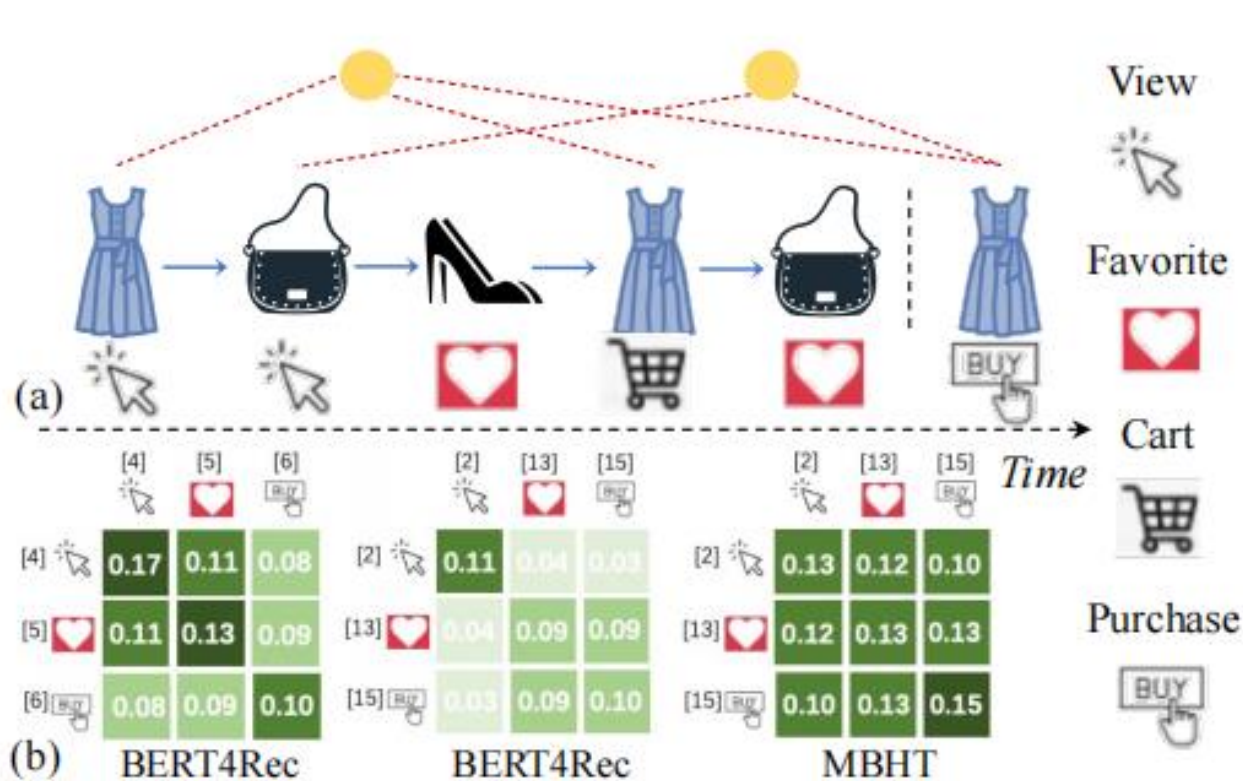
简介

序列推荐

- 旨在学习动态用户偏好的序列推荐任务已经成为许多在线平台（如视频分享网站、电子商务系统）的一个重要的组成部分。已有的模型基于各种架构，例如，循环神经网络（GRU4Rec）和自注意力机制（SASRec）。在这一任务最新的进展中，图神经网络（GNN）也被越来越多研究者作为骨干模型。
- 然而现有的方法都集中在单一类型的互动的商品序列表示上，因此对于捕捉用户和商品之间在序列上表现出的动态异质关系结构（例如，点击、页面浏览、收藏、购买）是有限的。在实际的在线平台上，用户行为往往同时表现出时间依赖性和交互类型的异质性，涉及不同类型的用户-商品互动。

序列推荐

- 图 (a) 中展示了一个 Tmall 数据集的例子, 这一用户在时间序列上以不同的方式交互了不同的商品。我们还在 (b) 中可视化了 SOTA 模型 BERT4Rec 和 MBHT 模型所学到的相邻的{ [4], [5], [6] }之间的短期项目相关性和{ [2], [13], [15] }之间的长期依赖性的行为感知依赖性权重。
- 在这个场景中, 我们更希望模型关注到用户在序列上对于同一商品的不同交互行为。容易发现, 相比于捕捉短期多行为语义依赖的容易, BERT4Rec 很难注意到跨度较长的有效语义依赖。而我们提出的 MBHT 模型则更好地建模了这一重要的语义关系。



