

文章编号: 2095-2163(2022)12-0196-06

中图分类号: TP391

文献标志码: A

# 基于 Transformer 的融合用户负反馈的重排序推荐方法

胡德敏, 光 萍

(上海理工大学 光电信息与计算机工程学院, 上海 200093)

**摘要:** 针对现有的 Top -  $n$  推荐模型只利用用户正反馈进行推荐的问题, 本文提出一种基于 Transformer 的融合用户负反馈的重排序推荐方法。具体来说, 本文使用 GRU 提取用户负反馈行为特征; 基于门控单元思想构建一个融合单元, 融合用户正负反馈行为特征, 获得完整的用户行为特征; 使用 Transformer 编码器整合用户特征和物品的位置信息, 经过 *softmax* 计算物品得分获得重排序列表。实验结果表明, 与几种先进的重排序推荐模型相比, 本文的模型在 *AUC* 和 *Precision@k* 指标上有显著提升。

**关键词:** 重排序; 负反馈; Transformer; 位置信息

## Re-ranking recommendation method based on Transformer with user negative feedback

HU Demin, GUANG Ping

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

**[Abstract]** Aiming at the problem that the existing Top -  $n$  recommendation model only applies users' positive feedback for recommendation, this paper proposes a re-ranking recommendation method based on Transformer with users' negative feedback. Specifically, this paper uses GRU to extract the characteristics of users' negative feedback behavior; based on the idea of gating unit, a fusion unit is constructed to fuse the positive and negative feedback behavior characteristics of users to obtain complete user behavior characteristics. After that, Transformer encoder is used to integrate the user characteristics and the position information of items, and the reordering list is obtained by calculating the item score through *softmax* function. The experimental results show that compared with several advanced re-ranking recommendation models, *AUC* and *Precision@k* are significantly improved.

**[Key words]** re-ranking; negative feedback; Transformer; position information

## 0 引言

随着互联网规模和数字信息资源的不断增长, 信息数量呈几何级数激增, 使得人们难以快速、准确地从海量的信息资源中寻找所需的信息。推荐系统是可以有效解决信息过载问题的重要手段, 通过对用户行为数据分析和建模, 预测并推荐用户可能感兴趣的产品。Top -  $n$  推荐作为推荐系统的场景之一, 利用用户对物品做出的反馈记录, 为用户推荐一个可能感兴趣的包含  $n$  个物品的列表, 如学术头条对论文的推荐、网易云课堂对课程的推荐, 其少量准确的推荐更符合用户的选择习惯, 因此近年来 Top -  $n$  推荐已逐渐成为推荐领域内的研究热点。

Top -  $n$  排序模型基于学习排序 (Learning to rank, LTR) 的思想, 分为单文档方法、文档对方法和文档列表方法, 分别从不同角度综合考虑多个排序

特征, 来对若干物品进行排序, 其中心问题是如何有效准确地捕捉用户兴趣, 以提高推荐物品列表的排名质量。大多数现有的 Top -  $n$  推荐模型将用户的正反馈信息, 即用户-物品交互中点击或购买的行为序列作为输入, 却忽略了用户负反馈信息, 即更加丰富的未点击曝光物品, 最终生成的推荐列表趋于同质化和趋热化, 损害了用户体验。在完整的用户反馈信息中, 未点击的物品占相当大的份额, 能够作为辅助信息补充点击行为。

一些研究建模物品之间的相互影响, 再次调整 Top -  $n$  排序模型给出的候选列表, 被称为重排序方法。各种重排序模型<sup>[1-2]</sup> 基于 Transformer 网络结构, 应用自注意力机制和可并行计算的特点, 实现了在常量距离内高效建模列表中任意物品之间的相互影响。受这些重排序研究的启发, 本文基于 Transformer 编码器, 将 Top -  $n$  推荐生成的候选列表

**基金项目:** 国家自然科学基金(61170277, 61472256)。

**作者简介:** 胡德敏(1963-), 男, 博士, 副教授, 主要研究方向: 计算机网络、分布式计算、云计算; 光 萍(1997-), 男, 硕士研究生, 主要研究方向: 推荐系统、深度学习。

收稿日期: 2022-03-30

作为部分输入,首次在重排序过程中额外集成用户负反馈信息,构建一个重排序推荐模型(Re-ranking recommender integrating user negative feedback, NRR),用来解决现有的 Top -  $n$  模型忽略用户负反馈信息的问题。

