

Beyond Clicks: Modeling Multi-Relational Item Graph for Session-Based Target Behavior Prediction

Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, Hongyuan Zha

摘要

作者提出了一种用于基于会话的目标行为预测的新型多关系图神经网络模型,简称 MGNN-Spred。基于会话的目标行为预测任务是为了预测出下一个进行特定行为类型(例如点击)交互的物品。现有的方法大多利用表示学习的方法编码物品,然后去捕捉不同物品之间的关系。但之前的工作专注于一种行为类型的建模预测,本工作引入其他行为作为辅助行为,以提升模型预测能力。作者构建了使用所有会话的所有行为序列构建多关系项目图 (MRIG),涵盖了目标和辅助行为类型。基于 MRIG, MGNN-SPred 学习全局物品到物品之间的关系并进一步获得用户偏好。利用目标和辅助行为序列分别构建当前用户对不同行为的偏好表示。^[1]最后,使用两个真实世界数据集进行大量实验证明了本工作的优越性。

关键词: 序列推荐; 图神经网络; 用户行为建模

1 引言

推荐系统旨在从海量的候选物品中有效地为每个用户展示其最可能感兴趣的子集。

本工作针对推荐系统中的会话推荐问题。会话推荐指在一次访问会话里,基于一个用户的少数交互记录预测其最可能感兴趣的物品。如一个用户进入淘宝后,与一些物品进行了交互(点击、收藏、添加购物车和购买),预测其下一个最可能购买的物品。由于会话内的交互记录通常是较短的,因此已有一些基于图神经网络的工作被提出来对这种数据稀疏性进行缓解。然而,这些方法仍然存在一些局限:

首先,这些方法通常只考虑相同类型的用户行为来进行预测,而忽视了对于其他类型的用户行为的利用。例如为了预测用户下一个可能购买的物品,只考虑该会话内先前的其他购买行为。但是购买行为通常是较少的,且其他类型的行为也可能有助于购买的预测。

其次,这些方法通常每次都只对一个用户的行为序列进行建模,而忽视了不同用户之间的行为序列的可迁移的类似行为模式。例如,如果许多其他用户在购买商品 A 之后又购买了商品 B,则如果目标用户刚购买商品 A,则商品 A 和商品 B 之间的关系就显得尤为重要。

针对以上两条问题,文中提出了 Multi-relational Graph Neural Network (MGNN) 来实现基于会话的目标行为预测。即在构建一个 item2item 的全局关系图的时候,同时使用目标行为和辅助行为信息。

2 相关工作

2.1 基于会话的行为预测

在以往的工作中,基于会话的推荐技术使用的是循环神经网络^[2],将过去交互过的项目作为不同时间步长的输入。后来有的研究^[3]使用注意力机制来捕获用户的长期兴趣和短期偏好。随着图神经网络

络的兴起，也随之被引入到推荐领域。但是以往的工作都存在着不足，只关注使用相同类型的用户行为作为下一个项目预测的输入，而忽略了利用其他类型的行为作为辅助信息的潜力^[1]。

2.2 多行为建模

通常在实际场景中会有多种行为类型数据的存在，许多研究使用多种类型的行为数据提升模型建模效果。最初的研究基本上是通过矩阵分解的方法进行建模，模型预测能力有所改善但效果并不显著。随着神经网络的发展，研究^[4]提出了使用神经网络方法来学习具有不同行为的用户-项目交互的表示。作者的工作在根本上不同于上述研究，因为它们都假设不同用户-项目的交互具有独立性，而作者的研究通过考虑对用户按时间顺序的一系列行为进行建模，这显得更为符合实际^[1]。

2.3 图神经网络

图神经网络是用于生成图结构数据表示的方法，例如社交网络和知识图谱^[1]。最初，研究^[5]通过提出一个模型 DeepWalk 来扩展 Word2vec^[6]，根据从图中采样的序列来学习节点表示。后来出现了基于图的卷积结构模拟方法^[7]，也就是原始的 GCN 版本。后面出现了大量针对图卷积方法的研究，也有的工作将其应用于推荐领域。