挑战

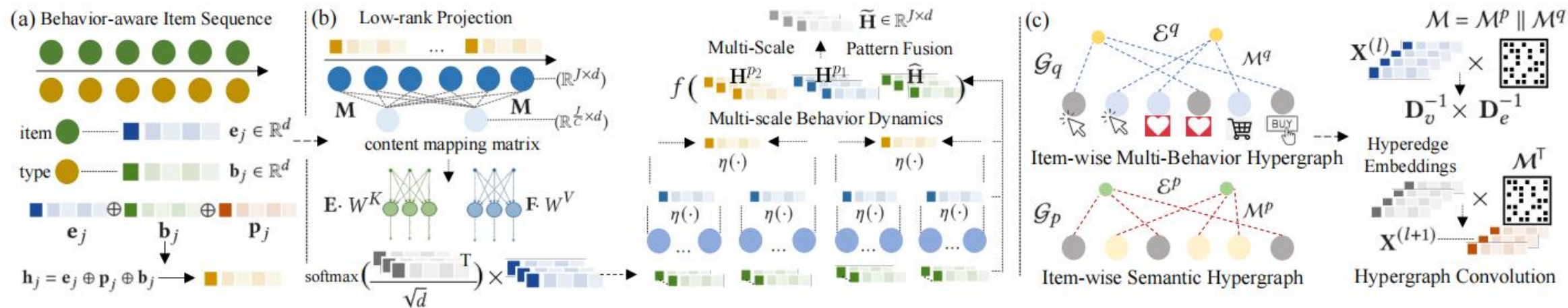
- 总的来说，通过对动态序列中的异质用户-商品互动的探索，有效地增强用户偏好学习，是在多行为序列推荐任务下做出准确的排序的关键。然而，我们的挑战存在于：
 1. 动态的、行为感知的商品转移关系。对于不同类别的物品（如日常用品、季节性服装），存在不同的周期性行为模式（如每天、每周、每月）。因此，有必要显式地捕捉行为感知的商品转移关系在不同时间粒度上反映的用户兴趣。
 2. 用户个性化的全局多行为交互。不同类型的行为之间的隐性依赖关系因用户而异。例如，对于感兴趣的商品，一些用户希望将商品加到收藏夹中，而另一些人只会将那些非常可能购买的商品加入收藏。因此，所设计的模型需要对不同用户的多行为依赖关系进行有针对性的建模。
- 因此，为了应对这一挑战，我们设计了一个多行为超图增强的 Transformer 框架（MBHT），以在交互序列中捕捉短期和长期的跨行为依赖。基于不同范式的低秩注意力，我们设计了一个多尺度的 Transformer 以联合编码来自细粒度和粗粒度的行为感知的序列用户兴趣。此外，我们将全局的多行为依赖性建模纳入到超图神经网络中，以个性化的方式捕捉不同用户的长程行为关联。实验结果表明，我们的 MBHT 在不同的环境中比各种先进的推荐解决方案更有优势。

02

MBHT

MBHT模型流

- (a)多尺度行为感知上下文嵌入层
- (b)多尺度transformer架构
- (c)超图神经结构



多尺度行为感知上下文嵌入层

多行为

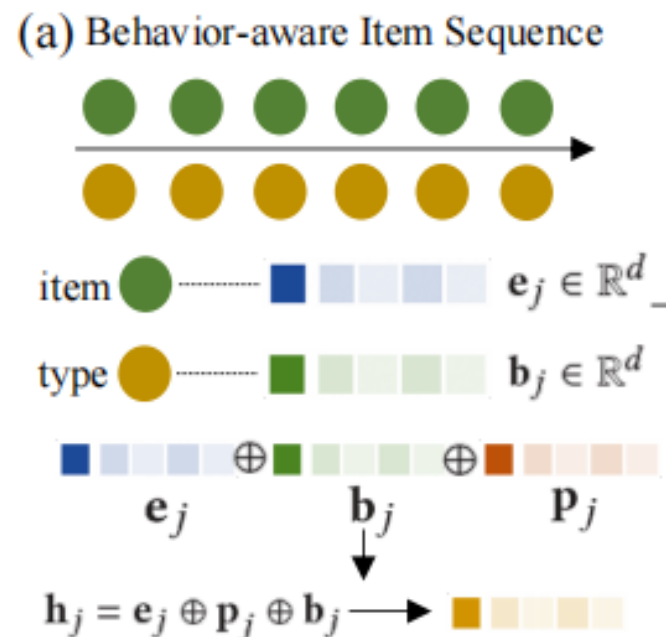
为了使得序列编码器能够建模多行为的特性，将行为感知的交互上下文注入到我们的顺序学习框架中，我们在商品项目嵌入的基础上加入了行为类别嵌入，以联合编码单个项目信息和相应的交互行为上下文信号。给定一个项目 v_j ，执行以下操作得到它的行为感知潜在表示 $\mathbf{h}_j \in \mathbb{R}^d$

$$\mathbf{h}_j = \mathbf{e}_j \oplus \mathbf{p}_j \oplus \mathbf{b}_j$$

其中 \mathbf{e}_j 表示初始化的项目嵌入， \mathbf{p}_j 表示序列的位置编码，
 \mathbf{b}_j 表示用户 u_i 和项目 v_j 之间的交互的行为类型编码。