## 1 相关工作

各种 Top -  $n$  推荐模型<sup>[3-4]</sup>分为单文档方法、文档对方法<sup>[5-6]</sup>和文档列表方法<sup>[7-9]</sup>三类。其中,单文档方法将推荐任务视为一个二元分类问题,借助手工特征工程,全局学习给定的用户-物品对的评分函数,但忽视了候选物品集合中不同物品之间的差异和相互影响,为了解决这一问题,文档对方法和文档列表方法模型的研究逐渐受到多方关注。文档对方法将 2 个物品之间的先后顺序关系纳入研究,但没有考虑物品在整个排序列表中的位置信息和相关物品的数量差异。而文档列表方法的思想是对所有训练实例进行训练得到最优的评分函数,对于一个新的查询请求,评分函数会对每一个物品进行打分并重新做排序。虽然文档列表方法弥补了单文档方法与文档对方法的不足,但现有的文档列表方法通常只依赖用户的正反馈信息进行推荐。

近年来,一些研究者提出重排序模型,通过调整上述 Top -  $n$  模型生成排名列表。Ai 等人<sup>[10]</sup>使用单向 GRU<sup>[11]</sup>将整个列表的信息编码到每个物品的表示中。Zhuang 等人<sup>[12]</sup>使用 LSTM<sup>[13]</sup>不仅对整个排名列表信息进行编码,而且还通过解码器生成重排序列表。对于使用 GRU 或 LSTM 对物品相关性进行编码的方法,存在编码空间的长距离依赖性的问题,Pei 等人以及 Chen 等人提出基于 Transformer<sup>[14]</sup>的 PRM 模型和 BST 模型,应用自注意力机制和并行计算的优势,高效地学习用户兴趣。

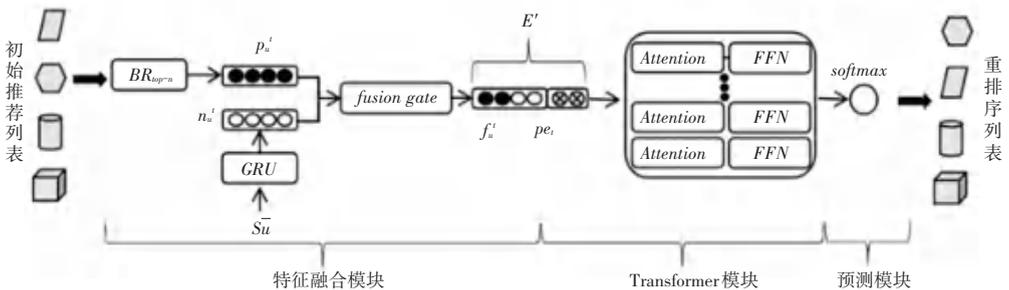


图 1 NRR 模型整体框架图

Fig. 1 Overall framework diagram of NRR model

### 2.2.1 特征融合模块

特征融合模块的工作是获取初始推荐列表的特

## 2 NRR 重排序推荐模型

### 2.1 Top - $n$ 推荐模型与重排序模型的任务描述

Top -  $n$  推荐的任务是根据历史记录中的用户与物品的交互序列,从中学习用户的兴趣偏好,挑选出用户可能感兴趣的包含  $n$  个物品的推荐列表。研究中,Top -  $n$  推荐通过对候选物品集合中的每个物品进行评分来输出一个有序列表,并使用式(1)的损失函数来训练整个 Top -  $n$  推荐模型:

$$L_{topn} = \sum_{r \in R} l(y_i, \{p(y_i | x_i; \theta) | i \in I_r\}) \quad (1)$$

其中,  $R$  是所有用户的查询集合;  $I_r$  是对应某次查询  $r \in R$  的候选物品集;  $x_i$  是物品  $i$  的特征向量;  $y_i$  是对应物品  $i$  的标签(点击与否);  $p$  是基于 Top -  $n$  模型的所有参数  $\theta$  计算出的点击物品  $i$  的概率;  $l$  为该次计算的损失。

而本文的重排序推荐模型则是额外利用用户的负反馈信息,对 Top -  $n$  推荐生成的初始列表的物品重新排序,即给定 Top -  $n$  推荐生成的候选列表  $S_{init} = [i_1, i_2, i_3 \dots, i_n]$ , 通过本文的重排序模型 NRR, 将其调整为最终列表  $S_{re-rank} = [j_1, j_2, j_3 \dots, j_n]$ . 本文 NPRR 模型的全局损失函数定义为:

$$L_{re-rank} = \sum_{r \in R} l(y_i, \{p(y_i | x_i, S_u^-, \hat{\theta}) | i \in S_{init}\}) \quad (2)$$

