计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(1):197−205 [doi: 10.15888/j.cnki.csa.008887]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于自注意力网络的时间感知序列化推荐①

孟志鹏, 成卫青

(南京邮电大学 计算机学院, 南京 210023)
通信作者: 孟志鹏, E-mail: 1106039087@qq.com

摘　要: 随着信息技术的发展, 推荐系统作为信息过载时代的重要工具, 正扮演着越来越重要的角色. 基于内容和协同过滤的传统推荐系统, 倾向于以静态方式对用户与商品交互进行建模, 以获取用户过去的长期偏好. 考虑到用户的偏好往往是动态的, 且具有非持续性和行为依赖性, 序列化推荐方法将用户与商品的交互历史建模为有序序列, 能有效捕获商品的依赖关系和用户的短期偏好. 然而多数序列化推荐模型过于强调用户-商品交互的行为顺序, 忽视了交互序列中的时间信息, 即隐式假设了序列中相邻商品具有相同的时间间隔, 在捕捉包含时间动态的用户偏好上具有局限性. 针对以上问题, 文中提出基于自注意力网络的时间感知序列化推荐 (self-attention-based network for time-aware sequential recommendation, SNTSR) 模型, 该模型将时间信息融入改进的自注意力网络中, 以探索动态时间对下一商品预测的影响. 同时, SNTSR 独立计算位置相关性, 以消除可能引入的噪声相关性, 增强捕获用户序列模式的能力. 在两个真实世界数据集上的大量实验表明, SNTSR 始终优于一组先进的序列化推荐模型.

关键词: 推荐系统; 序列化推荐; 自注意网络; 时间信息; 位置嵌入; 注意力机制

引用格式: 孟志鹏,成卫青.基于自注意力网络的时间感知序列化推荐.计算机系统应用,2023,32(1):197–205. http://www.c-s-a.org.cn/1003-3254/8887.html

Time-aware Sequential Recommendation Based on Self-attention Network

MENG Zhi-Peng, CHENG Wei-Qing

(School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: As information technology develops, recommendation system serves as an important tool in the era of information overload and plays an increasingly important role. Traditional recommendation systems based on content and collaborative filtering tend to model the interaction between users and items in a static way to obtain users' previous long-term preferences. Because users' preferences are often dynamic, unsustainable, and behavior-dependent, sequential recommendation methods model the interaction histories between users and items as ordered sequences, which can effectively capture the dependencies between items and users' short-term preferences. However, most sequential recommendation models overemphasize the behavior order of user-item interaction and ignore the temporal information in interaction sequences. In other words, they implicitly assume that adjacent items in the sequences have the same time interval, which leads to limitations in capturing users' preferences that include temporal dynamics. In response to the above problems, this study proposes a self-attention-based network for time-aware sequential recommendation (SNTSR) model, which integrates temporal information into an improved self-attention network to explore the impact of dynamic time on the prediction of the next item. At the same time, SNTSR independently calculates position correlation to eliminate the noise correlations that may be introduced and enhance the ability to capture users' sequential patterns. Extensive experimental studies are carried out on two real-world datasets, and results show that SNTSR consistently outperforms a set of state-of-the-art sequential recommendation models.

① 基金项目: 江苏省研究生教育教学改革课题 (JGZZ19_038)
收稿时间: 2022-05-12; 修改时间: 2022-06-15; 采用时间: 2022-06-27; csa 在线出版时间: 2022-08-26
CNKI 网络首发时间: 2022-11-15

Key words: recommendation system; sequential recommendation; self-attention network; temporal information; position embedding; attention mechanism

数据时代的到来, 海量的信息涌入人类社会, 随之而来的就是信息过载. 推荐系统作为解决此类问题的工具应运而生, 并得到快速发展. 传统的推荐方法包括基于内容的推荐系统和基于协同过滤的推荐系统, 它们更倾向于以静态的方式建模用户行为, 捕获长期偏好. 而序列化推荐方法将用户-商品 (物品) 交互视为动态序列, 考虑序列的依赖关系以获取用户当前和最近的偏好[1], 得到更精准的推荐.

考虑到现实世界的各种因素, 用户的偏好具有动态性, 用户对某一商品的兴趣在不同时间段往往是不同的. 尽管先前序列化推荐方法使推荐效果进一步提升, 但是这仍不足以得到令人满意的结果. 它们过于强调交互的顺序相关性, 这些模型中忽视了关键的时间动态信息, 而这些信息就存在于不断演化的用户-商品交互中, 并与序列模式共存[2]. 这就是说, 这些模型隐含地假设了序列中所有相邻的商品具有相同的时间间隔, 对下一个商品的影响因素只包含前一个商品的位置与编号, 这显然不够合理[3]. 例如, 用户在购买手机之后, 短时间内很有可能购买一个手机壳, 然而过了很长一段时间, 用户可能已经购买过了或是已经不需要了, 对手机壳的购买欲望大大降低.