$\mathbf{e}_j, \mathbf{p}_j, \mathbf{b}_j \in \mathbb{R}^d$

用户 u_i 的行为感知交互项目序列 S_i 的项目表示矩阵 $\mathbf{H} \in \mathbb{R}^{J \times d}$ 。



基于 Transformer 架构的多尺度序列兴趣编码器

在实际的推荐场景中，用户与商品的交互偏好可能会随着时间的推移呈现出多尺度的模式。这一特性不仅反映在周期性的交互模式中（如，季节性时装），更由不同用户的个性化多行为交互模式所表现（如，一些用户在收藏某件商品后会很快购买，而另一些用户则会在更多的交互后决定）。因此，我们设计了一个基于 Transformer 架构的多尺度序列兴趣编码器。

Transformer模型在许多自然语言处理应用程序中取得了非常成功的成果。transform模型主要分为两个部分，分别是编码器和解码器。编码器负责把自然语言序列映射称为隐藏层（含有自然语言序列的数学表达），然后解码器把隐藏层在映射为自然语言序列。transformer由self-attention和feedforword neural network组成，基础的transformer的attention layer的结构：

$$f(Q, K, V) = \text{softmax}\left(\frac{QW_Q \cdot (KW_K)^T}{\sqrt{d}}\right)VW_V$$

针对self-attention的情况，我们假设句长为 n ，embedding维度为 d ，则 $Q, K, V \in \mathbb{R}^{n \times d}$ ， $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ 。

因此，当 $n \gg d$ 时，上述的attention层的算法复杂度就是 $O(n^2)$ 。

但对于长序列来说，训练和部署这些模型的成本很高。如何优化以避免这种二次运算？

低秩投影自注意力机制

Linformer: Self-attention with linear complexity

本文从理论和经验上证明了自注意机制所形成的随机矩阵可以近似为低秩矩阵。

秩就是矩阵的行列之间的相关性。低秩矩阵每行或每列都可以用其他的行或列线性表出，包含大量的冗余信息。由于高维数据可由数据潜在的低维特征组合得出，通过寻求关联矩阵来提取数据的潜在特征，从而把数据从高维空间嵌入或投影到低维空间。

思路：将attention投影到一个低阶矩阵当中，在计算Key和Value时分别加入一个线性投影矩阵，使得n阶的attention降维到固定的k维矩阵。

$$f(Q, K, V) = \text{softmax}\left(\frac{QW_Q \cdot (E \cdot KW_K)^T}{\sqrt{d}}\right)(F \cdot VW_V)$$

其中, $E, F \in \mathbb{R}^{k \times n}$, k 是一个常数。

由此，我们就可以将 $O(n^2)$ 的计算复杂度降维到 $O(kn)$ ，其中 k 不过是一个常数而已。

低秩自注意模块

我们定义两个线性投影矩阵E、F，在低秩投影空间中的注意力操作可以表示为：

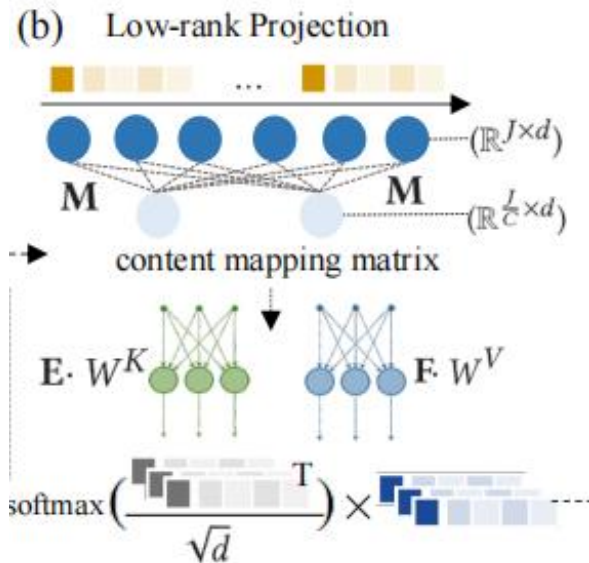
$$\hat{H} = \text{softmax}\left(\frac{H \cdot W^Q (E \cdot H \cdot W^K)^T}{\sqrt{d}}\right) \cdot F \cdot H \cdot W^V$$

