

# 基于序列感知信息的序列推荐模型

宁春梅, 孙 博

(陕西国防工业职业技术学院 计算机与软件学院, 陕西 西安 710300)

**摘要:** 由于用户行为序列随时间动态增长, 推荐算法需要捕获用户行为的动态变化, 以及由于用户偏好随时间变化且会影响用户行为, 推荐算法需要综合考虑用户的特征属性和即时兴趣来做 Top-K 推荐。文章提出了一种基于序列感知信息的序列推荐模型 TPSR, 其利用自注意力机制模拟用户行为序列的长期依赖关系来得到用户的长期偏好表征, 即用户特征属性。因此, TPSR 模型能够综合考虑用户的属性特征和即时兴趣, 从而得到用户真实行为偏好。文章在亚马逊数据集的子数据集下进行验证实验, 与对比算法相比, TPSR 能够为用户推荐更符合其偏好的物品。

**关键词:** 序列推荐; 用户建模; 序列感知信息; 门控学习网络

**中图分类号:** TP391.3

**文献标识码:** A

**文章编号:** 2096-9759(2023)06-0081-03

## 0 引言

通常来说, 一个推荐算法包括两个部分: 候选生成和候选排序。候选物品的生成阶段旨在从大量的物品集合中检索出少量的用户感兴趣的物品。候选物品的排序就是根据点击率(CTR)或者评分对检索到的物品进行排序。例如基于矩阵分解或深度神经网络的推荐算法都可以使用这种检索方式。

然而, 利用向量点积来计算物品的相关性限制了某些神经网络模型的功能, 如最简单的单层神经网络模型 AutoRec(自编码器推荐)<sup>[1]</sup>。为了解目标用户和候选物品之间更深层次的非线性关系, 一些研究人员针对候选排序阶段提出了一些效果表现更好的模型, 例如 DIN(Deep Interest Network)模型<sup>[2]</sup>。由于本文旨在帮助用户从海量物品信息中筛选出用户可能感兴趣的 Top-K 个物品, 并不关注这 K 个物品的排序, 因此本文主要关注推荐算法的候选生成阶段。

序列推荐的输入可以是用户的时序行为序列, 即将用户与物品的交互行为按照时间顺序排列得到的序列。受自然语言处理领域启发, 许多深度神经网络被引入到序列推荐领域, 并用于处理用户的时序行为序列。

序列推荐模型是根据用户的历史行为预测用户未来可能关注(即交互)的物品。马尔代夫链(Markov Chain, MC)就是一种传统的序列推荐方法。这种方法在稀疏的数据信息中表现良好, 但是在复杂电商推荐场景中可能无法准确获取到用户真正的偏好意图。Rendle 结合了矩阵分解(MF)和马尔可夫链(MC)的方法提出了 FPMC(Factorized Personalized Markov Chains)模型。任俊伟<sup>[3]</sup>将循环神经网络(RNN)应用于会话推荐领域, 其利用 GRU 循环神经网络对点击序列进行建模, 通过考虑序列的顺序特征来提升 CTR 预测的准确性。此外, 顾军华<sup>[4]</sup>验证了用户的点击意图取决于用户的长期稳定的兴趣爱好和短期的意图。赵程栋<sup>[5]</sup>提出了一种基于注意力机制的 RNN 推荐模型, 其利用循环神经网络和注意力机制学习用户偏好的演进过程。

实际上, 序列中存在的不仅是顺序依赖关系, 序列中不连续的行为发生时间信息也是处理序列推荐的重要信息。最近的一些研究通过添加时间门控函数来改进 GRU 或 LSTM 单元, 从而捕获用户行为序列的时间信息。宁春梅<sup>[6]</sup>提出了一种加入时间门控的方法来提升 LSTM 的效果, 目的就是

为了捕获用户的长期和短期偏好, 从而实现更为准确的用户行为预测。陈一凡<sup>[7]</sup>探讨了不同时间间隔对用户未来行为预测的准确性的影响, 其实验结果表明用户与物品的交互行为的时间间隔对于捕获用户行为之间的相关性具有重要影响。