近年来, 受到 Transformer 模型[4] 的启发, 自注意力网络被用于各种推荐任务中, 并取得了良好的效果. SASRec[5] 对历史交互序列进行建模, 在不使用任何循环或者卷积操作的情况下, 模型表现出优异的推荐性能. BERT4Rec[6] 采用深层双向自注意力对用户行为序列进行建模, 利用 Cloze 任务双向训练目标, 进一步提高了其在 SASRec 上的表现. 自注意力网络为每个历史商品分配一个注意权重并聚合这些商品, 从而推断出商品的序列嵌入关系. 自注意力权重体现了在某个时间戳之前, 商品对当前状态的影响所占比重. 但是, 多数基于自注意力机制的序列化推荐任务也存在弊端. 首先, 自注意力网络要求用户的所有历史商品之间产生交互, 时间与空间复杂度较高, 加大运行成本, 导致推荐效率降低. 其次, 在自注意力机制中, 目前还没有确定的位置嵌入的最优方法, 例如在 Transformer 中, 模型使用了不同频率的正余弦函数进行固定位置嵌入, 而 SASRec 与 BERT4Rec, 将可学习的位置嵌入与商品嵌入相加得到含有位置信息的嵌入; 在 TiSASRec 模型[3] 中, 采用了两个可学习的位置嵌入分别表示自注意力机制中的 key 和 value. 实际上, 商品与绝对位置之间并没有呈现出强相关性, 这种处理可能会引入噪声相关性并限制模型捕获用户序列模式的能力.

综合以上分析, 本文提出了一种基于自注意力网络的时间感知序列化推荐 (self-attention-based network for time-aware sequential recommendation, SNTSR) 模型. 受 TiSASRec 启发, SNTSR 不仅考虑到 LightSANs[7] 中对用户兴趣进行低阶分解, 还将时间信息以不同形式注入到自注意力网络中的 query 和 key 中, 以推断用户在时间动态下的兴趣偏好. 在序列商品的位置嵌入方面, 文中使用独立的位置编码, 单独计算位置相关性, 以避免引入混合相关性[8], 增强模型获取序列模式的能力. 本文的主要贡献总结如下.

(1) 本文根据用户-商品交互序列中隐含的时间动态信息, 将不同时间戳注入模型, 对序列交互中的关系进行建模.

(2) 结合自注意力机制的特点, 提出了一种基于自注意力网络的时间感知序列化推荐方法, 结合交互顺序、时间信息、商品权重以及位置信息共同捕捉序列模式, 进行精准推荐.

(3) 本文提出的 SNTSR 模型在两个真实世界公开数据集上进行了大量实验, 结果表明 SNTSR 模型在两个评价指标方面始终优于一组先进的序列化推荐模型.

1 相关工作

1.1 序列化推荐

早期的序列推荐模型通常利用马尔可夫模型捕获序列模式. 例如, Feng 等[9] 先将马尔可夫链嵌入欧几里德空间, 而后根据其欧几里德距离计算交互之间的转移概率. Rendle 等[10] 提出的 FPMC 模型, 将矩阵分解和马尔可夫链结合来建模序列行为和长期偏好. 此外, 高阶马尔可夫链更多地考虑到先前的商品[11]. 由于马尔可夫链只考虑一个或多个最近的交互, 忽略了长期的依赖, 因此难以捕捉用户的长期偏好. 近年来, 神经

网络逐渐被用于序列化推荐方法. 例如, 基于卷积神经网络的 Caser 模型[12], 将先前的某些商品的嵌入矩阵视为“图像”, 通过卷积神经网挖掘商品间的转移. 循环神经网络对整个用户序列进行建模[13], 得到了广泛应用. 其中, 长短期记忆网络 (LSTM)[14] 和门控循环单元 (GRU)[15] 都是 RNN 的重要变体, 利用各种门控机制保留重要特征, 保证信息在长期传播时不会丢失, 但是模型训练时间可能会很长. 最近, 基于注意力模型[5,6,16] 的序列化推荐方法也开始被探索, 使得序列化推荐性能进一步提升.

1.2 注意力机制

深度学习中的注意力机制与人类视觉的注意力机制类似, 即在诸多信息中把注意力集中在关键信息上, 忽略其他不重要的信息. 注意力模型被广泛使用在自然语言处理[17] 和计算机视觉[18] 等不同类型的深度学习任务中, Transformer 也充分利用自注意力机制, 进一步提高了多头并行性, 在机器翻译任务上取得了较大进展. 为了提高推荐效率, 推荐系统在自注意力机制方面进行了探索, 例如 Kang 等[5] 受 Transformer 启发, 使用自注意力机制对序列化推荐方法进行优化; SHAN[19] 模型使用两层注意力网络, 第 1 层用来学习用户的长期偏好, 第 2 层通过耦合用户长期和短期偏好输出最终用户表示. 在本文中, 受到 LightSANs 模型启发, 其将用户历史商品投影为固定的低阶潜在兴趣, 因此, 每个用户的历史商品只需要与这些固定的潜在兴趣进行交互就能建立真实的上下文感知, 而不是用户的所有历史商品相互交互, 使得模型的时间空间复杂度与用户历史序列长度成正比. 同时, 在对自注意力网络添加位置信息时, 没有采用 SASRec 与 BERT4Rec 中将物品嵌入与位置嵌入相加的方法, 而是单独进行位置相关性的计算, 不需要商品参与, 以减少带给自注意网络的随机性. 文中将时间信息融入到自注意力网络中, 相比之前的注意力网络, SNTSR 模型更具轻量性.

1.3 融合时间信息的推荐

在实际推荐应用中, 用户交互的上下文信息会随着时间而变化, 推荐领域一直在探索如何更充分地利用时间信息.