E, F ∈ ℝ J/C × J, C表示低秩尺度, J表示输入的交互序列的长度, J/C表示输入的交互序列在低秩隐空间的长度。

在我们的低秩自注意模块中, E和F被用来将ℝ J × d维Key和Value转换表示 H · W^K and H · W^V投射到 (ℝ J/C × d)维潜在低秩嵌入。Ĥ ∈ ℝ J/C × d

综上所述, 通过对原始注意操作的低秩因子分解, 我们计算了维数 RJ × J/C的上下文映射矩阵M, 计算成本从O (J × J) 显著降低到O (J × J/C)

- 1) 降低了单注意力层的运算时间, 使得我们能够后续叠加更加复杂的序列建模操作;
- 2) 将原始长度 J 的用户序列抽象为 J/C 的低秩序列, 能够更好地在捕捉用户反映在交互序列上的兴趣中更为抽象、全局性的部分。



多尺度行为动态

进一步地，我们在低秩注意力的基础上引入一个层次化的结构来捕捉不同时间粒度的用户行为动态多样性。我们提出一个时间粒度感知的聚合器来生成特定粒度的序列表示 \mathbf{g}_p ，它保留了短期行为中的用户兴趣。其中 p 被定义为子序列的长度。我们使用聚合表示 $\mathbf{\Gamma}_p \in \mathbb{R}^{J/p \times d}$ 和 $\boldsymbol{\gamma} \in \mathbb{R}^d$ 生成时间粒度感知的序列 embedding:

$$\mathbf{\Gamma}_p = \{\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_{\frac{J}{p}}\} = [\eta(\mathbf{h}_1, \dots, \mathbf{h}_p); \dots; \eta(\mathbf{h}_{J-p+1}, \dots, \mathbf{h}_J)]$$

\mathbf{h}_j 为项目 v_j 的行为感知潜在表示， $\eta(\cdot)$ 表示为了捕捉短期兴趣所在连续的一段 item embedding 上作用的聚合器，在模型实现中，利用平均池化来执行嵌入聚合。
接下来，我们将时间粒度感知的序列表示送入一个自注意力层来进行编码：

$$\mathbf{H}_p = \text{softmax}\left(\frac{\mathbf{\Gamma}_p \cdot \mathbf{W}_p^Q (\mathbf{\Gamma}_p \cdot \mathbf{W}_p^K)^T}{\sqrt{d}}\right) \cdot \mathbf{\Gamma}_p \cdot \mathbf{W}_p^V$$

其中 $\mathbf{H}_p \in \mathbb{R}^{J/p \times d}$ 。注意到该计算的形式与上面提到的低秩注意力操作相似，而不同的是 query 向量也进行了聚合操作，所以该注意力计算在 $J/p \times J/p$ 空间中进行，在很好地捕捉用户短期兴趣变化的同时，也节省了计算的时间复杂度。

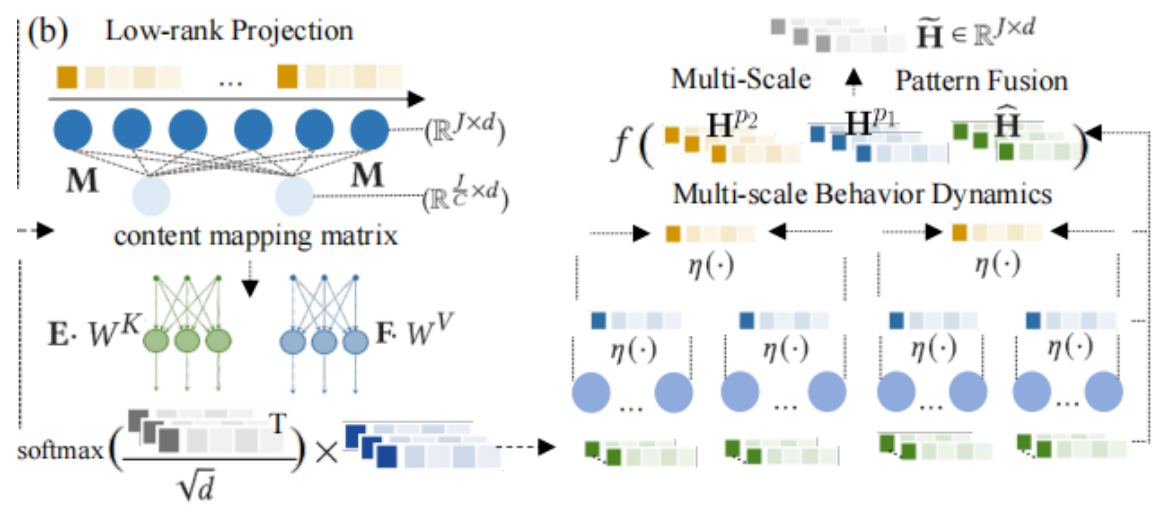
层次化

在 MBHT 模型中，我们将超参数为 C 的低秩注意力和两个不同粒度 p_1 和 p_2 的时间粒度感知注意力机制进行堆叠，引入一个层级化的注意力操作，它可以表达为：