其中,  $S_u^-$  表示包含用户负反馈行为序列,  $\hat{\theta}$  表示重排序推荐模型的参数集合。

### 2.2 NRR 模型总体架构

NRR 模型的体系结构如图 1 所示。由图 1 可知,该模型包含特征融合模块、Transformer 模块和预测模块三个部分,可将 Top -  $n$  推荐生成的候选列表作为输入,输出重新排序的列表。对此拟展开研究分述如下。

征嵌入、用户负反馈行为特征嵌入,并进行特征融合,得到完整的用户行为特征嵌入。文中给出各流

程步骤的阐释解析如下。

(1) 用户正反馈行为特征。大多数传统 Top- $n$  推荐方法基于用户的正反馈行为,在  $t$  时刻输出一个特定长度为  $n$  的推荐列表  $S_{init}$  提供给用户。本文选取某种基于用户正反馈的 Top- $n$  推荐模型作为基础推荐器 (Basic Recommender, BR), 输入推荐列表  $S_{init}$ , 获取对应的表示用户正反馈行为的特征矩阵  $p_u^t$ :

$$p_u^t = BR_{top-n}(S_{init}, e_u, \theta) \quad (3)$$

其中,  $e_u$  是用户嵌入向量,  $\theta$  表示 Top- $n$  模型的所有参数。

(2) 用户负反馈行为特征。本文使用门控图神经网络 (GRU) 网络提取用户负面偏好。相比传统的序列模型 RNN, GRU 解决了 RNN 的梯度消失和梯度爆炸问题, 同时相对于 LSTM, GRU 的参数数量更少, 训练收敛速度更快, 本文选择 GRU 抽取用户负反馈行为特征。GRU 结构如图 2 所示。

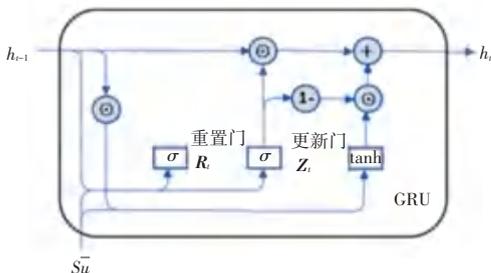


图2 GRU 单元图

Fig. 2 Diagram of GRU unit

本次研究中, 给定用户  $u$  与物品的未点击序列  $S_u^- = \{i_1^-, i_2^-, \dots, i_t^-\}$ , 利用 GRU 提取用户负反馈行为特征  $n_u^t = \{h_1, h_2, \dots, h_t\}$ , 见下列公式:

$$\begin{aligned} Z_t &= \sigma(W_z q_t + U_z) \\ R_t &= \sigma(W_r q_t + U_r) \end{aligned} \quad (4)$$

$$h'_t = \tanh(W q_t + R_t \odot U)$$

$$h_t = Z_t \odot h_{t-1} + (1 - Z_t) \odot h'_t$$

其中,  $q_t$  为物品  $i_t^-$  对应的初始嵌入向量;  $\sigma$  为 Sigmoid 激活函数;  $Z_t$  为遗忘门用于控制前一时刻的状态信息被带入到当前状态中的程度;  $R_t$  为重置门控制前一状态有多少信息被写入到当前的新信息  $h'_t$  上; “ $\odot$ ” 为同或操作;  $W_z, W_r, W, U_z, U_r, U$  为 6 组需要学习的参数矩阵。

(3) 特征融合。特征融合的目的是不同角度的用户行为中提取的特征, 合并成一个具有更精准判别能力的特征。Lv 等人<sup>[15]</sup> 提出一种融合门 (fusion gate) 单元来融合用户的长短期偏好特征进行推荐, 受该研究启发, 本文基于门单元思想构建一个特征

融合单元, 来结合用户负反馈特征  $n_u^t$  与负反馈特征  $p_u^t$ , 获得完整的用户行为特征表示  $f_u^t$ , 特征融合单元结构如图 3 所示。