TimeSVD++模型[20] 将整个时间周期内的时间动态进行建模, 结合因子分解机和邻域模型进行推荐, 取得了显著的效果. Xiong 等[21] 提出的贝叶斯张量分解 (BPTF) 算法, 在传统的基于因子的协同过滤算法中引入时间特征, 从而学习潜在特征的全局演化. 在 MR-TSSSM[22] 中引入时间概念, 将用户特征分为长期兴趣和短期兴趣, 在新闻数据上取得了优异的性能. 此外, 序列化推荐模型也考虑到了时间信息对推荐性能的影响. TIEN[23] 采用了交互时间间隔, 在短时间间隔内, 相似用户的信息有助于识别目标用户的新兴趣. TiSASRec 将个性化时间间隔建模为两商品之间的关系, 对 SASRec 进行改进, 取得不错的效果. Wu 等[16] 提出的基于上下文的时间注意机制 (CTA), 是一种基于序列化神经结构的注意力, 该模型通过学习来衡量用户历史行为的影响, 这不仅包括它是什么行为, 还包括行为发生的时间和方式. 文中我们将时间戳信息进行缩放处理, 将其限制在商品编号的范围之内, 减少了时间戳过大引起的异常信息, 由于对时间戳信息线性缩放, 因此时间信息的损失比较小.

2 SNTSR 模型

2.1 问题陈述

在序列化推荐方法中, 为了便于说明, 设 U 表示一组用户集合, $I=\{i_1, i_2, \cdots, i_{|I|}\}$表示一组商品集合. 对于每一个用户$u \in U$, 都对应着一个历史交互序列$S^u=(S_1^u, S_2^u, \cdots, S_{|S^u|}^u)$, 其中$S_t^u \in I$. 同时, 时间戳序列$T=(t_1, t_2, \cdots, t_{|T|})$与行为序列相对应. 在某一时间戳$t_x$, 模型根据之前交互的 x 个商品以及它们包含的时间信息, 对下一个要交互的商品进行预测. 因此, 若模型的输入是$(S_1^u, S_2^u, \cdots, S_{|S^u|-1}^u)$, 那么期望得到的输出为$(S_2^u, S_3^u, \cdots, S_{|S^u|}^u)$. 结合 LightSANs 方法, 我们考虑将时间信息融入自注意网络中, 更充分地利用上下文信息, 提出 SNTSR 模型以更好地捕捉序列模式, 完成高性能推荐. SNTSR 模型主要分为 3 层: 包含时间信息的注意力层、位置信息嵌入层和输出预测层. 图 1 为 SNTSR 模型总体框架. 同时表 1 列出了文中相关符号及其说明.

2.2 包含时间信息的注意力层

2.2.1 商品嵌入与时间嵌入

受文献 [3] 启发, 文中将输入的行为序列$(S_1^u, S_2^u, \cdots, S_{|S^u|-1}^u)$变换为固定长度为 n 的序列$S=(S_1, S_2, \cdots, S_n)$, 其中 n 表示模型输入序列的最大长度. 若序列长度大于等于 n, 则只考虑前 n 个交互商品; 若序列长度小于 n, 则在左侧添加填充商品直至长度为 n. 同样, 可以得到固定长度为 n 的时间序列$t=(t_1, t_2, \cdots, t_n)$. 之后创建所有物品嵌入矩阵$M \in R^{|I| \times d}$, 其中 d 为潜在维度, 经过查找检索前 n 项操作后得到最终的输入物品 (商品) 嵌入矩阵$E \in R^{n \times d}$, 其中$m_{s_i} \in R^d$.