$$\tilde{\mathbf{H}} = f(\hat{\mathbf{H}} \parallel \mathbf{H}^{p_1} \parallel \mathbf{H}^{p_2})$$

其中 $\mathbf{H} \in \mathbb{R}^{J/C \times d}$, $\mathbf{H}^{p_1} \in \mathbb{R}^{J/p_1 \times d}$, $\mathbf{H}^{p_2} \in \mathbb{R}^{J/p_2 \times d}$ 。其中 $f(\cdot)$ 表示进行 $\mathbb{R}^{(\frac{J}{C} + \frac{J}{p_1} + \frac{J}{p_2}) \times d} \rightarrow \mathbb{R}^{J \times d}$

的映射函数，是一个可学习的权重，它将层次化的序列表示还原到原始的序列长度空间中。注意到低秩注意力与时间粒度感知的注意力不仅仅能够从有用的视角更好地捕捉序列中的用户兴趣，同时这一层次化的计算并没有带来不可接受的额外时间开销。



超图构造

对于用户个性化的、长程的多行为依赖，准确的建模有以下两个挑战：

- 多阶的多行为依赖性。当决定向用户推荐一个特定的商品时，探索用户和这个商品之间过去的多行为互动是很有用的。但这样的关系不再是简单的 **user-item** 一阶交互关系，而是引入了更加复杂的多行为信号。
- 个性化的交互模式。用户的多行为模式因人而异。在现实生活中，一些用户喜欢将许多商品添加到他们的最爱列表或购物车中，但以后只会购买其中的几件。相反，另一些用户只有在表现出强烈的购买意愿时才会将他们感兴趣的产品标记为收藏。因此，这种复杂的、个性化的多行为模式需要我们的模型来保留不同的跨类型行为依赖。

因此，我们选在超图结构上建立全局多行为依赖性编码。超图具有通过一条边连接任意个节点的灵活性，因此可以利用它结构来捕捉高阶的多行为依赖关系。此外，考虑到不同用户的行为意识交互序列，我们根据序列构建不同的超图结构，以个性化的方式编码不同用户的多行为依赖。

项目级语义依赖超图

为了编码商品语义依赖信息，我们引入了一个基于度量学习的商品语义编码器，设计了一个具有多头权重函数 $\tau(\cdot)$ 的项目之间的可学习度量 β^{nj}, j' ：

$$\beta_{j,j'} = \frac{1}{N} \sum_{n=1}^N \hat{\beta}_{j,j'}^n; \mathbf{v}_j = \mathbf{e}_j \oplus \mathbf{b}_j$$
$$\hat{\beta}_{j,j'}^n = \tau(\mathbf{w}^n \odot \mathbf{v}_j, \mathbf{w}^n \odot \mathbf{v}_{j'})$$

其中 \mathbf{e}_j 表示初始化的商品嵌入, \mathbf{b}_j 此次交互的行为类型编码。商品 j,j' 间的语义相似度由一个多头的可学习权重来表示， \mathbf{w}^n 是特定学习头的度量权重， $\beta^{nj,j'}$ 表示 \mathbf{v}_j 和 $\mathbf{v}_{j'}$ 之间可学习的特定学习头依赖权重。（ $\mathbf{w}^n \odot \mathbf{v}_j$ ）表示嵌入投影操作。权值函数 $\tau(\cdot)$ 为余弦相似度。估计将平均池化操作应用于所有学习到的特定学习头项目语义依赖得分（ $\beta^{nj,j'}$ ），以获得商品 \mathbf{v}_j 和 $\mathbf{v}_{j'}$ 之间的最终相关性 $\mathbf{B}_{j,j'}$ 。

通过对不同商品之间的语义依赖性进行编码，我们用超边同时连接多个高度依赖的商品来生成全局的语义依赖超图。具体地说，我们为序列中特定的 `item_id` 构建一组超边，超边连接着商品的 `top-k` 语义依赖商品，超边的权重被设定为它们之间语义依赖相似度的可求导值：