本文提出的 TPSR 模型充分利用了用户行为序列中的时间和位置特征, 来建模用户行为序列, 综合考虑了用户自身特征属性和即时兴趣的变化, 从而能够得到真实的用户兴趣偏好表征, 准确预测用户未来行为。

## 1 TPSR 模型总体框架

本文提出了一种基于循环神经网络和自注意力机制的序列推荐模型 TPSR。该模型能够根据用户的历史行为预测用户未来可能交互的物品。该模型充分利用用户行为的时间和位置信息, 对用户偏好动态建模。本节将详细介绍所提出的基于序列感知信息的序列推荐模型 TPSR。

### 1.1 基本定义

假设在推荐系统中有  $M$  个用户和  $N$  个物品, 用户  $u$  的历史行为序列为  $S_u = (b_{u,1}, b_{u,2}, \dots, b_{u,|S_u|})$ 。本文定义用户的交互行为  $b_{u,i} = (x_{u,i}, t_{u,i}, e_{u,i}, p_{u,i})$  作为用户历史行为序列中的第  $i$  次交互行为, 其中  $x_{u,i} \in V$  是用户  $u$  在时间  $t_{u,i}$  交互的物品,  $p_{u,i}$  表示物品  $x_{u,i}$  在序列  $S_u$  中的位置,  $e_{u,i}$  表示用户行为的上下文信息  $e_{u,i}$  可以包括很多和行为相关的重要特征(如: 物品的类别、交互行为所属的类别和物品所属的商家等)。

TPSR 模型所解决的问题是, 当给定一个用户  $u$  的历史行为序列为  $S_u = (b_{u,1}, b_{u,2}, \dots, b_{u,i})$  时, 序列推荐模型可以预测出用户  $u$  将要在时间  $t_{u,i+1}$  交互的物品  $x_{u,i+1}$ 。本文提出了一个模型方法 TPSR, 当模型得到用户历史行为序列  $S_u = (b_{u,1}, b_{u,2}, \dots, b_{u,i})$ 、交互行为发生的目标时间  $t_{target}$  和目标行为在序列中的位置  $p_{target}$  时, 可以得到模型方法给出的 Top-K 个物品列表  $y = (y_1, y_2, \dots, y_k)$ , 定义如下:

$$y = \text{TPSR}(S_u, t_{target}, p_{target}) \quad (1)$$

### 2.2 建模用户长期偏好特征

为了捕获用户行为序列中时间感知和位置感知的上下文信息, 本文将时间间隔和位置间隔建模为时间和位置门控, 门控函数定义为:

$$g_{ij} = f(h_i, h_j, t_{h_i} - t_{h_j}, p_{h_i} - p_{h_j}) \quad (2)$$

收稿日期: 2023-03-13

作者简介: 宁春梅(1995-), 女, 河南商丘人, 硕士, 助教, 研究方向: 机器学习、智能推荐; 孙博(1996-), 男, 辽宁沈阳人, 硕士, 助教, 研究方向: 云计算、计算机存储。