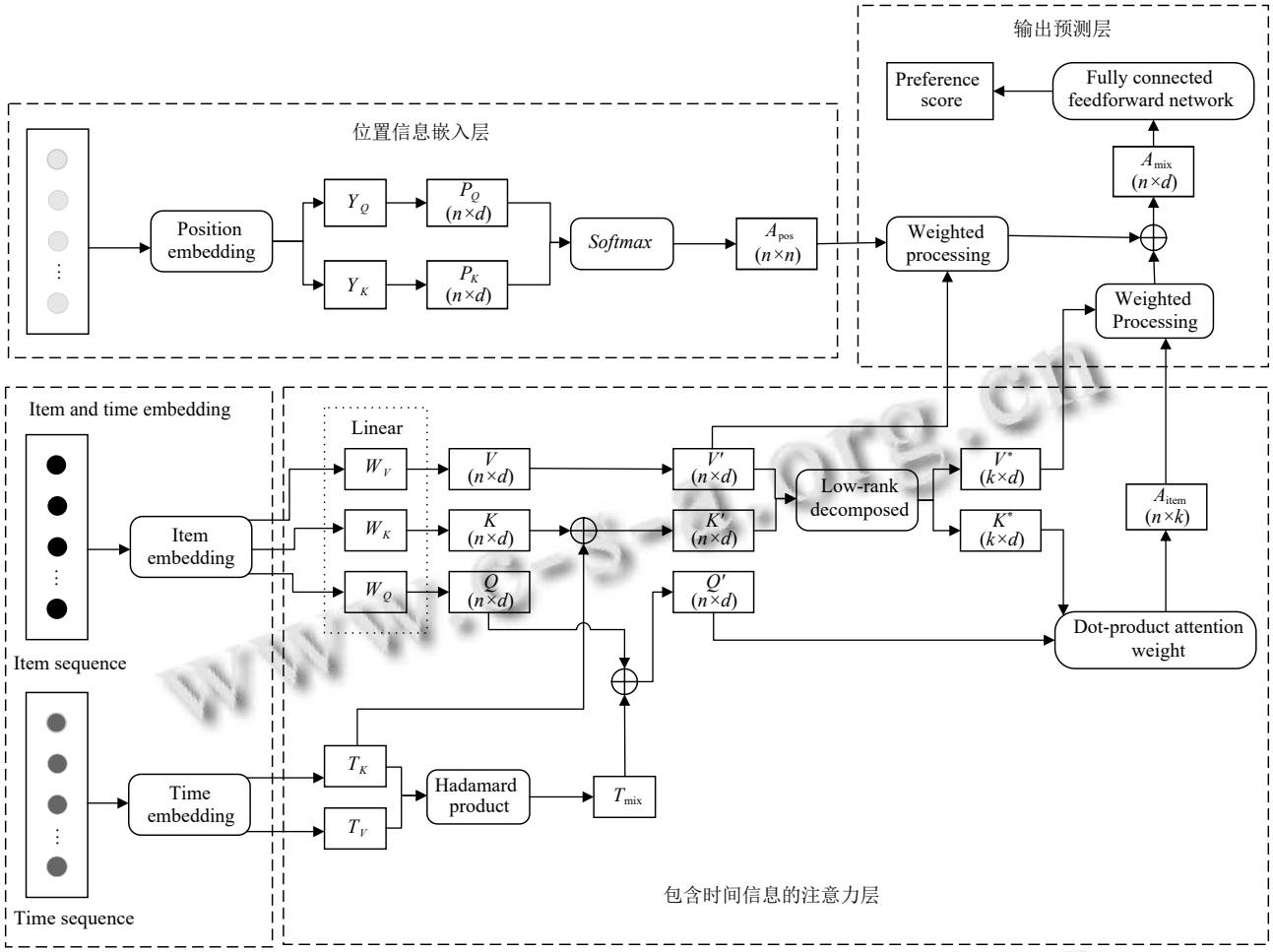

图 1 SNTSR 模型总体框架

$$E=\begin{bmatrix} m_{s_1} \\ m_{s_2} \\ \vdots \\ m_{s_n} \end{bmatrix} \tag{1}$$

为了不同量级的特征能够进行线性计算和加权处理, 避免出现数值计算问题, 我们首先将数值较大的时间戳缩放至商品编号的范围之内. 而后创建两个不同的可学习的时间嵌入矩阵$T_K \in R^{n\times d}, T_V \in R^{n\times d}$, 下标 K 和 V 分别对应自注意力网络中的 key 与 value. 这种方法更适合自注意力网络, 因为它不需要额外的线性变换.

2.2.2 融合时间信息的自注意力网络

商品嵌入矩阵E经过线性投影W_Q, W_K, W_V后得到矩阵Q, K, V分别表示自注意力网络中的 query, key 和 value. 本文对时间矩阵T_K和T_V进行 Hadamard 内积运算, 以捕获更深层的时间相关性:

$$T_{\text{mix}} = T_K \odot T_V \tag{2}$$

而后将Q与T_{mix}相加, 得到融合了时间信息的$Q' \in R^{n\times d}$:

$$Q' = Q + T_{\text{mix}} \tag{3}$$

将K与T_K相加得到融合时间信息的$K' \in R^{n\times d}$, 另外, 保持V'与V相等, 可得到式 (4) 和式 (5):

$$K' = K + T_K \tag{4}$$

$$V' = V \tag{5}$$

接下来, 对兴趣值进行低秩分解以降低参数规模并减少复杂度[7], 即将用户的 n 个历史交互商品投影为 k 个潜在兴趣偏好, 只需要考虑这 k 个潜在兴趣. 结合 LightSANs 模型, 首先对K'低秩分解, 得到$K^* \in R^{k\times d}$, 其中, Θ为可学习参数矩阵.

$$K^* = (Softmax(K' \cdot \Theta^{\mathrm{T}}))^{\mathrm{T}} \cdot K' \tag{6}$$

对V'执行相同的运算操作, 得到兴趣分解后的$V^* \in R^{k\times d}$:

$$V^* = (Softmax(V' \cdot \Theta^{\mathrm{T}}))^{\mathrm{T}} \cdot V' \tag{7}$$

进一步得到具有时间信息的上下文感知注意力权重$A_{\text{item}} \in R^{n\times k}$:

$$A_{\text{item}} = Softmax\left(\frac{Q' \cdot (K^*)^{\mathrm{T}}}{\sqrt{d/h}}\right) \tag{8}$$

其中, h 表示注意力头数, d 为向量的隐藏维度, 采用多头注意力的目的在于更充分地融合用户兴趣与时间信息.

表 1 相关符号

符号	描述信息
U, I	用户集合与商品集合
S^u	用户u的历史交互序列
T	与S^u相对应的时间戳序列
n	输入序列最大长度
d	向量的隐藏维度
h	注意力头数
k	潜在兴趣数量
Θ	可学习的参数矩阵
Q', K', V'	经Q, K, V变换得到
T_K, T_V	可学习的时间嵌入矩阵
T_{mix}	深层时间相关性
P, P_K, P_Q	包含位置信息的嵌入矩阵
A_{item}	包含时间信息的上下文注意力权重
A_{pos}	包含位置信息的注意力权重
A_{mix}	最终上下文信息感知
M	所有物品嵌入矩阵
E	输入商品嵌入矩阵

2.3 位置信息嵌入层

由于自注意网络模型中不包含任何递归或卷积模块, 因此它无法感知先前商品的位置信息, 这里位置信息是指历史交互序列中, 商品之间是存在着前后顺序的, 因此有必要在模型中加入位置信息嵌入模块. 由于商品与绝对位置之间并没有表现出强相关性, 故文中不采用将商品嵌入与位置嵌入相加这一常见方法, 而是单独进行位置相关性的计算. 首先将位置序列进行嵌入, 得到可学习的位置嵌入矩阵$P \in R^{n\times d}$. 之后对初始位置嵌入矩阵进行线性变换:

$$P_Q = P \cdot Y_Q \tag{9}$$

$$P_K = P \cdot Y_K \tag{10}$$

这里$Y_Q \in R^{d\times d}$和$Y_K \in R^{d\times d}$都是可学习的参数矩阵. 接下来利用 $Softmax$ 函数求出位置信息注意力权重$A_{\text{pos}} \in R^{n\times n}$:

$$A_{\text{pos}} = Softmax\left(\frac{P_Q \cdot (P_K)^{\mathrm{T}}}{\sqrt{d/h}}\right) \tag{11}$$

显然, 这里没有任何物品嵌入表示, 位置信息的注意力权重完全独立于物品输入信息, 且在每个输入批次中只需要计算一次, 这有助于减少计算成本.