3 本文方法

3.1 问题定义

对于会话集合 S 中的一个会话 s ，我们根据不同的行为类型划分，可以分为目标行为序列 $P^s = [p_1^s, p_2^s, p_3^s, \dots, p_{|P^s|}^s]$ 和辅助行为序列 $Q^s = [q_1^s, q_2^s, q_3^s, \dots, q_{|Q^s|}^s]$ 。同时，我们还构建了一个多关系项目图 $\text{MRIG} \mathcal{G} = (\nu, \varepsilon)$ ，其中 ν 是图中所有项目的节点集， ε 是图中多种行为的有向边集。每一条边都包含了 3 个信息：起始项目、指向项目和行为类型。举例：我们基于点击和购买行为构建了一个图，然后一条边 $(a, b, \text{share}) \in \varepsilon$ 代表的意思是一个用户分享了项目 a 之后再分享了项目 b 。这时候，我们就可以把问题形式化地表达为：

基于会话的目标行为预测。给到一个会话 $s \in S$ 及其目标和辅助行为序列 P^s 、 Q^s ，以及多关系项目图 \mathcal{G} 。本问题的目标是学习到一个模型，这个模型可以产生 k 个候选项目，这 k 个项目是该用户在这个会话中接下来最有可能进行交互的^[1]。

3.2 总览

本文所提出的 MGNN-SPred 的总体架构如图 1 所示。MGNN-SPred 的输入包含多关系项目图 (MRIG) 和两种类型的行为序列。SR-MRIG 首先通过图神经网络从 MRIG 中学习项目之间的相关性，并将它们编码为项目表示。之后，一个用户的两个行为序列被视为 MRIG 中的两个子图，每个子图中的项目是分别连接一个虚拟节点（图 1 中的“T”或“A”）。随后，SR-MRIG 将每个子图的节点聚合到对应的虚拟节点，从而得到每个行为序列的表示。最后，为了融合两种行为表示并获得用户偏好表示，采用门控机制自适应地决定不同行为的重要性并对它们进行加权求和。基于推荐的目的，SR-MRIG 通过双线性乘积计算用户和项目表示得出每个项目的分数，并使用分数对它们进行排名以进行推荐。^[1]

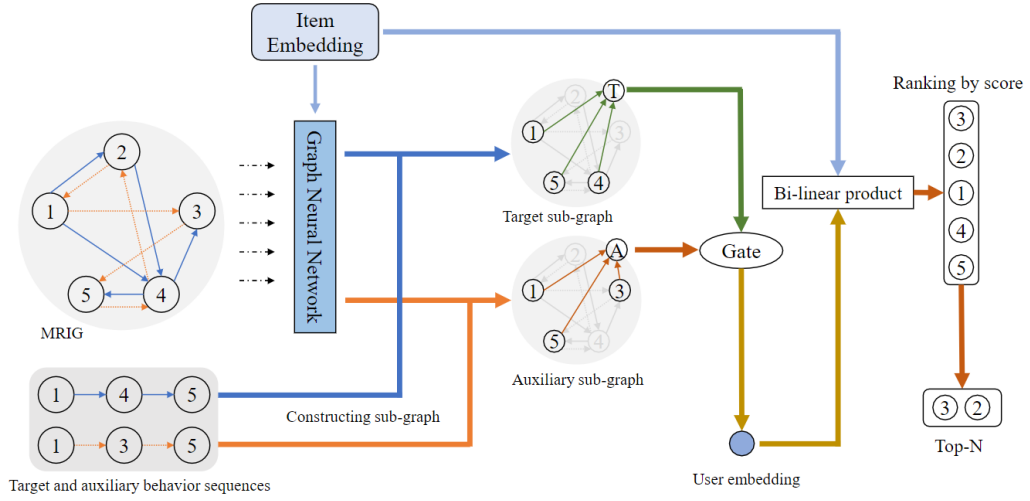


图 1: 方法示意图 (原论文图)

3.3 图结构

对于一个购买的场景，我们通常会点击很多的物品进行浏览，最后购买其中一个。我们很容易就能想到，浏览的这些物品之间应该是存在着某种关系的，例如相关性。而前后购买的两个物品也应该存在某种关系，例如依赖性。我们通过图结构去试图捕获这样的关系。在图中，我们将所有的物品作为节点。对于一条会话序列，我们将其拆分成两条子序列，分别为目标行为序列和辅助行为序列。对每条子序列，我们将序列中前后相邻的两个物品对应图中的节点连接上一条边，并标记上对应的行为类型。

