带有时间预测辅助任务的会话式序列推荐

陈 聪 张 伟 王 骏
(华东师范大学计算机科学与技术学院 上海 200062)

摘 要 会话式序列推荐旨在根据短期匿名行为序列预测该用户近期行为,因为该任务设定考虑了用户偏好的变化而备受学术和工业界关注.现有方法集中于以单任务模式预测下一交互对象,忽略了行为时间预测辅助任务中的额外语义.在面向事件、地点的一般序列推荐问题中,有少数方法以并行方式同时预测下一交互对象及对应时间,然而这并不完全契合用户先产生交互意图再选择合适时间的实际情况.为缓解上述问题影响,本文提出了一种基于序列式多任务学习的会话式序列推荐方法,具有两方面特色.首先,该方法通过将下一交互对象预测结果作为下一时间预测的输入,赋予两个任务顺序依赖性.相比于传统的并行式交互对象和时间预测方式该方法耦合性更强.其次,本文开发了一种经过改进的双向时间间隔感知自注意力方法,使得会话中每个位置可以融合来自左右两侧的交互对象及时间间隔信息.相比于过去的单向会话式序列推荐方法增强了建模会话上下文的能力,该方法有利于更好地刻画用户兴趣表征.本文在 Tianchi 电商数据集、Lastfm 音乐数据集以及 Foursquare 地点行为轨迹数据集上进行了实验.结果表明:(1)所提出方法在常用序列推荐指标上一致优于所采用的比较方法.特别地,在 *NDCG*@5 评价指标上比最好的基线方法 TiSASRec 平均提高 13.51%;(2)序列式多任务学习和双向时间间隔感知自注意力机制对于预测性能均能带来正面提升.

关键词 会话式序列推荐;交互对象预测;时间预测;序列式多任务学习;自注意力网络
中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2021.01841

Session-Based Sequential Recommendation with Auxiliary Time Prediction

CHEN Cong ZHANG Wei WANG Jun
(*School of Computer Science and Technology, East China Normal University, Shanghai* 200062)

Abstract Session-based sequential recommendation aims to predict a target user's behavior in the near future based on its anonymous short-term behavior sequence in a given current session. Since this task setting provides a good chance of considering user dynamic preference, it has gained much attention from both the academic and industrial domains. Most of the existing relevant methods in the literature are attributed to the category of only focusing on predicting the next items to interact for a considered user in a single-task mode, which inevitably overlooks the additional useful semantic information that is contained in the auxiliary task of predicting user behavior time. For general sequential recommendation problems that take events or locations as the input, there exist a few approaches that investigate the ways of predicting the next item and its corresponding time at the same time in a parallel fashion. However, this paradigm is not very consistent with the actual situation of sequential behaviors, where users tend to first determine the intention of an interaction and then choose the appropriate time in the future to perform the interaction. To mitigate the effect of the above issue, this paper develops a novel session-based

收稿日期:2020-08-30;在线发布日期:2020-12-25.本课题得到国家重点研发计划项目(2019YFB2102600)、国家自然科学基金项目(61702190,61672236)、人工智能教育部重点实验室开放基金资助.**陈 聪**,硕士研究生,主要研究方向为数据挖掘.E-mail:51194501031@stu.ecnu.edu.cn.**张 伟**(通信作者),博士,副研究员,中国计算机学会(CCF)会员,主要研究方向为数据挖掘.E-mail:zhangwei.thu2011@gmail.com.**王 骏**(共同通信作者),博士,教授,主要研究领域为计算机视觉.E-mail:wangjun@gmail.com.

sequential recommendation method by utilizing the framework of sequential multi-task learning. The devised method possesses two aspects of innovations. First of all, it predicts the interacted items in the next time and further take them as the input of the next time prediction method, thereby this manner empowers the two prediction tasks with sequential dependency. As a result, compared with the parallel fashion of the joint prediction of items and their corresponding timestamps, the proposed model can couple the two tasks more tightly. Secondly, an improved bidirectional time interval aware Self-attention approach is developed in this paper, which enables the item at each position of a target session to have the sequential context information, i.e., the other interacted items and the time intervals between the current item and the other items, from both the left and right sides of the same session. As such, compared with the conventional session-based sequential recommendation approaches that model each session in a one-way mode, this proposed bidirectional approach is welcomed for enhancing the ability of modeling the contextual information of sessions, which in turn obtains more precise user interest representations for later prediction. The comprehensive experiments in this paper are conducted on three publicly available datasets with different origins, including the Tianchi dataset for the e-commerce scenario, the Lastfm dataset used in music listening, and the Foursquare dataset that is generated by spatio-temporal trajectories. The experimental results on the three datasets reveal that: (1) The proposed method improves the adopted strong baselines consistently in terms of different evaluation metrics commonly used in personalized sequential recommendation. In particular, it outperforms the best baseline method, i.e., TiSASRec, on average by 13.51% in terms of the *NDCG*@5 evaluation metric. (2) Both the sequential multi-task learning part and bidirectional time interval aware Self-attention mechanism can bring positive improvements over the next item and time prediction performance.

Keywords session-based sequential recommendation; item prediction; time prediction; sequential multi-task learning; Self-attention networks

1 引 言

会话式序列推荐(Session-based Sequential Recommendation)是推荐系统中的重要研究方向,其适用场景主要体现在当用户处于匿名状态或者用户的长期交互行为未被记录时,推荐系统只能依靠用户在当前会话中的短期行为序列产生推荐.例如,因为用户隐私问题,电商平台在很多情况下无法跟踪用户的长期浏览足迹,从而依赖于当前会话捕捉用户的短期购买意图,预测用户感兴趣的目标商品;另外一个常见的例子是,用户在未登录状态下使用音乐软件,虽然无法确定用户身份,但平台仍然可以根据用户在当前会话中的短期听歌记录推荐符合其兴趣偏好的相关音乐.由于广泛存在的应用场景和较高的实用价值,近年来该方向得到了越来越多研究者的关注.

在早期研究中,基于马尔可夫链(Markov Chain,MC)[1-2]的推荐方法被用于会话式序列推荐中.但由于仅考虑最近行为的影响,该类方法无法从会话序列整体角度建模用户兴趣,因此预测准确性面临瓶颈.近年来,深度学习在会话式序列推荐中逐渐成为主流方法,主要思路为将整个会话建模为序列,并通过深度神经网络从中捕捉用户兴趣以及交互对象间的依赖关系,以实现下一交互对象的预测.Hidasi等人[3]首次将循环神经网络(Recurrent Neural Network,RNN)应用到会话式序列推荐中,效果相比传统方法取得显著提升;Li等人[4]利用RNN和注意力机制(Attention Mechanism,Attention)建模用户的序列行为和主要目的;Liu等人[5]则利用注意力机制和多层感知机(Multilayer Perceptron,MLP)捕捉用户的整体兴趣和当前兴趣;Wu等人[6]将会话建模为图结构,试图利用图神经网络(Graph Neural Network,GNN)学习交互对象表示.

虽然上述深度学习方法在会话式序列推荐任务中取得了不错的性能，但仍然存在一些因素的制约.

首先，现有方法多以单任务模式预测下一交互对象，忽略了下一时间预测辅助任务中的额外语义. 实际上，用户行为不仅涉及到交互对象序列，还包含每次交互对应的具体时间戳信息. 因此，如何使模型能够同时利用交互对象和时间信息作为监督信号进行训练对于提高序列推荐质量十分关键. 少数前人[7-8]在这方面有所探索，提出序列神经网络与时序点过程[9]融合的思路，以并行式多任务学习方式[10]同时预测交互对象及时间. 这种方式虽然简单直接，但与现实场景中用户先产生交互意图再选择合适时间的实际情况不完全契合. 例如，用户在电商平台购买手机后，出于保护和装饰需要，很可能会接着挑选手机壳；而对于蔬菜、水产等生鲜产品，出于食材新鲜程度考虑，用户大多只会购买当日所需，次日再购买其他生鲜，这样相邻两次行为时间间隔较大. 因此，下一交互对象不同，用户偏好的交互时间也存在差异.