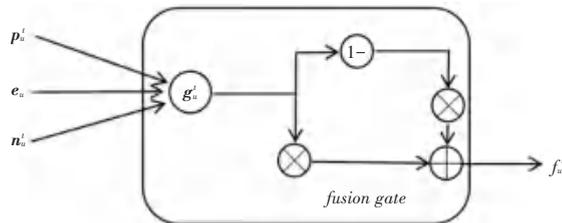


图3 特征融合单元结构图

Fig. 3 Diagram of feature-fusion unit structure

具体来说, 给定用户正负反馈行为特征, 首先使用一个带 Sigmoid 激活的门控向量  $g_u^t$  来控制两者的权重, 见公式:

$$g_u^t = \text{Sigmoid}(W^1 e_u + W^2 p_u^t + W^3 n_u^t + b) \quad (5)$$

其中,  $e_u$  为用户嵌入向量;  $b$  为偏置项;  $W^1, W^2, W^3$  为权重矩阵。

基于门控向量  $g_u^t$ , 得到完整的用户行为特征  $f_u^t$ :

$$f_u^t = (1 - g_u^t) \odot p_u^t + g_u^t \odot n_u^t \quad (6)$$

### 2.2.2 Transformer 模块

考虑到交互序列中物品的顺序性, 即当前时刻推荐给用户的物品列表排名位置会对用户下一时刻交互的物品产生影响。本文使用 Transformer 网络来集成物品位置信息与完整的用户行为特征  $f_u^t$ 。

首先本文将完整的用户行为特征  $f_u^t$  与物品位置信息  $pe_t$  连接, 作为 Transformer 网络的输入向量  $E'$ :

$$E' = f_u^t + pe_t \quad (7)$$

这里,  $pe_t$  可由如下计算公式来求取:

$$\begin{aligned} pe_{t(pos, 2i)} &= \sin(pos / 1000^{2i/d_{model}}) \\ pe_{t(pos, 2i+1)} &= \cos(pos / 1000^{2i/d_{model}}) \end{aligned} \quad (8)$$

其中,  $pos$  是物品在推荐列表中的位置;  $i$  是物品维度;  $d_{model}$  是物品嵌入向量的维度大小。

本文构建的 Transformer 模块包含  $N$  块并行的 Transformer 编码器 (encoder), Transformer 编码器结构见图 4, 每个编码器包含一个注意力层和一个前馈网络 (FFN) 层, 使用的注意力函数如下:

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d}} \right) \frac{V}{d} \quad (9)$$

其中,  $Q, K, V$  分别表示查询、键和值,  $d$  是矩阵  $K$  的维数。

为了模拟更加复杂的物品交互影响, 本文中自

注意力层将最终嵌入矩阵  $E'$  作为输入, 通过线性投影操作将其转换为 3 个矩阵, 输入到多头自注意力层, 如式 (10) 所示:

$$S' = MH(e) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^H \quad (10)$$

$$\text{head}_i = \text{Attention}(E' W^Q, E' W^K, E' W^V) \quad (11)$$

其中,  $W^Q, W^K, W^V \in R^{d \times d}$  为 3 个投影矩阵,  $h$  为多头自注意力的头数。

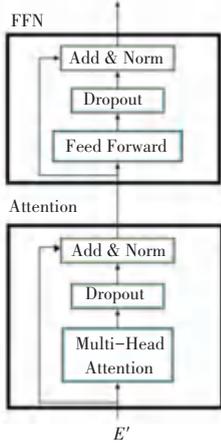


图 4 Transformer 编码器结构图

Fig. 4 Diagram of Transformer encoder structure

接下来, 本文利用 point-wise 式前馈网络 FFN 的非线性特性, 来增强模型的效率, 见下式:

$$F = \text{FFN}(S) \quad (12)$$

为了避免过拟合, 本文同时在注意力层和 FFN 中使用 *dropout* 和 *LeakyReLU* 激活函数, 注意力层和前馈网络的输出分别为:

$$S' = \text{LayerNorm}(S + \text{Dropout}(MH(S))) \quad (13)$$

$$F = \text{LayerNorm}(S' + \text{Dropout}(\text{LeakyReLU}(S' W^{(1)} + b^{(1)}) W^{(2)} + b^{(2)})) \quad (14)$$