构建多关系物品图的伪代码如下所示：

Procedure 1 多关系物品图构造

Input: 会话集 S , 包含了目标和辅助行为序列 $P^s, Q^s, \forall s \in S$

Output: 多关系物品图 MRIG $\mathcal{G} = (\nu, \varepsilon)$

```

 $\nu \leftarrow \emptyset, \varepsilon \leftarrow \emptyset$ 
for  $s \in S$  do
     $\nu \leftarrow \nu \cup \{P^s[1]\}$ 
    for  $i = 2$  to  $|P^s|$  do
         $\nu \leftarrow \nu \cup \{P^s[i]\}, \varepsilon \leftarrow \varepsilon \cup \{(P^s[i-1], P^s[i], \text{target})\}$ 
    end
     $\nu \leftarrow \nu \cup \{Q^s[1]\}$ 
    for  $i = 2$  to  $|Q^s|$  do
         $\nu \leftarrow \nu \cup \{Q^s[i]\}, \varepsilon \leftarrow \varepsilon \cup \{(Q^s[i-1], Q^s[i], \text{auxiliary})\}$ 
    end
end

```

* 伪代码源自原论文

3.4 学习项目的嵌入表示

对于一个节点 $v \in V$, 我们使用独热编码 $\bar{e}_v \in \mathbb{R}^{|V|}$ 表示, 通过一个可学习的嵌入矩阵将其转化为低维的嵌入表示 $e_v = E^\top \bar{e}_v$ 。

我们将转换后的嵌入表示 $e_v (\forall v \in \nu)$ 和图 MRIG \mathcal{G} 输入到 GNN 中进行卷积操作来更新节点。对于一个节点 v , 更新它的嵌入表示需要先找出它的邻居。根据行为类别和边的方向，我们可以将节点

v 的邻居分为 4 种类型，我们分别命名为：“target-forward”、“target-backward”、“auxiliary-forward”和“auxiliary-backward”。以“target”行为类型为例，得到节点 v 对应的邻居组如下：

$$\mathcal{N}_{t+}(v) = \{v' \mid (v', v, \text{target}) \in \mathcal{E}\}, \mathcal{N}_{t-}(v) = \{v' \mid (v, v', \text{target}) \in \mathcal{E}\} \quad (1)$$

对于“auxiliary”类型，其邻居组即 $\mathcal{N}_{a+}(v)$ 和 $\mathcal{N}_{a-}(v)$ ，通过相同的方式获取。

节点更新操作中，我们先使用均值聚合每组邻居：

$$\mathbf{h}_{t+,v}^k = \frac{\sum_{v' \in \mathcal{N}_{t+}(v)} \mathbf{h}_{v'}^{k-1}}{|\mathcal{N}_{t+}(v)|}. \quad (2)$$

其余三组通过类似的操作进行计算。最终，我们将求得的 4 组嵌入表示进行求和操作：

$$\bar{\mathbf{h}}_v^k = \mathbf{h}_{t+,v}^k + \mathbf{h}_{t-,v}^k + \mathbf{h}_{a+,v}^k + \mathbf{h}_{a-,v}^k \quad (3)$$

最后，我们对中心节点 v 进行更新：

$$h_v^k = h_v^{k-1} + \bar{\mathbf{h}}_v^k \quad (4)$$

进行迭代后，节点的嵌入表示为： $g_v = h_v^K$ 。

3.5 学习序列的嵌入表示

原作者使用了多种方法计算虚拟节点的的嵌入表示，包括使用注意力机制。最终发现，使用简单的均值操作就能取得良好的效果，同时还能降低算法的复杂度。对于目标行为序列 P 和辅助行为序列 Q ，对序列中的节点嵌入表示取均值：

$$\mathbf{p} = \frac{\sum_{i=1}^{|P|} \mathbf{g}_{p_i}}{|P|}, \mathbf{q} = \frac{\sum_{i=1}^{|Q|} \mathbf{g}_{q_i}}{|Q|} \quad (5)$$