其次，现有方法大多只考虑会话序列中交互对象时间戳信息所蕴含的序列模式，忽视了交互对象之间的时间间隔对于表征用户兴趣的作用. 直观上，由于会话序列中相邻交互对象的时间间隔不同，具有更近时间戳的交互对象不仅更能反映用户当前兴趣[11]，而且对于时间临近的交互对象的影响也会越大. 因此，考虑交互对象之间的时间间隔有利于对序列中的每个交互对象实现精细化建模，进而推动用户会话序列的整体表征效果.

为解决以上问题，本文在会话式序列推荐中引入了时间预测辅助任务，以序列式多任务学习方式依次预测下一交互对象及时间，即将下一交互对象预测结果作为下一时间预测的输入. 这种策略不仅引入了时间知识促进模型学习，而且使得交互对象和时间的预测顺序也更贴近现实场景. 另外，最近 Li 等人[11]在序列推荐任务中将时间间隔信息融入到自注意力机制(Self-Attention Mechanism，Self-Attention)，单向建模交互对象之间的时间间隔对预测的影响. 本文将其推广到会话式序列推荐并改进为双向时间间隔感知自注意力机制(Bidirectional Time Interval Aware Self-Attention)，使得会话中每个位置可以融合来自左右两侧的交互对象及时间间隔信息，增强了建模会话上下文的能力，可以更为准确地从会话序列整体角度刻画用户兴趣表征.

总体而言，本文的贡献总结如下：

(1) 提出了一种针对会话式序列推荐任务的新型序列式多任务深度推荐模型，通过时间预测辅助任务促进交互对象的预测，达到了提升推荐性能的目的.

(2) 开发了一种经过改进的双向时间间隔感知自注意力方法，赋予模型显式建模用户会话中每个位置与其左右两侧对象交互的能力，并能有效融入其对应时间间隔信息.

(3) 在 Tianchi 电商数据集、Lastfm 音乐数据集、Foursquare 地点数据集进行了实验. 实验结果表明，所提出方法比基线方法取得显著提升. 另外也验证了以序列式多任务学习方式组织主辅任务、双向时间间隔感知自注意力机制等核心部分的有效性.

2　相关工作

2.1　基于深度学习的会话式序列推荐

近年来，深度学习被证明是有效的序列数据建模方法，RNN、Attention、GNN 等相关技术均在会话式序列推荐中得到了广泛应用. Hidasi 等人[3]首次尝试使用 RNN 变体门控循环单元(Gate Recurrent Unit，GRU)建模会话序列，Tan 等人[12]在此基础上通过数据增强技术和考虑用户行为的时序变化进一步提高模型性能；Li 等人[4]在利用 GRU 建模会话序列的同时利用 Attention 计算不同交互对象权重，捕捉用户在当前会话中的主要目的；Liu 等人[5]利用 Attention 和 MLP 完全替代 RNN，并强调了最近一次交互对反映用户当前兴趣的重要作用；Wu 等人[6]将图结构应用到会话序列，建模交互对象间的复杂关系，Qiu 等人[13]则在会话图中捕捉交互对象转换模式的内在顺序，最近 Wang 等人[14]联合考虑了目标行为序列和辅助行为序列，通过构建多关系会话图从全局角度学习交互对象之间的关系，虽然考虑了将另一种行为作为辅助信息提高推荐质量，但是也没有在利用时间信息上做出更多探索；Kang 等人[15]受 Self-attention 在自然语言处理领域取得优异表现的启发，在序列推荐中利用 Self-attention 替代 RNN 建模用户行为序列，无论是序列模式挖掘还是并行计算能力，Self-attention 均表现出了巨大优势，该方法同样能够适用于会话式序列推荐.

虽然上述方法取得了显著效果，但是忽略了时间预测辅助任务可学习到的额外语义，也没有考虑交互对象之间的时间间隔对于表征用户兴趣的重要

作用，本文在这两个方面做出了更多探索.

2.2 时序点过程

时序点过程(Temporal Point Process)是由连续时间域上一系列离散事件构成的随机过程，作为时间预测最流行的方法之一，在许多工作中和深度学习技术紧密结合. Du 等人[9]使用 RNN 建模历史事件整体表示，将时序点过程的强度函数设计为历史事件的非线性函数；Mei 等人[16]提出了神经霍克斯过程(Neural Hawkes Process)，利用长短期记忆(Long Short-Term Memory，LSTM)学习历史事件的累积信息，由事件之间的相互影响动态估计下一事件的类型和发生时间；最近时序点过程也被应用到地点预测问题的时间预测任务中，其中 Liang 等人[8]将用户长期偏好、动态兴趣和时序点过程组织成了统一的体系结构，同时预测下一地点及时间. 本文将在时间预测辅助任务中利用时序点过程预测下一交互对象发生时间.

时序点过程通常利用条件强度函数 $\lambda(t)$ 建模下一事件发生时间[17]. $\lambda(t)\mathrm{d}t = P\{t_{n+1} \in [t, t+\mathrm{d}t) \mid H_t\}$ 表示在历史事件 H_t 基础上下一事件将在小的时间窗口 $[t, t+\mathrm{d}t)$ 内发生的概率. 设 $f(t)$ 为条件密度函数，$F(t)$ 为上次事件在 t_n 发生以来下一事件会在 t 之前发生的累积概率，$\lambda(t)$ 可指定为

$$\lambda(t)\mathrm{d}t = \frac{f(t)\mathrm{d}t}{(1-F(t))} \tag{1}$$

由以上公式，$f(t)$ 可表示为

$$f(t) = \lambda(t)\exp\left(-\int_{t_n}^{t}\lambda(\epsilon)\mathrm{d}\varepsilon\right) \tag{2}$$

不同时序点过程取决于条件强度函数 $\lambda(t)$ 的不同选择. 在 4.3.2 节中，本文将 $\lambda(t)$ 设计为相关语义表示的非线性函数.

2.3 多任务学习

多任务学习[18]致力于利用多个相关任务同时学习，以提升所有或者目标任务性能. 多任务学习在推荐系统领域应用较为广泛. Hadash 等人[19]将推荐系统领域中的排序任务和评分预测任务通过多任务学习进行结合. Lu 等人[20]通过文本生成任务产生的输出，为评分预测任务提供可解释性. 最近，腾讯公司提出了渐进式分层抽取模型，实现了基于共享知识和任务特定知识的渐进式路由机制，并在腾讯视频推荐数据集上验证了其优越性[21].

与本文最相关的多任务序列推荐工作[8-9]将时间预测和事件/地点预测融合到一个统一的多任务学习框架中. 然而，他们均采用并行式多任务学习方式，即利用模型得到的序列单个语义表示同时预测下一交互对象及对应时间，忽略了现实场景中用户往往先产生交互意图再选择合适时间的实际情况. 因此，本文侧重于通过序列式多任务学习解决这一问题.

3 问题定义

本节对会话式序列推荐进行形式化定义，包括所研究问题的正式描述和相关符号含义说明，并给出了本文使用的主要符号定义.

设数据集中交互对象集合 $\mathcal{V} = \{v_1, v_2, \cdots, v_m\}$，其中 m 为交互对象个数. 会话序列定义为 $S = [s_1, s_2, \cdots, s_n]$，其中 n 为序列长度，$s_i \in \mathcal{V}$ 按照时间顺序排列. S 对应的时间戳序列 $T = [t_1, t_2, \cdots, t_n]$，$t_i$ 为 s_i 的发生时间. 会话内交互对象之间的时间间隔定义为矩阵 $\boldsymbol{R}$，其中 $r_{i,j} = |t_i - t_j|$.