2.4 输出预测层

2.4.1 最终上下文感知表示

将时间信息注意力权重A_{item}与低秩分解后的V^*相乘, 即对V^*进行加权; 同时, 将位置信息注意力权重A_{pos}与V'相乘, 即对V'进行加权; 最后将赋予权重的V^*与V'线性相加, 得到最终的上下文信息感知表示$A_{\text{mix}} \in R^{n\times d}$:

$$A_{\text{mix}} = A_{\text{item}} \cdot V^* + A_{\text{pos}} \cdot V' \tag{12}$$

2.4.2 输出预测

尽管自注意力机制能够用自适应权重聚合所有先前的商品与时间信息嵌入, 但最终它仍然是一个线性模型. 类似于 Transformer 中的操作[7], 为了赋予模型非线性性质, 本文在每个自注意力网络层A^l_{mix}之后应用一个全连接的前馈网络, 其中包含一个 GELU 激活函数, 可以对神经网络的输入进行随机正则化, 这里上标 l 指的是自注意网络的第 l 层. 相比标准的 ReLU 激活函数, GELU 函数更为平滑, 得到结果F^l如下:

$$F^l = FFN\left(A^l_{\text{mix}}\right) = GELU\left(A^l_{\text{mix}} W_1 + b_1\right) W_2 + b_2 \tag{13}$$

其中, l 代表了当前的注意力层, $W_1, W_2 \in R^{d\times d}$, $b_1, b_2 \in R^d$都是可学习的参数矩阵, 它们在物品之间共享, 但是在不同层之间使用的参数是不同的.

如文献 [5] 中所述, 在堆叠自注意层和前馈网络后, 可能会出现过拟合、参数过多以及梯度消失等问题, 结合文献 [3,5], 文中采用 Layer normalization、Dropout 以及残差连接技术来缓解这些问题. 给定前 t 个物品, 将其进行编码后, 根据第 t 个商品的最后一层输出F^L_t进行下一个商品的预测, L 表示注意力网络的层数. 文中使用内积来评估用户对任意商品 i 的偏好分数, 以预测下一个交互商品:

$$R_{i,t} = \langle F^L_t, M_i \rangle \tag{14}$$

其中, $M_i \in R^d$是单个商品 i 的嵌入向量.

2.4.3 损失函数

文中采用交叉熵损失对模型进行训练:

$$\mathcal{L} = -\log \frac{\exp\left(\langle F^L_t, M_g \rangle\right)}{\sum_{i=1}^{|I|} \exp\left(\langle F^L_t, M_i \rangle\right)} \tag{15}$$

其中, M_g表示真实交互商品的嵌入, $|I|$代表了所有物品数量.

3 实验及分析

3.1 实验设置

3.1.1 数据集

实验所使用的数据集为 3 个真实世界的基准数据集 Amazon Books, Yelp 和 MovieLens-1M (以下简称 ML-1M). 数据集的统计信息如表 2 所示.

表 2 数据集统计信息

Dataset	#Users	#Items	#Actions	Sparsity (%)
Amazon	19214	60707	1733934	99.85
Yelp	56590	75159	2290516	99.94
ML-1M	6040	3629	836478	96.18

Amazon 数据集由文献 [24] 中引入, 其中包括了从 Amazon 网站上抓取的大型商品评价语料库, Amazon 的顶级产品类目被视为独立的数据集, 文中选择其中的图书数据集. 该数据集非常稀疏.

Yelp 是美国商户点评网站, 包括了各地餐饮、购物中心、酒店、旅游等领域的商户, 用户可在 Yelp 网站上对商户进行交流、评论、打分等. Yelp 公开数据集是 Yelp 的业务、评论和用户数据的子集, 用于个人、教育和学术目的, 该数据集同样非常稀疏, 时间跨度非常大.

MovieLens 包含了大量用户对不同电影的评级数据, 同时也包括电影的元数据信息 (电影风格类型、年代等) 和用户属性信息 (年龄、性别、职业等). MovieLens 数据集是一个用于评估协同过滤算法的经典基准数据集, 其包含多个版本, 数据非常密集. 文中我们采用版本 ML-1M 进行实验.

本文采用留一法来评估实验结果, 按照 8:1:1 的比例将数据集划分为训练集、验证集和测试集. 对于每个用户, 将行为序列最后一个交互商品作为测试数据, 将倒数第 2 个交互商品作为验证集, 剩余部分作为训练集, 该方法已在文献 [5,6,8,12] 中得到广泛应用.

3.1.2 评价指标

文中了采用文献 [7] 中两个常用的 Top-N 指标命中率 (hit rate, HR) 和归一化折损累计增益 (normalized discounted cumulative gain, NDCG) 来评估推荐性能, 具体采用 HR@10 和 NDCG@10. HR@10 表示预测正确的商品在 Top-10 列表中出现的比率. 由于每个用户只有一个测试商品, HR@10 将等价于 Recall@10, 且与 Precision@10 成比例[5]. 指标的值越大, 推荐性能越好.