由于考虑到每个用户对不同行为的关注度不一样，作者设计了一个门控机制来分配两个行为的权重 α ：

$$\alpha = \sigma(W_g[p; q]) \quad (6)$$

其中 $[p; q]$ 表示两个嵌入表示的拼接， σ 是 sigmoid 函数， $W_g \in \mathbb{R}^{1 \times 2d}$ 是我们模型的可训练参数。最后，我们通过加权求和得到当前会话的用户偏好表示：

$$o = \alpha \cdot p + (1 - \alpha) \cdot q \quad (7)$$

3.6 模型预测和训练

我们进一步使用项目嵌入表示 e_v 计算每个项目 $v \in \nu$ 的推荐分数 s_v 。这里我们采用双线性层计算：

$$s_v = o^\top W e_v \quad (8)$$

其中， $W \in \mathbb{R}^{d \times d}$ 是模型的可训练参数矩阵。

为了学习我们模型的参数，我们使用 softmax 函数对所有项目的分数 $s \in \mathbb{R}^{|\mathcal{V}|}$ 进行归一化以获得概率分布 \hat{y} ：

$$\hat{y} = \text{softmax}(s) \quad (9)$$

采用神经网络的反向传播通过最小化预测概率分布 \hat{y} 的交叉熵损失来优化模型。损失函数定义如

下:

$$\mathcal{L} = - \sum_{i=1}^m y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (10)$$

其中 y 表示目标项目的独热编码。

4 复现细节

4.1 与已有开源代码对比

本次目标复现的工作有基于 TensorFlow 框架下的源代码, 本人复现工作采用的是基于 Pytorch 框架下进行。由于两种框架下的代码差异较大, 并不具备太多的参考意义, 故 TensorFlow 框架下的代码只用作验证本人复现代码的基线。

4.2 数据集

本工作中, 将采用两个不同的数据集, 其中一个是原论文中使用到的 Yoochoose 数据集, 另一个是 UB 数据集。

Yoochoose 数据集来源于 RecSys Challenge 2015, 原数据集含有点击和购买两种数据, 我们根据时间顺序将数据集中的用户行为序列分割成会话, 其中所有用户都是匿名的。

UB 数据集是淘宝用户购物行为数据集。本数据集包含了 2017 年 11 月 25 日至 2017 年 12 月 3 日之间, 约一百万随机用户的所有行为 (行为包括点击、购买、加购、喜欢)。最终数据集将只保留点击和购买的行为记录。

4.3 数据处理

一个会话中给定目标行为序列 $P = [p_1, p_2, \dots, p_{|P|}]$ 和辅助行为序列 $Q = [q_1, q_2, \dots, q_{|Q|}]$, 我们将每个项目 $p_i (i \geq 2)$ 视为标签, 并使用 $[p_1, p_2, \dots, p_{i-1}]$ 作为目标行为的输入。辅助行为的处理方式略有不同, 因为用户很可能在购买或分享之前点击某个项目。为避免辅助行为输入已经看到标签, 我们只保留用户已购买或分享的目标商品之前的已点击商品。我们为两种类型的序列设置一个最大长度 L , 超出长度的将进行截取。

对两个数据集, 采用了两种不同的训练集划分方法, 分别是绝对时间划分和相对时间划分。

对于 Yoochoose 数据集, 我采用的是绝对时间划分方法。即确定数据集中行为发生的最早时间和最晚时间, 取该时间段的 70% 位置作为划分点, 在此之前发生的行为划进训练集, 之后发生的行为划进测试集。

对于 UB 数据集, 我采用的是相对时间划分方法。将数据集中相同的 (用户, 物品, 行为) 三元组删除后, 按照相同用户的行为历史构成会话序列。对于一条多行为序列, 将序列中倒数第二个目标行为之前的目标行为作为训练集的标签, 倒数第二个目标行为作为测试集的标签, 最后一个目标行为作为测试集的标签。

5 实验结果分析

在本次的工作中, 我是使用了两个数据集分别采用两种不同的数据处理方式进行试验。为了验证复现工作的正确性, 这里采用了基于 TensorFlow 框架的开源代码在相同条件下进行实验作为基线, 对

数据集	Yoochoose	UB
数据量	201961	212223
节点数	52740	81616
图 1 边数	225879	1459705
图 2 边数	3277411	193862
训练集	163005	167416
验证集	12985	14935
测试集	25971	29872