所研究问题可正式描述为：给定会话序列 S 以及交互对象之间的时间间隔 $\boldsymbol{R}$，预测下一交互对象及时间 (s_{n+1}, t_{n+1}). 其中通过间接预测下一交互对象发生的时间间隔 τ_{n+1}，得到 $t_{n+1} = t_n + \tau_{n+1}$.

本文使用的主要符号定义如表 1 所示.

表 1 符号定义

符号	描述
S, T	会话序列和时间戳序列
m	交互对象个数
n	序列长度
d	语义向量维度
$\boldsymbol{R}$	时间间隔矩阵
$\boldsymbol{E}^I$	交互对象嵌入矩阵
$\boldsymbol{E}^G$	时间间隔嵌入矩阵
$\boldsymbol{X}$	交互对象输入嵌入矩阵
$\boldsymbol{M}^K, \boldsymbol{M}^V$	时间间隔输入嵌入矩阵
$\boldsymbol{P}^K, \boldsymbol{P}^V$	位置嵌入矩阵

4 模 型

在本节中，详细介绍本文提出的以序列式多任务学习方式预测下一交互对象及时间的会话式序列推荐方法. 首先，给出对交互对象序列和时间间隔进行预处理并在嵌入层映射为稠密向量表示的方法. 然后，说明利用双向时间间隔感知自注意力网络对会话序列建模的具体细节. 接着，介绍如何以序列式多任务学习方式依次预测下一交互对象及时间. 最后，给出对主辅任务进行联合训练的方法. 模型结构如图 1 所示.

《计算机学报》

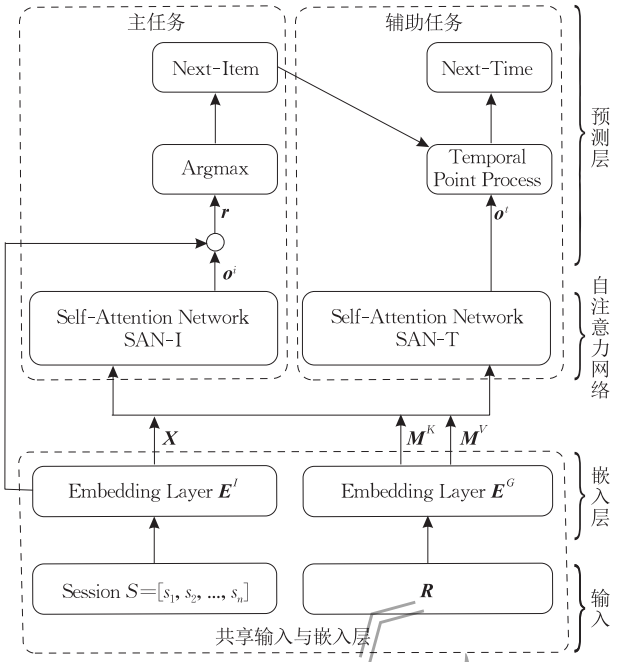

图 1　模型结构

4.1　嵌入层

嵌入层负责将输入的会话序列和时间间隔映射为稠密向量表示. 首先根据集合 $\mathcal{V}$，创建交互对象嵌入矩阵 $\boldsymbol{E}^I \in \mathbb{R}^{m\times d}$，其中 d 为语义向量维度. 对于会话序列，经填充 0 或保留最近交互将其处理为长度为 n 的定长序列，并通过 $\boldsymbol{E}^I$ 映射为矩阵 $\boldsymbol{X} \in \mathbb{R}^{n\times d}$，填充位置均为全零向量.

对于时间间隔 $\boldsymbol{R}$，根据 Li 等人[11]的方法，时间戳序列 T 的填充部分取值与第一个非填充位置相同，由 T 得到 $\boldsymbol{R} \in \mathbb{N}^{n\times n}$. 经过 $r_{i,j}=\lfloor r_{i,j}/\min(\boldsymbol{R})\rfloor$ 缩放和 $r_{i,j}=\min(a, r_{i,j})$ 裁剪处理，$\boldsymbol{R}$ 中元素被限定为 0 到 a 之间的自然数，其中 a 为控制时间间隔取值上限的超参数. 最后创建时间间隔嵌入矩阵 $\boldsymbol{E}^G \in \mathbb{R}^{2\times(a+1)\times d}$，第一个维度为 2 的原因是要将处理后的 $\boldsymbol{R}$ 映射为 $\boldsymbol{M}^K \in \mathbb{R}^{n\times n\times d}$ 和 $\boldsymbol{M}^V \in \mathbb{R}^{n\times n\times d}$，分别用于自注意力计算中的 key 和 value.

由于自注意力计算需要对位置信息进行显式编码，这里采取的方法是将可学习的位置嵌入矩阵 $\boldsymbol{P}^K \in \mathbb{R}^{n\times d}$ 和 $\boldsymbol{P}^V \in \mathbb{R}^{n\times d}$ 分别用于自注意力计算中的 key 和 value.

4.2　自注意力网络

近年来，Transformer 模型[22]及其变体在许多自然语言处理任务中取得最佳表现，其使用的自注意力机制也随之被大量应用于序列推荐中. 在本文工作中，为了学习到主辅任务之间的语义差异，利用两个自注意力网络分别对会话序列建模，定义为 SAN-I 和 SAN-T，两个网络结构一致，参数不同，共享模型输入与嵌入层. 另外，受 Li 等人[11]工作启发，本文开发了一种经过改进的双向时间间隔感知自注意力方法，会话中每个位置可以融合来自左右两侧的交互对象及时间间隔信息. 由于 SAN-I 和 SAN-T 的网络结构相同，下面将统一阐述. 网络结构如图 2 所示.

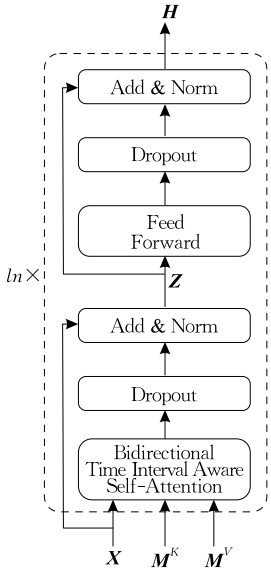

图 2　自注意力网络

4.2.1　双向时间间隔感知自注意力

Transformer 中使用的自注意力机制可定义为

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d}}\right)\boldsymbol{V} \tag{3}$$

其中 $\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$ 分别表示 query、key、value，缩放因子 $\sqrt{d}$ 可防止内积过大，d 为语义向量维度. 为了引入时间间隔和位置信息，需对该式进一步改进[11].

自注意力机制输入 $\boldsymbol{X}=[\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]$，定义输出 $\boldsymbol{Z}=[\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n]$，其中 $\boldsymbol{Z} \in \mathbb{R}^{n\times d}$. 在自注意力机制中加入时间间隔和位置信息的方法：

$$\boldsymbol{z}_i=\sum_{j=1}^{n}\alpha_{ij}(\boldsymbol{x}_j\boldsymbol{W}^V+\boldsymbol{M}_{ij}^V+\boldsymbol{P}_j^V) \tag{4}$$

其中 $\boldsymbol{W}^V \in \mathbb{R}^{d\times d}$ 在 value 中对输入进行线性变换.

权重系数 α_{ij} 可通过 softmax 函数计算：

$$\alpha_{ij}=\exp(e_{ij})\Big/\sum_{k=1}^{n}\exp(e_{ik}) \tag{5}$$

同时相关性分数 e_{ij} 的计算也考虑了交互对象之间的时间间隔和位置信息：

$$e_{ij}=\frac{\boldsymbol{x}_i\boldsymbol{W}^Q(\boldsymbol{x}_j\boldsymbol{W}^K+\boldsymbol{M}_{ij}^K+\boldsymbol{P}_j^K)^{\mathrm{T}}}{\sqrt{d}} \tag{6}$$

其中 $\boldsymbol{W}^Q \in \mathbb{R}^{d\times d}$ 和 $\boldsymbol{W}^K \in \mathbb{R}^{d\times d}$ 分别用于在 query 和 key 中对输入进行线性变换.