其中,  $W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}$  为可学习的参数矩阵。

为了建模交互序列背后的复杂信息, 至此本文并行堆叠  $N$  块 Transformer 编码器来提升计算效率。其中, 第  $b$  块的定义见下式:

$$S^b = \text{SA}(F^{(b-1)}) \quad (15)$$

$$F^b = \text{FFN}(S^b) \quad (16)$$

### 2.2.3 预测模块

预测模块的任务是为初始推荐列表  $S_{init}$  中的每个物品重新生成一个分数, 获得重排序列表  $S_{re-rank}$ 。由图 2 可看到, 本文使用一个 *softmax* 层计算输出每个物品的点击概率  $p(y_i | f_u^i, pe_i | \theta')$ , 作为其最终的分数, 研究推得的数学公式可写为:

$$p(y_i | f_u^i, pe_i | \theta') = \text{softmax}(F^{(N)} + b^f) \quad i \in S_{init} \quad (17)$$

其中,  $F^{(N)}$  是第  $N$  块 Transformer 编码块的输出;  $b^f$  是偏差项。为进行模型训练, 本文使用交叉熵损失:

$$L = -\frac{1}{N} \sum_D (y \log p(x) + (1 - y) \log(1 - p(x))) \quad (18)$$

其中,  $D$  表示样本空间;  $y \in (0, 1)$  为用户是否点击物品的标签;  $p(x)$  是 *softmax* 层的输出。

## 3 实验

### 3.1 数据集实验细节

本文采用 MovieLens-100k 数据集进行实验。MovieLens-100k 数据集包含链接、电影、评分和标签四个主要文件, 其中评分文件包含了每位用户对若干部电影打出的评分, 时间段为 1997-09 ~ 1998-04, 本文基于每位用户的交互数据, 按照时间顺序, 以 80% 作为训练集, 20% 作为验证集, 并使用训练集中的后 10% 作为测试集来评估模型的推荐效果。

本文使用 AdaGrad 作为优化器, 初始学习率设置为 0.001, 衰减率设置为 0.9。对于长度相似序列批量大小 *batch size* 设置为 128, 更新迭代次数为 100 次, 每 25 个 *epoch* 后学习率衰减。实验中各种输入数据的处理、融合网络以及 Transformer 编码器的构建均采用 Python 语言来实现。具体硬件环境为: CPU 选用 Intel Xeon E5-2669 v3, 内存选用 32 GB, 显卡选用 NVIDIA Geforce GTX 2060 Ti 两张, Window10 系统。

### 3.2 未点击物品的筛选规则

推荐系统每一时刻向用户展示一个可能感兴趣的物品列表, 其中未被用户点击的物品包含了用户的负反馈信息。在数据收集过程中, 一个关键的任务是筛选合适的未点击物品, 未点击物品的数量通常比点击物品大得多, 本文使用以下定义过滤规则来挑选未点击数据, 具体如下:

(1) 选择在过去 3 天内展示给用户、但未被点击的物品。

(2) 在上述未被点击的物品中, 只保留向用户公开超过  $k$  (本文取  $k = 1$ ) 次的未点击物品。

本文选择这些物品作为每位用户的负反馈序列, 未点击序列的最大长度被设置为 50。

### 3.3 评价指标

本文使用 *Precision@k* 和 *AUC* 两个指标来评估模型推荐性能。可做重点阐述如下。

(1) *Precision@k*。表示在 Top- $n$  推荐的所有

测试样本中,每检索  $k$  个物品,包含相关物品数量所占的比例,数学计算公式具体如下:

$$Precision@k = \frac{1}{|R|} \sum_{r \in R} \frac{\sum_{i=1}^k I(S_r(i))}{k} \quad (19)$$

其中,  $R$  是测试数据集中所有用户请求的集合;  $S_r$  是重排序模型给出的有序物品列表; 函数  $I$  用来判断第  $i$  个物品是否被用户点击。

(2)  $AUC$ 。衡量了模型对样本正确排名的能力,其值越大,说明模型的排名能力越好,数学计算公式具体如下:

$$AUC = \frac{\sum_{ins_i \in positiveclass} RS_{ins_i} - \frac{M \times (M + 1)}{2}}{M \times N} \quad (20)$$

其中,  $ins_i$  表示第  $i$  条样本;  $M$  和  $N$  分别表示正样本个数和负样本个数;  $positiveclass$  表示正样本类别。

### 3.4 消融对比实验

在本文中为了进一步验证一些拟议模块的有效性,进行了消融对比实验。首先使用 Wide&Deep 模型作为基础推荐器,生成初始推荐列表,接着设计 3 种对比结构,训练 50 个  $epoch$ ,并在  $AUC$  和  $Precision@k(k=5,10)$  指标上比较性能。各结构的内容表述见如下。