表 1: 数据集的基本信息

比两者的指标差异来确定是否完成复现内容。

最终的实验结果如表 2 所示，复现的代码在两个数据集上与参照代码的指标表现相近，可以认定复现工作正确。

	NDCG@10(%)	NDCG@100(%)
数据集	Yoochoose	UB
MGNN_tf	8.273	1.7724
MGNN_torch(ours)	8.334	1.7323

表 2: 两个数据集在两个模型上的 NDCG 指标表现

在实验过程中，我还对超参数的设置进行了实验，在 Yoochoose 数据集上，我对不同的序列长度和卷积深度进行实验比较。

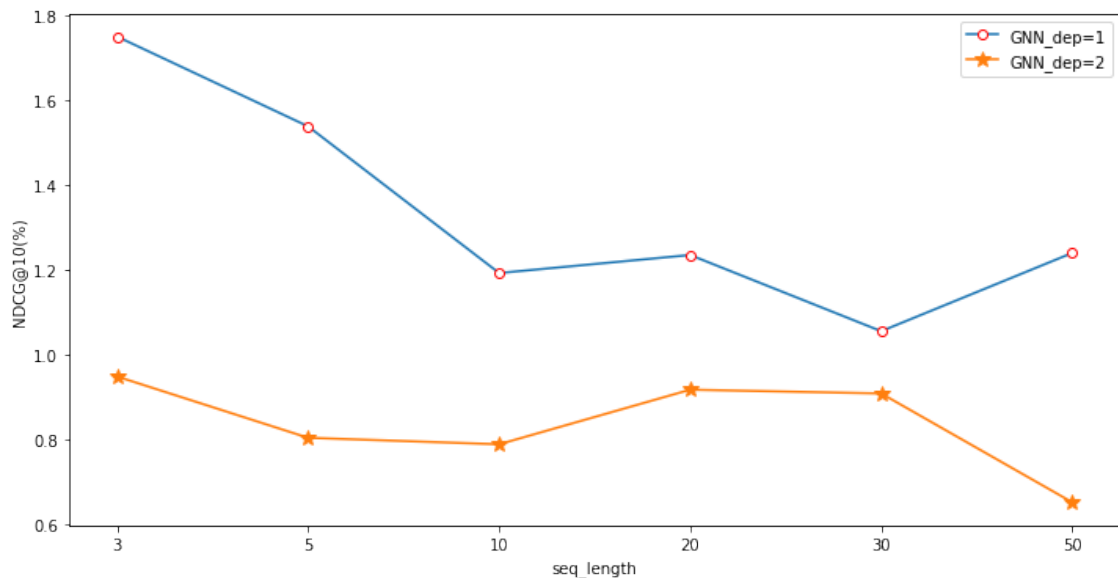


图 2: 在 Yoochoose 数据集上对不同序列长度和卷积深度实验

结果如图 2 所示, 我们可以看到随着序列长度的设定逐渐增大, 模型效果也随之变差, 当序列长度设定为 3 时取得最优, 这说明了会话中用户的下一目标行为项目与最近的交互过的项目更为相关。同时我们可以看到, 增加卷积深度的时候, 模型效率并不会增加, 这也验证了前面的观点。