《计算机学报》

Li 等人[11]对用户行为序列进行单向建模,序列中每个位置均参与训练,为了防止信息泄露,自注意力计算中每个位置均要屏蔽右侧的交互对象及时间间隔信息.这种方式虽然加快了训练速度,但也有很大缺陷,一方面每个位置只能利用左侧的交互数据,这无疑损失了大量可利用信息,限制了交互对象的表示能力,另一方面由于推荐场景存在复杂的未知因素,用户行为不一定会严格遵循顺序关系,从左右两侧捕捉上下文语义对于用户行为序列建模至关重要[23].

为了对会话序列双向建模,使得每个位置能够融合来自左右两侧的交互对象及时间间隔信息,本文对每个会话进行数据增强处理产生多个子序列,其中每个子序列只有最后的位置参与训练,比如会话$[s_1,s_2,\cdots,s_{z-1},s_z]$划分出的子序列及下一交互对象为$\{([s_1],s_2),([s_1,s_2],s_3),\cdots,([s_1,s_2,\cdots,s_{z-1}],s_z)\}$.因此,对于上述式(6)中的相关性分数计算以及式(4)中的输出序列计算,序列中每个位置均不用屏蔽右侧的交互信息,增强了建模会话上下文的能力,有利于更好地刻画用户兴趣表征.

4.2.2 前馈网络

自注意力机制本质仍是线性建模,为了赋予模型非线性能力并增强不同维度之间的交互,在双向时间间隔感知自注意力后使用两层前馈网络:

$$h_i=ReLU(\boldsymbol{z}_i\boldsymbol{W}_1+\boldsymbol{b}_1)\boldsymbol{W}_2+\boldsymbol{b}_2 \tag{7}$$

其中$\boldsymbol{W}_1\in\mathbb{R}^{d\times d}$、$\boldsymbol{W}_2\in\mathbb{R}^{d\times d}$、$\boldsymbol{b}_1\in\mathbb{R}^d$、$\boldsymbol{b}_2\in\mathbb{R}^d$为可学习参数.前馈网络输出可定义为$\boldsymbol{H}=[\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_n]\in\mathbb{R}^{n\times d}$.

4.2.3 堆叠自注意力模块

双向时间间隔感知自注意力和前馈网络可分别定义为 SA 和 FFN,两者的组合可看作基本的自注意力模块.通常堆叠多个自注意力模块可以捕捉更复杂的交互模式,但网络加深也会造成过拟合、梯度消失等问题,本文利用随机失活(Dropout)[24]、残差连接(Residual Connection)[25]、层归一化(Layer Normalization)[26]等技术解决上述问题.

将每层自注意力模块的$\boldsymbol{Z}$和$\boldsymbol{H}$进一步定义为$\boldsymbol{Z}^l$和$\boldsymbol{H}^l$,$l\in[1,\cdots,ln]$表示层数,在此给出整个自注意力网络的结构定义:

$$\boldsymbol{Z}^l=\mathrm{LN}(\boldsymbol{H}^{l-1}+\mathrm{Dr}(\mathrm{SA}(\boldsymbol{H}^{l-1},\boldsymbol{M}^K,\boldsymbol{M}^V))) \tag{8}$$

$$\boldsymbol{H}^l=\mathrm{LN}(\boldsymbol{Z}^l+\mathrm{Dr}(\mathrm{FFN}(\boldsymbol{Z}^l))) \tag{9}$$

其中 Dr 和 LN 分别表示随机失活和层归一化操作,$\boldsymbol{H}^0=\boldsymbol{X}$.另外,每层自注意力模块参数不同.

4.3 预测层

通过堆叠多层自注意力模块,自注意力网络的输出为$\boldsymbol{H}^{ln}\in\mathbb{R}^{n\times d}$.如图 1 所示,将 SAN-I 和 SAN-T 最后位置的输出$\boldsymbol{H}_n^{ln}$分别定义为$\boldsymbol{o}^i$和$\boldsymbol{o}^t$,作为会话的语义表示.接下来以序列式多任务学习方式预测下一交互对象及时间,下一交互对象预测结果会作为下一时间预测的输入.相比并行式,序列式更加契合用户先产生交互意图后选择合适时间的实际情况,同时也使得主辅任务的联系更加紧密.

4.3.1 下一交互对象预测

首先将序列语义表示$\boldsymbol{o}^i\in\mathbb{R}^d$与交互对象嵌入矩阵$\boldsymbol{E}^I\in\mathbb{R}^{m\times d}$中每个交互对象的嵌入表示做内积,得到当前序列与每个交互对象的相关性分数:

$$\boldsymbol{r}=\boldsymbol{o}^i(\boldsymbol{E}^I)^{\mathrm{T}} \tag{10}$$

其中$\boldsymbol{r}\in\mathbb{R}^m$,$\boldsymbol{r}_j$表示当前序列与交互对象$j$的相关性分数,数值越高意味着相关性越强.

然后选择相关性分数最高的交互对象作为下一交互对象预测结果:

$$\hat{s}_{n+1}=\arg\max(\boldsymbol{r}) \tag{11}$$

4.3.2 下一时间预测

在得到下一交互对象的预测结果后,预测其发生时间.首先通过以下方式将序列语义表示$\boldsymbol{o}^t$与下一交互对象嵌入表示$\boldsymbol{E}^I_{\hat{s}_{n+1}}$进行融合:

$$\boldsymbol{c}^t=[\boldsymbol{o}^t;\boldsymbol{E}^I_{\hat{s}_{n+1}}]\boldsymbol{W}_c+\boldsymbol{b}_c \tag{12}$$

其中$[\boldsymbol{o}^t;\boldsymbol{E}^I_{\hat{s}_{n+1}}]\in\mathbb{R}^{2d}$表示$\boldsymbol{o}^t$与$\boldsymbol{E}^I_{\hat{s}_{n+1}}$的拼接操作,$\boldsymbol{W}_c\in\mathbb{R}^{2d\times d}$、$\boldsymbol{b}_c\in\mathbb{R}^d$为可学习参数.另外,训练阶段式(12)中的下一交互对象如果直接取真实输出,会造成训练阶段和推断阶段的不一致,也就是曝光偏差问题,本文利用计划采样(Scheduled Sampling)[27]解决该问题,训练阶段每次当时间预测需融合下一交互对象时,以概率p选择真实输出s_{n+1},或以概率$1-p$选择由式(11)得到的预测结果$\hat{s}_{n+1}$,概率p随着训练轮次增加而指数衰减.概率p可计算如下:

$$p=\kappa^{epoch_num} \tag{13}$$

其中κ是控制指数衰减速率的超参数,$epoch_num$是当前已经完成的训练轮次.

然后,利用时序点过程预测下一时间,基于融合得到的表示$\boldsymbol{c}^t$,条件强度函数可定义为

$$\lambda(t)=\exp(\boldsymbol{c}^t\cdot\boldsymbol{u}^{t^{\mathrm{T}}}+w^t\cdot g_n+b^t) \tag{14}$$

其中$g_n=t-t_n$为上次交互以来的时间间隔,b^t为偏置项,$\boldsymbol{u}^t\in\mathbb{R}^d$、$w^t\in\mathbb{R}$、$b^t\in\mathbb{R}$均为可学习参数,指数函数用于非线性变换并保证了强度为正.

接着,用式(14)替代式(2)中的条件强度函数,得到条件密度函数的完整表达:

$$f(t)=\exp\left\{\boldsymbol{c}^t\cdot\boldsymbol{u}^{t^{\mathrm{T}}}+w^t\cdot g_n+b^t+\frac{1}{w^t}\exp(\boldsymbol{c}^t\cdot\boldsymbol{u}^{t^{\mathrm{T}}}+b^t)-\frac{1}{w^t}\exp(\boldsymbol{c}^t\cdot\boldsymbol{u}^{t^{\mathrm{T}}}+w^t\cdot g_n+b^t)\right\} \tag{15}$$

最后，下一时间预测结果可计算如下：

$$\hat{t}_{n+1}=t_n+\int_{t_n}^{\infty}t\cdot f(t)\mathrm{d}t \tag{16}$$

式(16)中的积分并没有解析解，可利用一维函数中常用的数值积分技术近似计算.