(1) 对比结构 A: 本文模型 NRR。

(2) 对比结构 B: 本文模型 NRR 去除物品位置信息  $pe_i$ , 考虑用户负反馈信息  $n_u$ , 用  $NRR - pe_i$  表示。

(3) 对比结构 C: 本文模型 NRR 去除用户负反馈信息  $n_u$ , 考虑物品位置信息  $pe_i$ , 用  $NRR - n_u$  表示。

消融对比模型性能评价结果见表 1。由表 1 分析可知,忽视用户负反馈信息后,本文模型的性能大大降低,在  $Precision@5$  上的降幅是最显著的,达到 10.43%,这表明将用户负反馈信息纳入 NRR 模型的重要性。当去除物品位置信息时,3 项指标略微降低,由此则表明考虑物品位置信息也能在一定程度上提升模型的性能。

表 1 消融对比模型性能评价表

Tab. 1 Performance evaluation table of ablation comparison models

	$NRR - n_u$	$NRR - pe_i$	NRR
$AUC$	0.614	0.663	0.675
$Precision@5$	0.524	0.572	0.585
$Precision@10$	0.487	0.532	0.543

### 3.5 模型对比实验

本文实验将 NRR 模型与主流重排序模型在

$AUC$  和  $Precision@k(k=5,10)$  进行了对比,用于生成初始推荐列表的 Top -  $n$  模型有 GRU4REC 和 Wide&Deep 两种,用于对比的重排序模型分别为 MIDNN、DLCM 和 PRM,下文是对 5 种模型的简要介绍:

(1) GRU4REC<sup>[16]</sup>: 基于会话的采用多层 GRU 单元小批量地学习用户偏好。

(2) Wide&Deep: 结合简单模型的“记忆能力”与深度神经网络的“泛化能力”。

(3) MIDNN: 利用复杂的手工特征提取输入列表信息。

(4) DLCM: 首先采用 GRU 将输入的排序列表编码为一个局部向量,然后将全局向量和每个特征向量相结合,学习一个全局评分函数,用于重排序列表。

(5) PRM: 应用自注意力机制,捕获输入排名列表中物品之间的相互影响。

各对比模型的性能评价结果见表 2。表 2 显示,与 3 种基线模型相比,本文的 NRR 模型实现了显著的性能改进。基于 GRU4REC 生成的候选列表,与 3 种基线模型相比,NRR 在  $AUC$  上平均提升 5.58%, 在  $Precision@5$  上平均提升 8.75%, 在  $Precision@10$  上平均提升 6.68%; 基于 Wide&Deep 生成的初始列表,与 3 种基线模型相比,NRR 在  $AUC$  上平均提升 7.72%, 在  $Precision@5$  上平均提升 9.59%, 在  $Precision@10$  上平均提升 6.70%。

表 2 对比模型性能评价表

Tab. 2 Performance evaluation table of comparison models

Top - $n$ 模型	重排序模型	$AUC$	$Precision@5$	$Precision@10$
GRU4REC	MIDNN	0.606	0.506	0.486
	DLCM	0.630	0.531	0.508
	PRM	0.642	0.548	0.523
	NRR	0.663	0.579	0.542
	Improved/%	5.58	8.75	6.68
Wide&Deep	MIDNN	0.599	0.505	0.490
	DLCM	0.624	0.529	0.509
	PRM	0.643	0.550	0.526
	NRR	0.674	0.584	0.546
	Improved/%	7.72	9.59	6.70

## 4 结束语

本文提出了基于 Transformer 的融合用户负反馈的重排序推荐方法来对传统的 Top -  $n$  推荐方法生成的初始列表进行重新排序。在 NRR 重排序模

型中,本次研究中建模用户未点击的物品序列来丰富用户行为特征表示,并使用 Transformer 编码器整合用户特征和物品的位置信息。实验表明,NRR 模型与几种流行的重排序模型相比有显著的性能提升。