$\mathbf{M} \in \mathbb{R}^J \times |\mathbf{E}_p|$ 为语义依赖超图的连接矩阵， $|\mathbf{E}_p|$ 对应于序列唯一项的数量。

$$m^p(v_j, \epsilon_{j'}) = \begin{cases} \beta_{j,j'} & v_{j'} \in A_j; \\ 0 & otherwise; \end{cases}$$

项目级多行为依赖超图

根据训练集中已观测到的多行为交互记录，我们构建超图结构，将同一 **item** 的不同行为使用不同语义的超边连接：

$$m^q(v_j^b, \epsilon_j^{b'}) = \begin{cases} 1 & v_j^b \in \mathcal{E}_j^q; \\ 0 & \text{otherwise;} \end{cases}$$

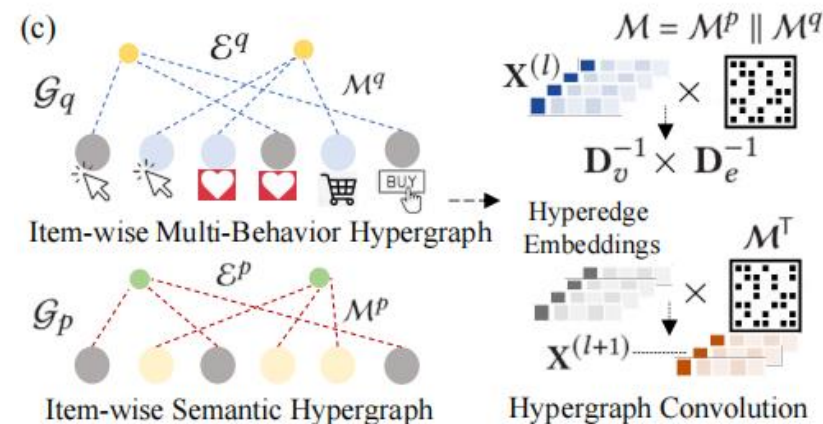
v_j^b 表示商品j在第b个行为类型下与用户 u_i 交互。 \mathcal{E}_j^q 表示多类型 u_i-v_j 相互作用的集合。
 $M^q \in \mathbb{R}^J \times |\mathcal{E}^q|$ 为多行为依赖超图的连接矩阵， $|\mathcal{E}^q|$ 对应于在超图 G^q 中超边的个数。

综上，我们将得到的多行为依赖超图与商品语义依赖超图进行拼接，即 $M = M^p \parallel M^q$ 作为最终的超图矩阵。注意到，不同用户的行为模式导致了不同的子图，这允许了我们的 **MBHT** 模型以个性化的方式建模用户的多行为依赖。

超图卷积

之后，我们引入了一个通用的超图卷积模块来将 **item embedding** 按照超图矩阵进行编码。在这个模块中，我们引入了带有卷积层的超图消息传递范式，以捕获随时间变化的全局多行为依赖关系。超图卷积层通常涉及传递的两阶段信息，即节点-超边和超边-节点嵌入传播，以及超图连接矩阵 \mathbf{M} ，用于细化项表示。特别地，我们将超图卷积层设计为：

$$\mathbf{X}^{(l+1)} = \mathbf{D}_v^{-1} \cdot \mathbf{M} \cdot \mathbf{D}_e^{-1} \cdot \mathbf{M}^T \cdot \mathbf{X}^{(l)}$$



其中， $\mathbf{X}^{(l)}$ 表示从超图卷积的第 l 层编码的项目嵌入。此外， \mathbf{D}_v 和 \mathbf{D}_e 分别是基于顶点度和边度进行归一化的对角矩阵。

注意到两阶段的信息传播比起经典的 **GNN** 模型要花费额外的时间，由 $\mathbf{M} \cdot \mathbf{M}^T$ 传递的两阶段消息需要 O

$((|\mathbf{E}_p| + |\mathbf{E}_q|) \times J^2)$ 的计算，这是相当耗时的。为了加速模型的训练和推理，我们利用前置知识来逼近两阶段的信息传递过程以降低时间复杂度。

Self-supervised multi-channel hypergraph convolutional network for social recommendation.