4.4　模型训练

本文将下一交互对象及时间预测主辅任务以多任务学习框架进行联合训练. 对于下一交互对象预测任务，以二元交叉熵损失为目标函数：

$$loss^i=-[\mathrm{In}(\sigma(\boldsymbol{r}_{s_{n+1}}))+\mathrm{In}(1-\sigma(\boldsymbol{r}_{s'}))] \tag{17}$$

其中对于每个真实输出 s_{n+1}，采样了一个负样本 s'，σ 为 sigmoid 函数，In 为自然对数函数.

对于下一时间预测任务，目标函数定义为 $-\mathrm{In}(f(t_{n+1}))$，具体形式如下：

$$loss^t=-\left\{\boldsymbol{c}^t\cdot\boldsymbol{u}^{t^{\mathrm{T}}}+w^t\cdot g_n+b^t+\frac{1}{w^t}\exp(\boldsymbol{c}^t\cdot\boldsymbol{u}^{t^{\mathrm{T}}}+b^t)-\frac{1}{w^t}\exp(\boldsymbol{c}^t\cdot\boldsymbol{u}^{t^{\mathrm{T}}}+w^t\cdot g_n+b^t)\right\} \tag{18}$$

其中 $g_n=t_{n+1}-t_n$.

最终的目标函数为主辅任务目标函数的和：

$$loss=\sum_{j=1}^{bs}(loss^i+\beta loss^t) \tag{19}$$

其中 β 是调整时间预测辅助任务权重的超参数，bs 是在优化器中使用的批量大小(Batch Size). 最后，可利用端到端的反向传播算法训练整个模型. 算法 1 给出了模型参数学习的主要步骤.

算法 1.　模型参数学习算法.

输入：训练集 $\mathcal{T}^{tr}$ 及其样本数量 $|\mathcal{T}^{tr}|$，交互对象集合 $\mathcal{V}$，批量大小 bs，总训练轮次 $epoch_total$，每个轮次的迭代次数 $iter_total=\lfloor|\mathcal{T}^{tr}|/bs\rfloor$

输出：训练好的模型参数

1. 随机初始化模型参数
2. FOR $i=1$ TO $epoch_total$ DO
3.　FOR $j=1$ TO $iter_total$ DO
4.　从训练集 $\mathcal{T}^{tr}$ 中采样 bs 个样本组成当前训练批量，每个样本包括会话序列 S、时间间隔 $\boldsymbol{R}$、下一交互对象及时间 (s_{n+1},t_{n+1})；
5.　对于上述每个样本的真实输出 s_{n+1}，采样一个负样本 s'；
6.　针对当前训练批量，根据式(19)计算目标函数值；
7.　基于优化器的反向传播算法优化上述目标函数，更新模型参数；
8.　END
9. END

5　实验设置

5.1　数据集①

本文在三个公开真实数据集上评估不同方法.

(1) Tianchi①. 阿里巴巴的公开数据集，基于电商平台真实用户-商品行为，提供了 2014 年 11 月 18 日到 2014 年 12 月 18 日一个月内 20 000 个用户对 4 758 484 个商品的 23 291 027 次交互.

(2) Lastfm②. 该数据集收集了音乐平台 Last.fm 上的用户听歌记录，提供了 2009 年 5 月 5 日之前 992 个用户的完整听歌记录，包括对 961 416 首歌曲的 19 150 868 次交互.

(3) Foursquare③. Foursquare 是一个基于地理位置信息的社交平台，用户可以通过分享自己当前位置进行签到. 该数据集提供了 2012 年 4 月 12 日到 2013 年 2 月 16 日美国纽约和日本东京的用户签到记录[28]. 本文使用东京的用户签到数据，包括 2293 个用户对 61 858 个地点的 573 703 次交互.

三个数据集包含用户的整个交互序列，数据预处理需要将每个用户的交互序列划分为多个会话，同时后续忽略用户 ID 等相关信息，以适应会话式序列推荐的用户匿名场景. 对 Tianchi，过滤掉少于 50 次交互的商品，以会话内相邻交互对象的时间间隔不超过 24 小时来划分序列，并剔除长度小于 3 的会话. 对 Lastfm 和 Foursquare 的处理流程相同. 对 Lastfm，过滤掉少于 20 次交互的歌曲，会话内相邻交互对象的时间间隔不超过 60 分钟. 对 Foursquare，过滤掉少于 5 次交互的地点，会话内相邻交互对象的时间间隔不超过 24 小时.

将会话划分为训练集、验证集和测试集，Tianchi 和 Foursquare 上每个用户前 80%会话用于训练，中间 10%用于验证，后 10%用于测试，由于用户会话数量较多，Lastfm 上该比例取 98%、1%、1%，数据集统计信息见表 2. 另外，训练集每个会话需要按照 4.2.1 节的方法经过数据增强处理划分为多个子序列用于训练. 对时间戳序列的处理方式相同，其中 Tianchi、Lastfm、Foursquare 的时间戳序列分别以小时、分钟、小时为单位.

由表 2 可知，三个数据集的会话平均长度均在 16 以内. 为了充分利用数据，本文将序列长度 n 统

① https://tianchi.aliyun.com/competition/entrance/231522/information
② http://ocelma.net/MusicRecommendationDataset/index.html
③ https://sites.google.com/site/yangdingqi/home/foursquare-dataset#h.p_ID_46

《计算机学报》

一设置为 20，训练和推断阶段均按照 4.1 节的方法对会话序列进行预处理.

表 2 数据集统计信息

数据集	交互对象数量	训练会话数量	测试会话数量	会话平均长度
Tianchi	40901	230512	17572	15.34
Lastfm	156294	927359	9980	13.90
Foursquare	15100	42316	5388	7.75

5.2 评价指标

对于下一交互对象预测任务，利用命中率 Hit 和归一化折损累计增益 $NDCG$ 评估推荐性能[29]，取了四个常用的 Top-k 评价指标 $Hit@5$、$Hit@10$、$NDCG@5$、$NDCG@10$. 其中 $Hit@k$ 衡量真实输出是否排在前 k，未考虑具体位置，$NDCG@k$ 则为更靠前的位置分配更大的分数. 为了减轻计算负担，对于每个真实输出随机采样了 100 个负例，真实输出和负例一同排序. $Hit@k$ 和 $NDCG@k$ 定义如下：

$$Hit@k=\frac{1}{|\mathcal{T}^{te}|}\sum_{i\in\mathcal{T}^{te}}\mathbb{I}(Rank_{r_{s_n+1}}\leqslant k) \tag{20}$$

$$NDCG@k=\frac{1}{|\mathcal{T}^{te}|}\sum_{i\in\mathcal{T}^{te}}\frac{2^{\mathbb{I}(Rank_{r_{s_n+1}}\leq k)}-1}{\log_2(Rank_{r_{s_n+1}}+1)} \tag{21}$$

其中 $\mathcal{T}^{te}$ 和 $|\mathcal{T}^{te}|$ 分别表示测试集及其样本数量，$\mathbb{I}$ 为指示函数，表示真实输出的相关性分数是否位于前 Top-k 位置. 由于每个序列样本只对应一个真实下一交互对象，$NDCG@k$ 可表示为 $DCG@k$ 形式.

对于下一时间预测任务，利用均方根误差 $RMSE$ 评估模型表现. $RMSE$ 可定义为

$$RMSE=\sqrt{\frac{1}{|\mathcal{T}^{te}|}\sum_{i\in\mathcal{T}^{te}}(\hat{t}_{n+1}-t_{n+1})^2} \tag{22}$$

5.3 基线方法

本文将所提出方法与以下方法对比.