3.1.3 对比模型

本节将本文模型 (SNTSR) 与一组较为先进的序列化推荐模型进行对比, 以证明 SNTSR 引入时间信息的有效性.

(1) POP: 这是一种根据商品的受欢迎度进行商品排名的最简单方法, 这里受欢迎度是根据交互次数来判断的.

(2) FPMC[10]: 该模型将矩阵分解与一阶马尔可夫链相结合, 既能捕获用户的长期偏好, 又能捕获商品间的动态转换.

(3) GRU4Rec[13]: 该模型使用基于排序损失的 GRU 对用户行为序列进行建模, 以获取推荐物品的排序分数, 实现了一种基于会话的推荐.

(4) NARM[25]: 在 GRU 的基础上加入了注意力机制, 用于建模用户在每个会话中的序列行为.

(5) SASRec: 该模型利用自注意力机制, 在每个时间步中自适应地为之前的商品分配权重, 在序列化推荐任务中表现出了优异的推荐性能.

(6) BERT4Rec: 该模型利用深层双向自注意力对用户行为序列进行建模, 并采用 Cloze 任务双向训练目标, 进一步提高了推荐表现.

(7) LightSANs: 该模型引入低秩分解的自注意力机制, 将用户历史行为序列映射成潜在兴趣, 在时间和空间上线性缩放了用户的历史行为序列长度, 缓解了过度参数化的问题.

3.1.4 实现细节

本文所提模型和基线方法都是基于开源推荐算法框架 RecBole[26] 实现的, 文中包含两层自注意力模块, 每层的注意力头数量都为 2. 优化器选择 Adam, 最终在测试集上验证模型效果. 本文所提模型 SNTSR 对应的部分最优超参数设置如表 3 所示.

3.2 实验结果

SNTSR 与以上 6 种基线方法对比的实验结果如表 4 所示, 文中对表现最好的结果进行了加粗. 同时, 将最好的结果与次好的结果对比得到提升率. 可以看出在表 4 中, 基于注意力机制的推荐方法都优于其他模型方法, 这是由于多头自注意力在捕捉用户行为序列模式方面更有效率, 因此上下文感知表示更加准确. 由于 POP 方法仅考虑交互次数来判断商品的受欢迎度, 考虑因素过于单一, 导致推荐效果比其他模型都差. GRU4Rec 和 NARM 模型的推荐性能明显优于 FPMC

方法, 这是因为神经网络方法相比于传统的基于马尔可夫链方法, 前者拥有更多的参数来捕捉高阶转换以及更强的能力以获取长期序列模式. 而 NARM 方法的性能优于 GRU4Rec, 是因为它使用了自注意力机制建模序列行为. 另外, SASRec 和 BERT4Rec 都采用了自注意力机制进行序列化推荐, 与前者从左向右编码用户历史交互信息不同的是, BERT4Rec 采用双向自注意力对序列进行建模, 它们的性能都要优于 NARM 模型. LightSANs 引入低阶分解的自注意力, 将用户的历史商品映射到固定数值的潜在兴趣, 并独立进行位置编码, 使得模型更加轻量化, 因此其性能要优于 SASRec 和 BERT4Rec.

本文提出的 SNTSR 模型与其他模型相比, 其评估指标 HR@10 和 NDCG@10 都为最高值, 而且在数据集 Amazon Books 与 Yelp 上有比较明显的提升, 这表明用户行为序列中隐含的时间信息是非常重要的. 引入上下文时间信息, 模型可以学习到更多的序列模式, 提高推荐效果. 与表 4 中次优方法 LightSANs 相比, SNTSR 以多种方式融合时间信息, 整个自注意力网络部分既包含了商品嵌入信息, 又融入了可学习的时间信息, 更充分地利用了上下文信息. 再将注入了时间信息的用户历史商品投影为常数个低阶潜在兴趣, SNTSR 方法将注意力矩阵的复杂度由 $O(n^2)$ 变为 $O(nk)$, 模型性能得到进一步提升.

表 3 超参数设置

Dataset	Learning rate	Batch size	Maximum sequence length	Hidden dimension	Latent interests	Dropout rate	Layer normalization
Amazon	0.000 3	1 024	150	64	15	0.5	1E−12
Yelp	0.001	1 024	100	64	10	0.5	1E−12
ML-1M	0.003	512	100	64	25	0.2	1E−12

表 4 不同模型的性能比较 (%)

Dataset	Metric	Pop	FPMC	GRU4Rec	NARM	SASRec	BERT4Rec	LightSANs	SNTSR	Improv.
Amazon	HR@10	3.95	7.97	8.08	8.16	8.43	8.10	8.86	**27.28**	207.90
	NDCG@10	1.56	3.98	4.02	4.12	4.14	4.03	4.36	**15.93**	265.37
Yelp	HR@10	1.70	2.31	4.37	4.49	5.09	4.89	5.55	**17.41**	213.69
	NDCG@10	0.82	1.08	2.15	2.29	2.76	2.62	2.91	**9.40**	223.02
ML-1M	HR@10	8.15	12.32	21.30	21.75	22.11	21.99	22.56	**23.32**	3.37
	NDCG@10	4.04	5.89	10.91	10.98	11.21	10.99	11.45	**12.47**	8.91

3.3 消融实验

考虑到本文模型引入序列中隐含的时间信息, 并独立计算商品位置相关性, 因此我们通过消融实验对模型的关键组件的有效性进行分析. 本文对两个部分进行了消融实验, 实验结果如表 5 所示, 同样我们对表现最好的方法进行了加粗. 我们将 SNTSR 位置编码部分去掉, 保留时间信息, 得到的新模型表示为 SNTSR-P. 另一方面, 我们将 SNTSR 去掉时间信息之后得到的模型表示为 SNTSR-T. 为了公平比较, 两个新模型的超参数设置与 SNTSR 保持一致.