多视图学习信号融合

我们进一步提出一个简单的多视图注意力机制来融合 Transformer 和超图模块学习到的丰富信息。我们开发了一个注意层来学习对于特定视图的项目嵌入的显式重要性。这一过程可以形式化地表达为：

$$\alpha_i = \text{Attn}(\mathbf{e}_i) = \frac{\exp(\mathbf{a}^\top \cdot \mathbf{W}_a \mathbf{e}_i)}{\sum_i \exp(\mathbf{a}^\top \cdot \mathbf{W}_a \mathbf{e}_i)}$$
$$\mathbf{e}_i \in \{\tilde{\mathbf{h}}_i, \tilde{\mathbf{x}}_i\}; \mathbf{g}_i = \alpha_1 \cdot \tilde{\mathbf{h}}_i \oplus \alpha_2 \cdot \tilde{\mathbf{x}}_i$$

其中 \mathbf{h}_i 、 \mathbf{x}_i 分别是来自两个学习视图的商品 embedding 编码。 \mathbf{a} 、 \mathbf{W}_a 为可训练的参数。

最后序列中第 i 项为项目 \mathbf{v}_j 的概率估计为 $\hat{y}_{i,j} = \mathbf{g}_i^\top \mathbf{v}_j$,

模型学习

为了更好地利用丰富的序列信息，我们引入 **Cloze task** 作为训练目标来挖掘序列中的双向信号。

Cloze task 字面意思是完形填空，通过mask掉一些items，然后根据其 **context** 预测这些被mask的item。这样，我们避免了 **information leakage**，并通过允许输入序列中每个项目的表示融合左和右上下文来学习双向表示模型。

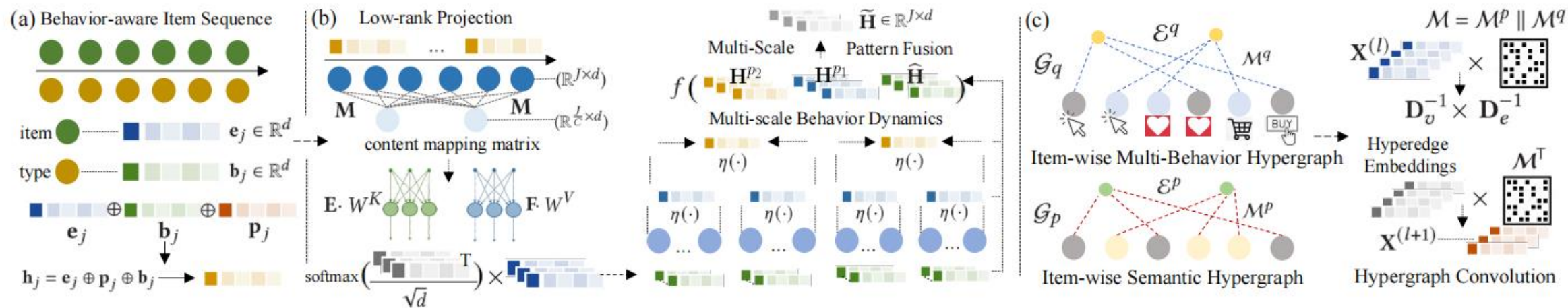
在我们的多行为序列任务中，**Cloze task** 可以被描述为：对序列中的所有预测目标行为类型（如购买）进行mask操作。我们用 [mask] 标记取代被屏蔽的商品 id 以及相应的行为类型，并在超图构建时同样屏蔽标记为 [mask] 的商品。

对于 [mask] 商品的超图 embedding，我们引入一个窗口机制，将该商品在序列上的周围位置内的商品 embedding 进行 mean pooling 后作为该商品的超图 embedding 编码。最后，我们基于概率估计来预测用户序列中的下一个购买商品。通过对 mask 位置的预测结果计算交叉熵，我们用以下的损失函数来优化模型参数：其中 M 是序列中 mask 位置的集合， T 是对应于这些 mask 位置的真实商品 ID 标签。

$$\mathcal{L} = \frac{1}{|T|} \sum_{t \in T, m \in M} -\log\left(\frac{\exp \hat{y}_{m,t}}{\sum_{j \in V} \exp \hat{y}_{m,j}}\right)$$

MBHT模型流

- (a)多尺度行为感知上下文嵌入层
- (b)多尺度transformer架构
- (c)使用超图神经结构捕获全局和个性化的多行为依赖学习。



03

实验

实验设置

以上表格的实验结果表明，我们提出的 MBHT 在针对不同数据集（淘宝、Retailrocket、IJCAI）、不同指标（HR@5、10；NDCG@5、10；MRR）以及不同 research line 丰富的 baseline 模型（通用序列推荐、基于图的序列推荐和多行为推荐）时都表现出了目前最好的效果。另外，加入多行为信号通常能够直接提升模型的表现（如 BERT4Rec-MB 对比 BERT4Rec），但如何更好地利用复杂的多行为信号建模更加丰富的用户兴趣隐向量表示，是研究的关键点之一。