第一类是基于深度学习的序列推荐方法，由于基于马尔可夫链的传统方法已被以下多数方法超越，本文忽略了与他的比较：

(1) GRU4Rec[3]. 在会话式序列推荐中首次使用 GRU 建模用户序列.

(2) NARM[4]. 在 GRU 的基础上加入 Attention 用来捕捉用户在会话中的主要目的.

(3) STAMP[5]. 利用 Attention 和 MLP 替代 RNN，建模用户的整体兴趣和当前兴趣.

(4) SR-GNN[6]. 将会话序列建模为图结构数据，使用 GNN 捕捉交互对象之间的复杂转换.

(5) SASRec[15]. 在序列推荐中使用 Self-attention 完全替代 RNN 去建模用户序列，该方法同样适用于会话式序列推荐.

(6) TiSASRec[11]. 对 SASRec 进一步改进，在 Self-attention 中加入了交互对象之间的时间间隔.

第二类选取了以多任务学习框架进行事件/地点和时间预测的方法，他们均只使用单个网络建模用户序列，以并行式预测下一交互对象及时间：

(1) RMTPP[9]. 将 RNN 与时序点过程结合，预测下一事件及时间.

(2) PRPPA-U[8]. 考虑了用户长期偏好和动态兴趣，结合 GRU 和时序点过程预测下一地点及时间. 实验去掉了其中利用用户 ID 表示长期偏好的部分.

5.4 实现细节

本文使用 TensorFlow 框架实现所提出方法和基线方法，优化器选择 Adam[30]，最优超参数的组合根据模型在验证集上的表现确定，并在测试集上验证模型效果. 本文方法对应的最优超参数设置如表 3 所示.

表 3 超参数设置

数据集	批量大小，bs	学习率，lr	自注意力模块层数，ln	语义向量维度，d	L2 正则化，$l2$	随机失活，$dropout$	时间间隔最大值，a	指数衰减速率，κ	辅助任务权重，β
Tianchi	512	0.001	2	128	0.0001	0.2	512	0.92	0.1
Lastfm	512	0.001	2	128	0.0001	0.1	1024	0.90	0.001
Foursquare	512	0.001	2	128	0.0001	0.1	512	0.97	0.1

6 实验结果

6.1 下一交互对象预测主任务方法对比

表 4、表 5、表 6 给出了下一交互对象预测主任务中不同方法在三个数据集上的实验结果对比. 在所有基线方法中，SASRec 和 TiSASRec 比基于 RNN、Attention 和 GNN 的方法效果更突出，证明了 Self-attention 建模序列数据的优势，其中 TiSASRec 是效果最好的基线方法，原因是在 Self-attention 中加入交互对象之间的时间间隔信息后，模型可以捕捉到时间间隔对于表征用户兴趣的重要作用.

表 4　Tianchi 上下一交互对象预测主任务不同方法对比

方法	*Hit*@5	*Hit*@10	*NDCG*@5	*NDCG*@10
GRU4Rec	0.4165	0.5338	0.3104	0.3482
NARM	0.5595	0.6069	0.4829	0.4983
STAMP	0.5837	0.6273	0.5297	0.5348
SR-GNN	0.6133	0.6529	0.5524	0.5701
RMTPP	0.5003	0.5569	0.4325	0.4508
PRPPA-U	0.5652	0.6116	0.5040	0.5195
SASRec	0.7718	0.8519	0.6239	0.6501
TiSASRec	0.7953	0.8724	0.6491	0.6742
Ours	0.8421	0.8938	0.7388	0.7557

表 5　Lastfm 上下一交互对象预测主任务不同方法对比

方法	*Hit*@5	*Hit*@10	*NDCG*@5	*NDCG*@10
GRU4Rec	0.2330	0.3303	0.1614	0.1928
NARM	0.4462	0.5001	0.2914	0.3088
STAMP	0.4586	0.5098	0.3027	0.3164
SR-GNN	0.4621	0.5265	0.3093	0.3278
RMTPP	0.3208	0.3859	0.2704	0.2913
PRPPA-U	0.3319	0.4008	0.2801	0.3023
SASRec	0.4731	0.6058	0.3452	0.3880
TiSASRec	0.4857	0.6183	0.3588	0.4016
Ours	0.5449	0.6591	0.4196	0.4597

表 6　Foursquare 上下一交互对象预测主任务不同方法对比

方法	*Hit*@5	*Hit*@10	*NDCG*@5	*NDCG*@10
GRU4Rec	0.5189	0.6139	0.4090	0.4397
NARM	0.5637	0.6397	0.4871	0.5085
STAMP	0.5965	0.6509	0.5164	0.5296
SR-GNN	0.6432	0.6822	0.5617	0.5703
RMTPP	0.5604	0.6386	0.4637	0.4890
PRPPA-U	0.5860	0.6481	0.5044	0.5215
SASRec	0.7548	0.8295	0.6455	0.6766
TiSASRec	0.7652	0.8318	0.6607	0.6825
Ours	0.8084	0.8634	0.7253	0.7432

本文所提出方法优于所有基线方法，在 *Hit*@5、*Hit*@10、*NDCG*@5、*NDCG*@10 指标上比 TiSASRec 分别取得了 7.91%、4.28%、13.51%、11.82%的提升(三个数据集取平均，下同)，原因一方面是引入了时间预测辅助任务，以序列式多任务学习方式预测下一交互对象及时间，使模型学习到了更多知识，另一方面双向时间间隔感知自注意力机制也使得序列中每个位置可以融合来自左右两侧的交互对象及时间间隔信息. 另外，虽然 RMTPP 和 PRPPA-U 方法也包含时间预测辅助任务，但和本文方法存在较大差距，除了 RNN 建模能力不如 Self-attention 的因素，另外的原因一方面是这些方法均在主辅任务上共享同一个 RNN，参数硬共享无法捕捉任务间的语义差异，另一方面并行式并不完全契合用户交互的实际情况，任务间无法产生更为紧密的联系.

为了进一步验证本文所提出方法相比于 TiSASRec 的优势，利用配对 t 检验(paired t-test)验证两个方法的测试结果是否存在显著性差异. 以一个指标为例，具体流程是：首先利用相同种子对两个方法进行模型参数初始化并完成训练和测试，在不同的种子下重复该过程 10 次，该指标测试结果可组成 10 对样本，也就是样本数量为 10；然后计算所有配对的差值，并假设差值满足正态分布；接着利用所得差值计算 t 值；最后查 t 检验临界值表，根据样本数量和 t 值确定 p 值范围，参照置信水平(设置为 95%)做出两组数据是否存在显著性差异的判断. t 值计算公式如下：

$$t_value = \frac{\bar{d}}{s_d / \sqrt{sam_num}} \tag{23}$$

其中 $\bar{d}$ 和 s_d 分别表示所得差值的平均值和标准差，sam_num 为样本数量.

接下来具体对 *Hit*@5 和 *NDCG*@5 指标进行显著性分析. 表 7 给出了 Foursquare 数据集上 *Hit*@5 和 *NDCG*@5 指标各自对应的 10 对测试结果. 根据式(23)计算 t 值，得到 $t_value_{Hit@5}$、$t_value_{NDCG@5}$ 分别为 51.34、57.27. 经查表可知，两个指标的 p 值均远小于 0.05，因此可以认为该数据集上两个方法的测试结果存在显著性差异.