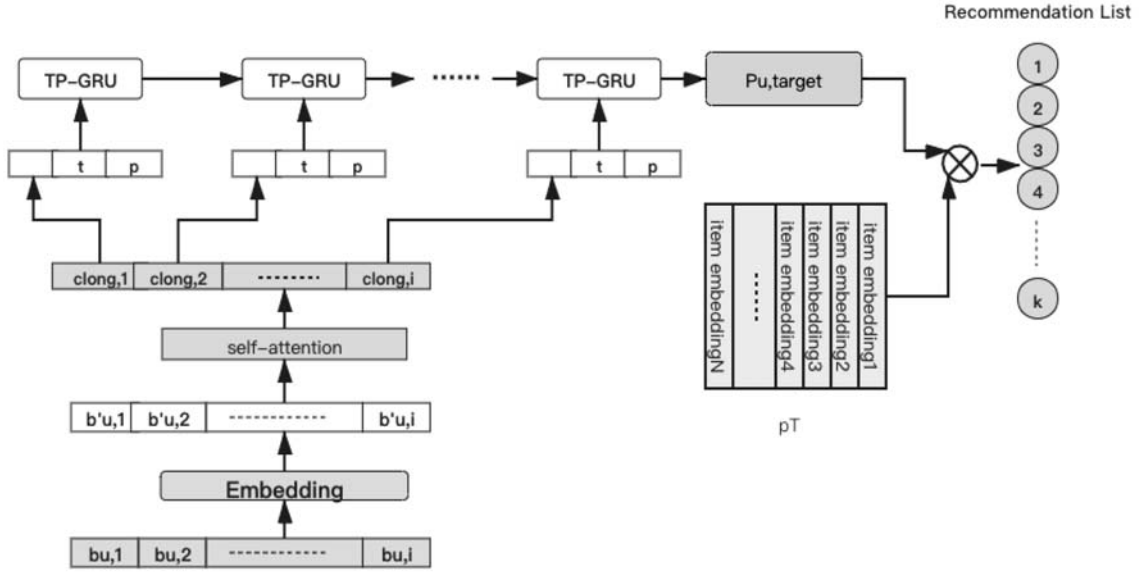


图1 TPSR 模型框架

其中 $(h_s, t_s, p_s)$ 分别是隐藏层的表征向量、交互行为发生的时间戳和交互行为在序列中的位置。通过门控函数，两个交互行为的时间和位置关系由时间间隔和相对位置及其各自的隐藏层状态的表征向量来确定。此外，循环神经网络的递归更新可以被定义为：

$$h_s = f(x_s, h_{s-1}) \quad (3)$$

其中 $(x_s, h_s)$ 分别是当前时刻的输入和隐藏层的状态向量。在循环神经网络中，GRU 是一种常用的循环单元。GRU 的计算规则用公式可以表示为：

$$z_s = \sigma([x_s, h_{s-1}]W_z + b_z) \quad (4)$$

$$r_s = \sigma([x_s, h_{s-1}]W_r + b_r) \quad (5)$$

$$h'_s = \phi([x_s, h_{s-1} \odot r_s]W_h + b_h) \quad (6)$$

$$h_s = z_s \odot h_{s-1} + (1 - z_s) \odot h'_s \quad (7)$$

其中， $[\cdot, \cdot]$ 表示将两个向量进行拼接操作； $+$ 和 $\odot$ 分别表示向量元素的对应相加和向量的点积计算； $\sigma$ 和 $\phi$ 分别表示 sigmoid 和 tanh 激活函数； $z_s$ 和 $r_s$ 表示 GRU 中的更新门和重置门； $h'_s$ 表示候选状态； $h_{s-1}$ 表示历史隐藏层状态向量； $h_s$ 表示隐藏层状态输出向量。此外，隐藏层的输出向量 $h_s$ 是由候选状态信息 $h'_s$ 和历史隐藏状态信息 $h_{s-1}$ 根据更新门 $z_s$ 进行线性变换得到的。

为了捕获序列中上下文的时间相关性和位置相关性，本文在 GRU 单元中新加入了时间门函数和位置门函数。时间和位置门控函数是由当前时刻的输入 $x_s$ 、历史隐藏状态 $h_{s-1}$ 、时间间隔 $t_s - t_{s-1}$ 和当前状态 $s$ 在序列中的位置信息定义的：

$$\delta_s = \phi(\log(t_s - t_{s-1} + 1) \odot W_{\delta_s} + b_{\delta_s}) \quad (8)$$

$$\tau_s = \phi([x_s, h_{s-1}]W_{\tau_s} + b_{\tau_s}) \quad (9)$$

$$\alpha_s = \phi(\log(p_{target} - p_s + 1) \odot W_{\alpha_s} + b_{\alpha_s}) \quad (10)$$

$$g_s = \sigma(\delta_s \odot W_{g_s\delta} + \tau_s \odot W_{g_s\tau} + \alpha_s \odot W_{g_s\alpha} + b_{g_s}) \quad (11)$$

其中，时间特征 $\delta_s$ 表示相邻两个交互行为的时间关系； $t_s - t_{s-1}$ 代表相邻两个交互行为的时间间隔； $\tau_s$ 表示此刻的上下文语义关系；位置特征 $\alpha_s$ 表示当前状态在序列中的位置信息； $p_{target}$ 代