在 UB 数据集上, 我对模型中项目的嵌入表示的特征维度选择也做了实验。

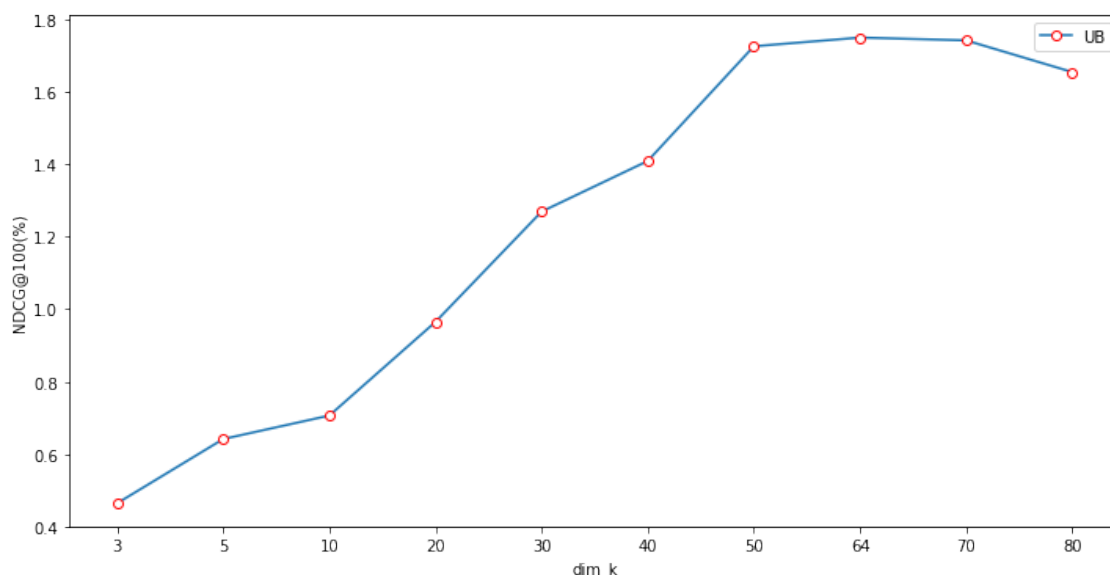


图 3: 在 UB 数据集上对不同特征维度实验

结果如图 3 所示，随着特征维度的增加，模型效果也会随之上升，上升到一定程度的时候就会下降。由图中可见，当特征维度设为 64 的时候取得最优。

6 总结与展望

本次复现工作，选取的是近几年在推荐领域里图卷积方法应用于序列推荐中比较受关注的一个工作。使用图卷积来构建复杂项目关系并应用于预测是这边比较新也是比较火的方法，通过复现这么一个工作，可以使我熟练地快速构建更多具有创新的新型图结构，便于后面工作更好地展开。在复现的过程中也遇到过不少的问题，比如说数据的处理，处理方式文字很好描述，但是落到代码层面就会出现很多需要注意的地方，有时候数据并不是十分的规整，这个时候就需要对数据进行切分、填充等操作，但又不能影响最终的结果。在数据组织上存在有两种组织方式，一种是相对时间组织，另一种是绝对时间组织，为了便于以后科研的展开，在本工作中两种方式均有采用。在图卷积代码的编写过程中，有考虑过使用纯手写代码还是使用卷积库。考虑到以后工作中可能会遇到各种各样的卷积方式，使用纯手写则会每次改动大量代码，所以本次工作中使用库实现卷积，这样实现起来更为规整。本工作采用了其他框架下的源代码进行对比，以验证复现工作的准确性，同时对各项超参数进行了大量实验对比选取最优值，最终的效果符合预期。

本次复现的工作是一个在序列推荐领域种比较有开创性的工作，模型相对简单，具备很多可以改进的地方，比如说模型中对边的选择是随机剪枝的，可以改为计算每一条边的权重，选取出权重较高的进行保留，较低的进行裁剪操作。本工作在图卷积过后，就直接将得到的嵌入表示进行线性计算，最后直接与所有项目的嵌入表示求内积计算分数，这一步相对简单，可以做一些更为复杂的操作，对模型效果应该会有提升。

参考文献

- [1] WANG W, ZHANG W, LIU S, et al. Beyond Clicks: Modeling Multi-Relational Item Graph for Session-Based Target Behavior Prediction[J]., 2020.
- [2] HIDASI B, KARATZOGLOU A, BALTRUNAS L, et al. Session-based Recommendations with Recur-

rent Neural Networks[J]., 2016.

- [3] QIAO L, ZENG Y, MOKHOSI R, et al. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation[J]., 2018.
- [4] GAO C, HE X, GAN D, et al. Neural Multi-task Recommendation from Multi-behavior Data[J]., 2019.
- [5] PEROZZI B, AL-RFOU R, SKIENA S. DeepWalk: Online Learning of Social Representations[J]. ACM, 2014.
- [6] MIKOLOV T, SUTSKEVER I, KAI C, et al. Distributed Representations of Words and Phrases and their Compositionality[J]., 2013.
- [7] BRUNA J, ZAREMBA W, SZLAM A, et al. Spectral Networks and Locally Connected Networks on Graphs[J]. Computer Science, 2013.