从表 5 中可以看出, SNTSR-P 模型效果逊于 SNTSR, 限制其性能的主要原因可能是, 由于一个用户历史行序列中为可能对应着许多相同的时间戳 (某些行为只有一个时间戳), 导致模型退化为没有任何位置信息的自注意力网络, 模型无法感知先前行为序列的位置信息, 推荐性能降低. SNTSR-T 模型性能最差, 相比于 SNTSR-P, SNTSR-T 性能急剧下降, 这表明时间信息对于提升模型性能起到重要作用. 性能最优的是本文提出的 SNTSR 模型, 它合并了更加丰富的用户-商品关系来计算注意力权重, 将时间信息融入注意力网络之中, 同时进行独立位置编码, 推荐效果得到提高.

表 5 消融实验分析

Dataset	Metric	SNTSR-P	SNTSR-T	SNTSR
Amazon	HR@10	0.261 8	0.094 9	**0.272 8**
	NDCG@10	0.151 9	0.045 5	**0.159 3**
Yelp	HR@10	0.164 1	0.056 3	**0.174 1**
	NDCG@10	0.086 6	0.029 8	**0.094 0**
ML-1M	HR@10	0.222 8	0.204 5	**0.233 2**
	NDCG@10	0.122 1	0.102 0	**0.124 7**

3.4 重要超参数影响分析

本节主要讨论文中所提方法涉及的一些重要超参数对模型效果的影响, 包括学习率 (learning rate) 和潜在兴趣 (latent interests), 由于 Yelp 与 Amazon 都非常稀疏, 因此我们选择 ML-1M 和 Amazon 数据集进行研究.

(1) 学习率. 对于学习率的每个取值, 其他超参数保持最优设置不变. 学习率的具体取值集合是{0.000 01,

0.000 3, 0.001, 0.003, 0.01, 0.1}, 图 2 给出了不同学习率对 HR 的影响效果. 可以看出, 两个数据集 HR@10 的变化趋势整体一致, ML-1M 和 Amazon在学习率分别为 0.003 与 0.000 3 时取得最佳表现, 均在 0.1 处表现较差, 原因是过大的学习率可能导致收敛困难, 甚至直接跳过最优值.

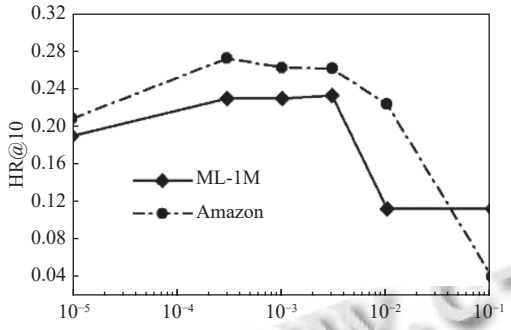

图 2 不同学习率对 HR 的影响

(2) 潜在兴趣. 文中采用低阶分解自注意力将用户的历史商品映射到固定数值的潜在兴趣, 因此, 采用不同数值的潜在兴趣, 可能会产生不同的推荐结果. 实验选取该参数的集合为{5, 10, 15, 20, 25, 30}, 其他超参数保持最优设置不变, 实验结果如图 3所示.

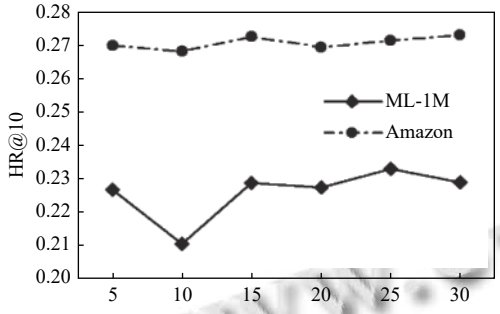

图 3 不同潜在兴趣值对 HR 的影响

可以看出, Amazon 数据集的 HR@10 指标在 0.27 附近波动, 总体处于稳定, 在潜在兴趣为 15 时推荐性能. 而 ML-1M 数据集在潜在兴趣值为 5 时, HR@10 值最小, 之后随着潜在兴趣值的变大呈上升趋势, 并在潜在兴趣值为 25 时达到最大. 在同一潜在兴趣下, Amazon 的 HR@10 要远高于 ML-1M, 其原因可能是两数据集的稀疏程度相差较大, 稀疏的历史商品映射到固定潜在兴趣相比密集的用户商品映射到固定潜在兴趣, 其变化更小.

4 结束语

为了提高推荐系统的性能, 本文提出 SNTSR 模型探究了融合时间信息的自注意网络对序列化推荐的影响. 首先, 考虑到用户历史商品之间的相互交互会带来过高的时间空间复杂度, 对历史商品交互进行低秩分解, 投影到常数个潜在兴趣上, 每个用户的历史商品只需要与这些潜在兴趣进行交互就能建立真实的上下文感知, 降低了参数规模并和复杂度, 同时优化了自注意力网络. 其次, 将隐含于用户交互序列中的时间信息注入自注意网络之中, 丰富了上下文信息, 提高推荐性能. 最后, 对位置信息独立进行编码, 解耦了位置嵌入, 使模型拓展性进一步提升.