表模型预测下一项在序列中的位置； $p_s$ 代表当前状态的位置。此外，时间和位置门控 $g_s$ 是由时间特征 $\delta_s$ 和位置特征 $\alpha_s$ ，以及上下文语义关系 $\tau_s$ 进行非线性组合而得到。

在序列推荐中，本文将注意力机制与循环神经网络相结合，可以协助循环神经网络模拟用户行为序列的长期依赖关系，并帮助所提出的模型关注到用户行为序列中的关键行为信息。

自注意力机制中，通过线性变换的方式，可以将原始输入矩阵映射成 $Q$ 、 $K$ 和 $V$ 三个矩阵，然后通过点积计算得出 $Q$ 和 $K$ 的关联矩阵，计算方式如下：

$$\text{similarity}(Q, K) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (12)$$

关联矩阵中的值表示序列中每对物品的相关程度，然后利用关联矩阵与映射后的矩阵 $V$ 进行加权求和，自注意力机制方法计算定义如下：

$$\text{Attention}(Q, K, V) = \text{similarity}(Q, K)V \quad (13)$$

在用户行为序列中，用户的每次交互行为都不是孤立存在的，它们在序列中都有迹可循。因此，为了模型更好地学习用户行为序列中潜在的用户行为兴趣，本文采用自注意力机制的方法构建序列中行为间的长期依赖关系，从而得到用户的长期偏好表征（即用户的属性表征）向量。

此外，作为一种基于矩阵分解的深度神经网络的模型，所提出的 TPSR 模型可以执行最近邻搜索以生成基于预测出的用户偏好 $P_{u,target}$ 和物品 Embedding 空间之间的点积相似度 $P_{u,target}P^T$ 的推荐列表：

$$y = P_{u,target}P^T \quad (14)$$

TPSR 模型采用标准的小批量梯度下降法来优化迭代，目标函数是多分类的交叉熵损失函数，损失函数定义如下：

$$y'_u = \text{softmax}(P_{target}P^T) \quad (15)$$

$$L(y', \hat{y}) = \sum_u^N y'_u \log \hat{y}_u \quad (16)$$

其中 $y'_u$ 表示预测下一个物品的概率分布， $\hat{y}_u$ 表示数据集真实下一个物品的独热(one-hot)编码。

### 3 实验与分析

#### 3.1 实验数据集

本文在真实数据集亚马逊电子产品数据集上进行两组实验(即 Top-5 和 Top-10 预测),并且对实验结果进行分析讨论。

表 1 数据集统计分析

数据集	#用户	#物品	#类别	行为序列平均长度	稠密度
亚马逊电子产品	18110	11826	588	17	0.66%

为了保证模型训练和预测的准确性,本文对亚马逊电子产品数据集进行了相应的预处理。本文过滤掉交互次数少于 5 次的用户和被交互次数少于 10 次的物品。表 1 中展示了亚马逊电子产品数据集在数据预处理后的统计结果。

#### 3.2 评价指标

为了验证实验中所有模型的效果,本文采用命中率(Hit Ratio, HR)和归一化折损累计增益(Normalized Discounted Cumulative Gain, NDCG)作为模型的评价指标。HR@K 和 NDCG@K 都是 Top-K 推荐模型的常用评价指标,其中的 K 指的是每次推荐含有 K 个项目的候选集。在本文实验中,设  $K=5$  和  $K=10$ 。

#### 3.3 对比模型

除了所提出的 TPSR 模型以外,本文还测试了最新的几种推荐模型在上述亚马逊电子产品数据集上的性能,包括 BERT4Rec 模型<sup>[8]</sup>、GRU 模型<sup>[9]</sup>。

#### 3.4 实验设置

为了确保实验的公平性,所有模型的训练集和测试集是相同的。对于所有的模型参数,本文将隐藏层单元和低维嵌入空间的维度(包括循环神经网络层和注意力层)都设置为 128, Embedding 空间的嵌入维度设置为 64。初始学习率设置为 0.001,并且每迭代 100 次采用衰减率为 0.995 指数学习率衰减的训练方法。同时设置 dropout rate=0.5, regularization rate=0.00001 避免模型的过度拟合。用户行为序列的最大长度设置为 50。所有模型都是用 tensorflow1.14 版本框架实现,并且在 Linux 服务器中(Tesla P100 GPU)进行模型的训练和测试。