Model	Taobao					Retailrocket					IJCAI				
	HR@5	NDCG@5	HR@10	NDCG@10	MRR	HR@5	NDCG@5	HR@10	NDCG@10	MRR	HR@5	NDCG@5	HR@10	NDCG@10	MRR
General Sequential Recommendation Methods															
Caser	0.082	0.058	0.123	0.071	0.070	0.632	0.539	0.754	0.578	0.535	0.134	0.092	0.167	0.104	0.109
HPMN	0.162	0.130	0.219	0.141	0.139	0.664	0.633	0.711	0.587	0.602	0.144	0.085	0.197	0.124	0.123
GRU4Rec	0.147	0.105	0.209	0.125	0.118	0.640	0.575	0.708	0.597	0.572	0.141	0.100	0.200	0.119	0.113
SASRec	0.150	0.110	0.206	0.128	0.123	0.669	0.644	0.689	0.650	0.645	0.146	0.110	0.191	0.124	0.122
BERT4Rec	0.198	0.153	0.254	0.171	0.163	0.808	0.670	0.881	0.694	0.639	<u>0.297</u>	<u>0.220</u>	<u>0.402</u>	<u>0.253</u>	<u>0.227</u>
Graph-based Sequential Recommender Systems															
SR-GNN	0.102	0.071	0.153	0.087	0.086	0.848	0.780	0.891	0.793	0.767	0.072	0.048	0.118	0.062	0.064
GCSAN	<u>0.217</u>	0.160	0.305	<u>0.188</u>	0.173	0.872	0.846	0.890	0.851	0.842	0.119	0.086	0.175	0.104	0.101
HyperRec	0.145	0.130	0.224	0.133	0.129	0.860	0.705	0.833	0.820	0.816	0.140	0.109	0.236	0.144	0.132
SURGE	0.122	0.078	0.193	0.100	0.093	<u>0.878</u>	<u>0.879</u>	<u>0.906</u>	<u>0.887</u>	<u>0.870</u>	0.226	0.159	0.322	0.190	0.171
Multi-Behavior Recommendation Models															
BERT4Rec-MB	0.211	<u>0.169</u>	0.263	0.186	<u>0.178</u>	0.875	0.858	0.889	0.863	0.857	0.257	0.189	0.342	0.216	0.197
MB-GCN	0.185	0.103	0.309	0.143	0.149	0.844	0.735	0.878	0.752	0.739	0.218	0.145	0.335	0.182	0.177
NMTR	0.125	0.082	0.174	0.097	0.103	0.827	0.697	0.858	0.724	0.741	0.109	0.076	0.184	0.099	0.106
MB-GMN	0.196	0.115	<u>0.319</u>	0.154	0.151	0.853	0.762	0.901	0.830	0.822	0.235	0.161	0.337	0.193	0.176
MBHT	0.323*	0.257*	0.405*	0.283*	0.262*	0.931*	0.933*	0.956*	0.950*	0.929*	0.346*	0.268*	0.437*	0.297*	0.272*
# Improve	48.84%	52.07%	26.95%	50.53%	47.19%	6.04%	6.14%	5.52%	7.10%	6.78%	16.50%	21.82%	8.71%	17.39%	19.82%

消融实验

- MB-Hyper: 消除模型超图编码中的多行为超图部分
- ML-Hyper: 消除模型超图编码中基于 Metric Learning 的商品语义依赖超图部分
- Hypergraph: 移除模型中全部的超图学习视图
- MS-Attention: 将模型中提出的多尺度自注意力机制替换为 Vanilla Attention 机制

Model Variants	Taobao		Retailrocket		IJCAI	
	HR@5	NDCG@5	HR@5	NDCG@5	HR@5	NDCG@5
MBHT	0.323	0.257	0.956	0.950	0.346	0.268
(-) <i>MB-Hyper</i>	0.261	0.206	0.883	0.861	0.320	0.249
(-) <i>ML-Hyper</i>	0.271	0.212	0.898	0.874	0.328	0.256
(-) <i>Hypergraph</i>	0.246	0.194	0.813	0.839	0.301	0.234
(-) <i>MS-Attention</i>	0.253	0.200	0.816	0.832	0.329	0.256

在 MBHT 中设计的每个关键组件都起到了提升推荐表现的作用。
超图学习模块对于更长序列的数据集上的预测起到了更大的作用

04

总结

总结

在本文中，我们提出了一个新的序列推荐框架 MBHT 来解决序列推荐问题中的多行为问题。MBHT 期望以显式的方式捕捉短期和长期的多行为依赖。为此，我们设计了一个多尺度的 Transformer 结构来编码细粒度和粗粒度的行为感知序列模式。为了捕捉全局性的跨类型行为依赖，我们为 MBHT 配备了一个多行为超图学习组件。在多样化的真实世界的数据集上的实证研究结果验证了我们的 MBHT 在与最先进的推荐方法比较时的各方面优势。

谢谢