用户负反馈信息尽管较为丰富,但是存在一定的噪声。如何对负反馈进行降噪处理来提升 NRR 模型的性能将是未来工作的研究重点。

## 参考文献

- [1] PEI Changhua, ZHANG Yi, ZHANG Yongfeng, et al. Personalized Re Ranking for Recommendation[C]//RecSys.Denmark;ACM,2019; 3-11.
- [2] CHEN Qiwei, ZHAO Huan, LI Wei, et al. Behavior sequence transformer for e-commerce recommendation in alibaba [C]// Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data. New York, USA; ACM, 2019;1-4.
- [3] CHENG H T, KOC L, HARMSSEN J, et al. Wide & deep learning for recommender systems[C]// Proceedings of the 1<sup>st</sup> Workshop on Deep Learning for Recommender Systems (DLRS 2016). New York, USA;ACM,2016; 7-10.
- [4] LI Ping, WU Qiang, BURGESS C J. Mcrank: Learning to rank using multiple classification and gradient boosting[C]// Advances in Neural Information Processing Systems. New York, USA; NIPS Foundation, 2008;897-904.
- [5] JOACHIMS T. Optimizing search engines using clickthrough data [C]// Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA;ACM, 2002;133-142.
- [6] BURGESS C, SHAKED T, RENSHAW E, et al. Learning to rank using gradient descent[C]// Proceedings of the 22<sup>nd</sup> International Conference on Machine Learning (ICML). Bonn, Germany; ACM, 2005;89-96.

- [7] TAYLOR M, GUIVER J, ROBERTSON S, et al. So rank: optimizing non-smooth rank metrics [C]//Proceedings of the 2008 International Conference on Web Search and Data Mining. New York, USA; ACM,2008; 77-86.
- [8] XU Jun, LI Hang. Adarank: A boosting algorithm for information retrieval[C]// Proceedings of the 30<sup>th</sup> Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Amsterdam; ACM, 2007;391-398.
- [9] PANG Liang, XU Jun, AI Qingyao Ai, et al. Setrank: Learning a permutation-invariant ranking model for information retrieval [C]// Proceedings of the 43<sup>rd</sup> International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM,2020;499-508.
- [10] AI Qingyao, BI Keping, GUO Jiafeng, et al. Learning a deep listwise context model for ranking Refinement[J]. arXiv preprint arXiv:1804.05936,2018.
- [11] CHO K, MERRIENBOER B V, BAHDANAU D, et al. On the properties of neural machine translation: Encoder-decoder approaches[J]. arXiv preprint arXiv:1409.1259,2014.
- [12] ZHUANG Tao, OU Wenwu, WANG Zhirong. Globally optimized mutual influence aware ranking in e-commerce search[J]. arXiv preprint arXiv:1805.08524, 2018.
- [13] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural Computation, 1997, 9 (8) :1735-1780.
- [14] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need. In Advances in Neural Information Processing Systems [C]//Advances in Neural Information Processing Systems. Long Beach, USA; NIPS Foundation, 2017; 5998-6008.
- [15] LV Fuyu, JIN Taiwei, YU Changlong, et al. SDM: Sequential deep matching model for online large-scale recommender system [C]//ACM International Conference on Information and Knowledge Management (CIKM). Beijing; ACM, 2019; 2635-2643.
- [16] HIDASI B, KARATZOGLOU A, BALTRUNAS L, et al. Session-based recommendations with recurrent neural networks [J]. arXiv preprint arXiv:1511.06939,2015.

(上接第 195 页)

- [4] 余胜,陈敬东,王新余. 基于深度学习的复杂场景下车辆识别方法[J]. 计算机与数字工程,2018,46(09):1871-1875,1915.
- [5] LI Ming, ZHANG Li, LI Linlin, et al. Yolo-based traffic sign recognition algorithm [J]. Computational Intelligence and Neuroscience, 2022,2022;2682921.
- [6] YU Shaoyong, WU Yun, LI Wei, et al. A model for fine-grained vehicle classification based on deep learning[J]. Neurocomputing, 2017, 257: 97-103.
- [7] 王林,张鹤鹤. Faster R-CNN 模型在车辆检测中的应用[J]. 计

算机应用,2018,38(03):666-670.

- [8] 王滢暄,宋焕生,梁浩翔,等. 基于改进的 YOLOv4 高速公路车辆目标检测研究[J]. 计算机工程与应用,2021,57(13):218-226.
- [9] 王银,王飞翔,孙前来. 多尺度特征融合车辆检测方法[J]. 系统仿真学报,2022,34(06):1219-1229.
- [10] 李松江,耿兰兰,等. 基于改进 Yolov4 的车辆目标检测[J/OL]. 计算机工程;1-11[2022-10-20]. <https://doi.org/10.19678/j.issn.1000-3428.0062943>.