#### 3.5 实验结果与分析

实验分为两部分,本文先分析并讨论了 TPSR 模型和其他模型在进行 Top-5 推荐时的实验结果。然后对比了在处理不同长度的用户行为序列时,各个模型的 Top-10 推荐准确率。

表 2 亚马逊电子产品数据集下的 HR 和 NDCG 比较

模型	亚马逊电子产品	
	HR@5	NDCG@5
GRU	0.0260	0.0168
BERT4Rec	0.0269	0.0173
TPSR	0.0359	0.0233
提升率	33.46%	34.68%

表 3 展示了当限制用户行为序列最大长度  $N$  时,各个模型在亚马逊电子产品数据集下的实验结果。这表明,对于没有重复交互行为的评论类型的数据集,用户行为序列越长,序列中包含的用户行为信息越丰富,更加有利于模型抓取用户

偏好并对用户未来行为作出预测。

在亚马逊电子产品数据集中,本文提出的 TPSR 模型在所有的对比模型中普遍表现更佳。因此,从整体实验指标效果来看 TPSR 模型的推荐能力是值得认可的。

表 3 在电子产品数据集下不同序列长度的比较

序列长度(N)	HR@10			
	10	20	50	100
SASRec	0.0472	0.0579	0.1084	0.1157
GRU	0.0462	0.0749	0.1248	0.1334
TPSR	0.0474	0.0759	0.1356	0.1413
序列长度(N)	NDCG@10			
	10	20	50	100
SASRec	0.0188	0.0230	0.0409	0.0442
GRU	0.0185	0.0284	0.0539	0.0572
TPSR	0.0208	0.0291	0.0568	0.0596

### 4 结语

本文提出了一个基于序列感知信息的序列推荐模型 TPSR,该模型对用户的历史行为进行建模,并利用深度神经网络获取用户偏好,从而实现对用户兴趣点的预测。本文在亚马逊电子产品公开测试集上测试的 TPSR 模型的性能。

实验结果表明,TPSR 模型在公开的数据集上的性能表现总体呈优,能够取得较好的预测结果。经过相关对比实验和结果分析,本文发现序列中的时间和位置信息对于序列推荐的任务尤为重要,可以帮助序列推荐模型得到显著的提升。

#### 参考文献:

- [1] 陈子健,李俊,岳兆娟,等.基于自编码器与属性信息的混合推荐模型[J].数据与计算发展前沿,2021,3(3):8.
- [2] Haralick R M, Shanmugam K, Dinstein I. Textural Features for Image Classification[J].Studies in Media and Communication,1973,SMC-3(6):610-621.
- [3] 任俊伟,曾诚,肖丝雨,等.基于会话的多粒度图神经网络推荐模型[J].计算机应用,2021,41(11):3164-3170.
- [4] 顾军华,余士耀,樊帅,等.基于用户长短期兴趣与知识图卷积网络的推荐[J].计算机工程与科学,2021,43(3): 511-517.
- [5] 赵程栋,庄继晖,程晓鸣,等.基于特征注意力机制的 RNN-Bi-LSTM 船舶轨迹预测[J].广东海洋大学学报,2022,42(5):8.
- [6] 宁春梅,孙博,肖敬先,等.基于时间感知注意力机制的混合编码网络方法[J].山东大学学报(工学版),2022(002):052.
- [7] 陈一凡,朱民耀,朱晓强,等.基于时序特征的图卷积协同过滤推荐算法[J].电子测量技术,2022,45(6): 79-85.
- [8] Nikzad-Khasmakhi N, Balafar MA, Reza Feizi-Derakhshi M, et al. BERTERS: Multimodal representation learning for expert recommendation system with transformers and graph embeddings [J]. Chaos Solitons Fractals, 2021, 151: 111260.
- [9] Dang TK, Nguyen QP, Nguyen VS. A study of deep learning-based approaches for session-based recommendation systems [J]. Sn Comput Sci, 2020, 1(4): 216.