目前, 虽然模型运行时的时空复杂度得到降低, 但是模型的规模较大, 如何在不提高时空复杂度的基础上进一步缩小模型规模是下一步的研究重点. 同时, 未来的工作还考虑提取更多的存在于用户交互序列中的上下文信息, 增加对用户交互行为的刻画, 将其注入模型以进一步提升推荐的准确性.

参考文献

1 Wang SJ, Hu L, Wang Y, *et al*. Sequential recommender systems: Challenges, progress and prospects. Proceedings of the 28th International Joint Conference on Artificial Intelligence. Macao: IJCAI, 2019. 6332–6338.

2 Fan ZW, Liu ZW, Zhang JW, *et al*. Continuous-time sequential recommendation with temporal graph collaborative transformer. Proceedings of the 30th ACM International Conference on Information & Knowledge Management. Queensland: ACM, 2021. 433–442.

3 Li JC, Wang YJ, McAuley J. Time interval aware self-attention for sequential recommendation. Proceedings of the 13th ACM International Conference on Web Search and Data Mining. Houston: ACM, 2020. 322–330.

4 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: ACM, 2017. 6000–6010.

5 Kang WC, McAuley J. Self-attentive sequential recommendation. Proceedings of the 18th IEEE International Conference on Data Mining (ICDM). Singapore: IEEE, 2018. 197–206.

6 Sun F, Liu J, Wu J, *et al*. BERT4Rec: Sequential recommendation with bidirectional encoder representations

from transformer. Proceedings of the 28th ACM International Conference on Information and Knowledge Management. Beijing: ACM, 2019. 1441–1450.

7 Fan XY, Liu Z, Lian JX, *et al*. Lighter and better: Low-rank decomposed self-attention networks for next-item recommendation. Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. Montreal: SIGIR, 2021. 1733–1737.

8 Ke GL, He D, Liu TY. Rethinking positional encoding in language pre-training. arXiv:2006. 15595, 2020.

9 Feng SS, Li XT, Zeng YF, *et al*. Personalized ranking metric embedding for next new POI recommendation. Proceedings of the 24th International Joint Conference on Artificial Intelligence. Buenos Aires: IJCAI, 2015. 2069–2075.

10 Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next-basket recommendation. Proceedings of the 19th International Conference on World Wide Web. Raleigh: ACM, 2010. 811–820.

11 He RN, McAuley J. Fusing similarity models with Markov chains for sparse sequential recommendation. Proceedings of the 2016 IEEE 16th International Conference on Data Mining. Barcelona: IEEE, 2016. 191–200.

12 Tang JX, Wang K. Personalized top-N sequential recommendation via convolutional sequence embedding. Proceedings of the 11th ACM International Conference on Web Search and Data Mining. Marina Del Rey: ACM, 2018. 565–573.

13 Hidasi B, Karatzoglou A, Baltrunas L, *et al*. Session-based recommendations with recurrent neural networks. arXiv:1511.06939, 2015.

14 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780. [doi: 10.1162/neco.1997.9.8.1735]

15 Luo AJ, Zhao PP, Liu YC, *et al*. Adaptive attention-aware gated recurrent unit for sequential recommendation. Proceedings of the 24th International Conference on Database Systems for Advanced Applications. Chiang Mai: Springer, 2019. 317–332.

16 Wu JB, Cai RQ, Wang HN. Déjà vu: A contextualized temporal attention mechanism for sequential recommendation. Proceedings of the 29th International Conference on World Wide Web. Taipei: ACM, 2020. 2199–2209.

17 Shen T, Zhou TY, Long GD, *et al*. DiSAN: Directional self-attention network for RNN/CNN-free language understanding. Proceedings of the 32nd AAAI Conference on Artificial Intelligence. New Orleans: AAAI, 2018. 5446–5455.

18 Zhang WF, Yu J, Hu H, *et al*. Multimodal feature fusion by relational reasoning and attention for visual question answering. Information Fusion, 2020, 55: 116–126. [doi: 10.1016/j.inffus.2019.08.009]

19 Ying HC, Zhuang FZ, Zhang FZ, *et al*. Sequential recommender system based on hierarchical attention network. Proceedings of the 27th International Joint Conference on Artificial Intelligence. Stockholm: IJCAI, 2018. 3926–3932.

20 Koren Y. Collaborative filtering with temporal dynamics. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris: ACM, 2009. 447–456.

21 Xiong L, Chen X, Huang TK, *et al*. Temporal collaborative filtering with Bayesian probabilistic tensor factorization. Proceedings of the 2010 SIAM International Conference on Data Mining. Columbus: SDM, 2010. 211–222.

22 Song Y, Elkahky AM, He XD. Multi-rate deep learning for temporal recommendation. Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. Pisa: ACM, 2016. 909–912.

23 Li X, Wang C, Tong B, *et al*. Deep time-aware item evolution network for click-through rate prediction. Proceedings of the 29th ACM International Conference on Information & Knowledge Management. Online: ACM, 2020. 785–794.

24 McAuley J, Targett C, Shi QF, *et al*. Image-based recommendations on styles and substitutes. Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. Santiago: ACM, 2015. 43–52.

25 Li J, Ren PJ, Chen ZM, *et al*. Neural attentive session-based recommendation. Proceedings of the 26th ACM International Conference on Information and Knowledge Management. Singapore: ACM, 2017. 1419–1428.

26 Zhao WX, Mu SL, Hou YP, *et al*. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. Proceedings of the 30th ACM International Conference on Information & Knowledge Management. Online: ACM, 2021. 4653–4664.

(校对责编: 牛欣悦)