表 7　Foursquare 上 paired t-test 显著性分析原始测试结果

编号	*Hit*@5		*NDCG*@5	
	Ours	TiSASRec	Ours	TiSASRec
1	0.8075	0.7624	0.7252	0.6591
2	0.8088	0.7688	0.7238	0.6653
3	0.8125	0.7673	0.7258	0.6645
4	0.8082	0.7661	0.7228	0.6644
5	0.8076	0.7670	0.7268	0.6614
6	0.8111	0.7663	0.7267	0.6610
7	0.8119	0.7637	0.7287	0.6603
8	0.8118	0.7669	0.7274	0.6617
9	0.8104	0.7683	0.7262	0.6653
10	0.8106	0.7635	0.7256	0.6600

按照上述方法同样可以对 Tianchi 和 Lastfm 数据集上的 *Hit*@5 和 *NDCG*@5 指标进行显著性分析，此处不再赘述. 表 8 给出了三个数据集的显著性分析结果，最终可得出的结论是，本文所提出方法在三个数据集上均比 TiSASRec 取得显著提升.

表 8　三个数据集上 paired t-test 显著性分析结果

数据集	测试指标	样本数量	t 值	置信水平/%
Tianchi	*Hit*@5	10	79.19	95
	NDCG@5	10	132.12	95
Lastfm	*Hit*@5	10	69.61	95
	NDCG@5	10	91.95	95
Foursquare	*Hit*@5	10	51.34	95
	NDCG@5	10	57.27	95

计算机学报

6.2 下一时间预测辅助任务方法对比

表 9 给出了下一时间预测辅助任务中不同方法在三个数据集上的 RMSE 实验结果对比. 另外,时序点过程建模使用的时间单元过小可能会造成训练困难、计算溢出等问题,本文通过增大时间单元解决[8]. 由于时间戳单位和下一时间间隔数值范围(见 5.1 节)不同,Tianchi、Lastfm、Foursquare 的时间单元分别取的是 4 小时、5 分钟、4 小时.

表 9 三个数据集上不同方法的 RMSE 结果对比

方法	Tianchi	Lastfm	Foursquare
RMTPP	0.8136	0.7740	4.3231
PRPPA-U	0.7791	0.7708	4.1486
Ours	0.7668	0.7662	3.9554

由实验结果可以看出:在主任务取得优异表现的同时,辅助任务效果也超过了基线方法,比 RMTPP 和 PRPPA-U 分别提升了 5.09%和 2.28%.

6.3 不同主辅任务形式影响分析

为了验证本文提出的以序列式多任务学习方式组织主辅任务方法的有效性,本节对比了主辅任务组织的五种形式:

(1)不加入时间预测辅助任务的单任务形式,也就是只预测下一交互对象.

(2)主辅任务共享同一个自注意力网络,以并行式同时预测下一交互对象及时间.

(3)主辅任务共享同一个自注意力网络,以序列式依次预测下一交互对象及时间.

(4)主辅任务使用不同自注意力网络,以并行式同时预测下一交互对象及时间.

(5)主辅任务使用不同自注意力网络,以序列式依次预测下一交互对象及时间.

表 10、表 11、表 12 给出了不同形式在三个数据集上的实验结果. 由实验结果可以看出:其他四种主辅任务形式均比单任务形式有明显提升,证明了时间预测辅助任务对提高推荐效果的重要作用;当共享同一个网络或使用不同网络,序列式比并行式在 *NDCG*@5 指标上分别提升了 1.71%和 3.78%,证明了以序列式多任务学习方式组织主辅任务方法的优势;另外,使用不同网络比共享网络的效果更为突出,原因是相比于单个网络,使用不同网络能够捕捉到不同任务间的语义差异.

表 10 Tianchi 上不同主辅任务形式对比

方法	*Hit*@5	*Hit*@10	*NDCG*@5
单任务	0.8002	0.8766	0.6711
共享网络+并行式	0.8115	0.8810	0.6985
共享网络+序列式	0.8204	0.8825	0.7092
不同网络+并行式	0.8278	0.8868	0.7164
不同网络+序列式	0.8421	0.8938	0.7388

表 11 Lastfm 上不同主辅任务形式对比

方法	*Hit*@5	*Hit*@10	*NDCG*@5
单任务	0.5031	0.6239	0.3640
共享网络+并行式	0.5109	0.6308	0.3826
共享网络+序列式	0.5174	0.6392	0.3909
不同网络+并行式	0.5236	0.6457	0.3981
不同网络+序列式	0.5449	0.6591	0.4196

表 12 Foursquare 上不同主辅任务形式对比

方法	*Hit*@5	*Hit*@10	*NDCG*@5
单任务	0.7743	0.8432	0.6786
共享网络+并行式	0.7801	0.8463	0.6845
共享网络+序列式	0.7888	0.8490	0.6942
不同网络+并行式	0.7969	0.8525	0.7055
不同网络+序列式	0.8084	0.8634	0.7253

6.4 自注意力网络结构影响分析

为了验证本文开发的双向时间间隔感知自注意力方法的有效性,本节在自注意力网络中对时间间隔和双向结构进行擦除实验. w/o G 表示去掉时间间隔,每个位置只融合来自左右两侧的交互对象,不考虑时间信息;w/o S 表示去掉双向结构,每个位置只融合左侧的交互对象及时间间隔信息;w/o G/S 表示同时去掉时间间隔和双向结构,每个位置只融合左侧的交互对象.

表 13、表 14、表 15 给出了不同网络结构在三个数据集上的实验结果. 由实验结果可以看出:当去掉时间间隔或双向结构,模型效果均明显下降,其中

表 13 Tianchi 上不同自注意力网络结构对比

方法	*Hit*@5	*Hit*@10	*NDCG*@5
w/o G/S	0.8168	0.8775	0.6965
w/o G	0.8219	0.8808	0.7136
w/o S	0.8304	0.8841	0.7217
Ours	0.8421	0.8938	0.7388

表 14 Lastfm 上不同自注意力网络结构对比

方法	*Hit*@5	*Hit*@10	*NDCG*@5
w/o G/S	0.5032	0.6344	0.3819
w/o G	0.5177	0.6390	0.3960
w/o S	0.5295	0.6453	0.4002
Ours	0.5449	0.6591	0.4196

表 15 Foursquare 上不同自注意力网络结构对比

方法	*Hit*@5	*Hit*@10	*NDCG*@5
w/o G/S	0.7895	0.8507	0.6959
w/o G	0.7947	0.8571	0.7065
w/o S	0.7980	0.8576	0.7104
Ours	0.8084	0.8634	0.7253

w/o S、w/o G、w/o G/S 在 *NDCG*@5 指标上分别下降了 3.00%、3.88%、6.25%，一方面说明了时间间隔对模型预测的重要影响，另外也验证了融合左右两侧交互对象及时间间隔信息的必要性.

6.5　重要超参数影响分析

本节主要讨论了本文所提出方法涉及到的一些重要超参数对模型效果的影响，包括辅助任务权重 β、语义向量维度 d、时间间隔最大值 a.

(1) 辅助任务权重. 由 6.1 节和 6.3 节的分析可知，引入时间预测辅助任务对提高推荐效果具有关键作用，现在研究不同辅助任务权重对模型效果的影响，该超参数的具体取值集合是 $\{10^{-4},10^{-3},10^{-2},10^{-1},10^{0}\}$. 图 3 给出了三个数据集上 *NDCG*@5 指标在该集合上的变化曲线，对于辅助任务权重的每个取值，其他超参数保持最优设置不变. 从图中可以看出：辅助任务权重对模型效果影响较大，Tianchi、Lastfm、Foursquare 分别在 10^{-1}、10^{-3}、10^{-1} 处取得最佳表现；三个数据集均在 10^{-4} 处表现最差，原因是辅助任务权重过小使得主任务无法充分利用辅助任务提高推荐效果；但过高的权重 10^{0} 并没有取得较好的效果，可能原因是模型过多地关注辅助任务反而会对主任务产生一定的抑制作用.

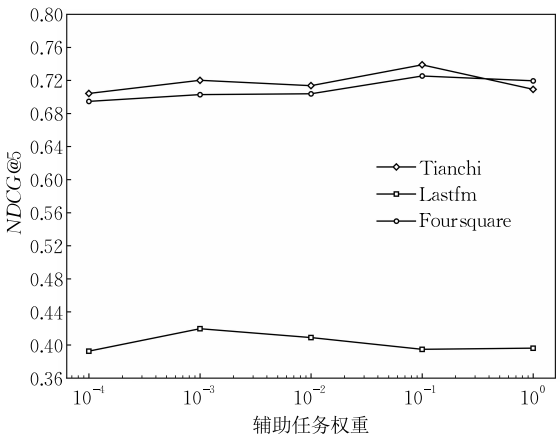

图 3　辅助任务权重对模型效果的影响

(2) 语义向量维度. 接下来分析语义向量维度如何影响模型效果，该超参数的具体取值集合是 {16,32,64,128,256}. 图 4 给出了三个数据集上 *NDCG*@5 指标的变化曲线. 从图中可以看出：三个数据集上 *NDCG*@5 的变化趋势整体一致，随着语义向量维度增加，模型效果也随之提高，最终在 d 取 128 时达到最佳；当继续增加 d 到 256，模型效果均有所下降，原因是更大的维度反而会导致过拟合，降低了模型的泛化能力.

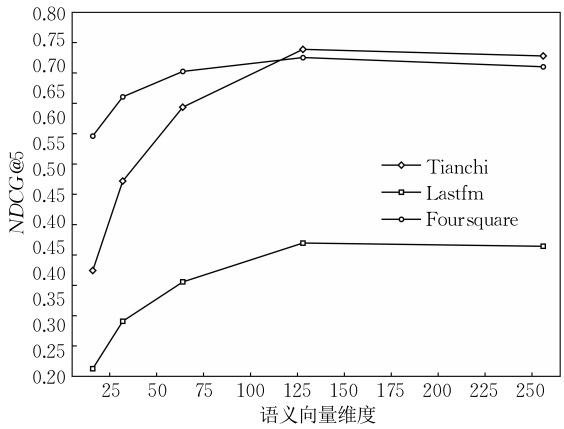

图 4　语义向量维度对模型效果的影响

(3) 时间间隔最大值. 该超参数将会话序列中交互对象之间的时间间隔限定为 0 到 a 之间的自然数，具体取值集合是 {64,128,256,512,1024,2048}. 图 5 给出了三个数据集上 *NDCG*@5 指标的变化曲线. 从图中可以看出：模型效果随着时间间隔最大值的增加而提高，Tianchi、Lastfm、Foursquare 分别在 512、1024、512 处达到最佳；之后当时间间隔最大值进一步增加时，模型效果趋于稳定.

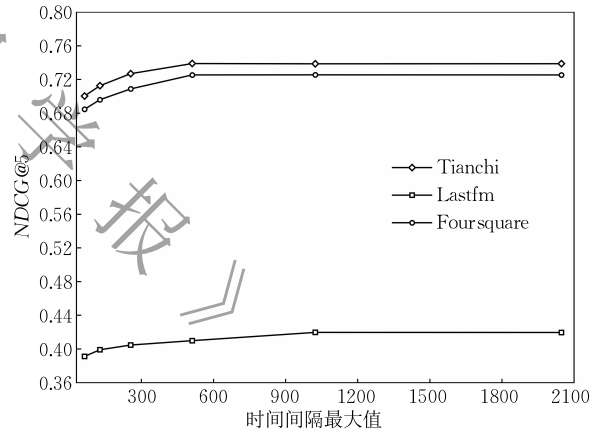

图 5　时间间隔最大值对模型效果的影响

6.6　模型训练时长分析

本文在主辅任务中使用不同网络分别对会话序列建模，模型主体结构相比于 TiSASRec 多使用了一个自注意力网络，因此在提高推荐效果的同时也会增加训练时间开销. 本节对相关训练时长进行了对比分析.

表 16 给出了 TiSASRec 和本文所提出方法在三个数据集上的训练轮次、训练时长、每轮次的平均训练时长(单位为秒). 相关训练时长均在同一运行环境(显卡：NVIDIA GeForce RTX 2080Ti)下测量. 由结果可以看出：两个方法训练轮次一致，本文所提出方法相比于 TiSASRec 在训练时长上增加了

《计算机学报》

35.75%.如何在提升模型推荐效果的同时尽可能减少额外时间开销值得未来做进一步探索.

表 16 三个数据集上不同方法的训练时长对比

数据集	方法	训练轮次	训练时长	平均训练时长
Tianchi	TiSASRec	10	24423	2442.30
	Ours	10	33732	3373.20
Lastfm	TiSASRec	8	65035	8129.38
	Ours	8	84020	10502.50
Foursquare	TiSASRec	50	12852	257.04
	Ours	50	17987	359.74

7 总 结

本文针对会话式序列推荐这一重要问题进行研究,提出了一种新型的序列式多任务深度推荐模型,不仅引入了时间知识促进模型学习,而且使得交互对象和时间的预测顺序也更贴近现实场景.另外,开发了一种经过改进的双向时间间隔感知自注意力方法,赋予模型显式建模用户会话中每个位置与其左右两侧对象交互的能力,并能有效融入其对应时间间隔信息.在三个公开数据集上的实验结果表明了所提出方法的优越性.

参 考 文 献

[1] 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, USA, 2010: 811-820

[2] Gu W R, Dong S B, Zeng Z Z. Increasing recommended effectiveness with Markov chains and purchase intervals. Neural Computing and Applications, 2014, 25(5): 1153-1162

[3] Hidasi B, Karatzoglou A, Baltrunas L, Tikk D. Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939, 2015

[4] Li J, Ren P J, Chen Z M, et al. Neural attentive session-based recommendation//Proceedings of the 26th ACM International Conference on Information and Knowledge Management. Singapore, 2017: 1419-1428

[5] Liu Q, Zeng Y F, Mokhosi R, Zhang H B. STAMP: Short-term attention/memory priority model for session-based recommendation//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. London, UK, 2018: 1831-1839

[6] Wu S, Tang Y Y, Zhu Y Q, et al. Session-based recommendation with graph neural networks//Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu, USA, 2019: 346-353

[7] Vassøy B, Ruocco M, Silva E D S D, Aune E. Time is of the essence: A joint hierarchical RNN and point process model for time and item prediction//Proceedings of the 12th International Conference on Web Search and Data Mining. Melbourne, Australia, 2019: 591-599

[8] Liang W W, Zhang W, Wang X L. Deep sequential multi-task modeling for next check-in time and location prediction//Proceedings of the 24th International Conference on Database Systems for Advanced Applications. Chiang Mai, Thailand, 2019: 353-357

[9] Du N, Dai H J, Trivedi R, et al. Recurrent marked temporal point processes: Embedding event history to vector//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, USA, 2016: 1555-1564

[10] Crawshaw M. Multi-task learning with deep neural networks: A survey. arXiv preprint arXiv:2009.09796, 2020

[11] Li J C, Wang Y J, Mcauley J. Time interval aware self-attention for sequential recommendation//Proceedings of the 13th International Conference on Web Search and Data Mining. Houston, USA, 2020: 322-330

[12] Tan Y K, Xu X X, Liu Y. Improved recurrent neural networks for session-based recommendations//Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. Boston, USA, 2016: 17-22

[13] Qiu R H, Li J J, Huang Z, Yin H Z. Rethinking the item order in session-based recommendation with graph neural networks//Proceedings of the 28th ACM International Conference on Information and Knowledge Management. Beijing, China, 2019: 579-588

[14] Wang W, Zhang W, Liu S K, et al. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction//Proceedings of the Web Conference 2020. Taipei, China, 2020: 3056-3062

[15] Kang W C, Mcauley J. Self-attentive sequential recommendation//Proceedings of the 2018 IEEE International Conference on Data Mining. Singapore, 2018: 197-206

[16] Mei H Y, Eisner J. The neural Hawkes process: A neurally self-modulating multivariate point process//Proceedings of the 31st Conference on Neural Information Processing Systems. Long Beach, USA, 2017: 6757-6767

[17] Yan J C, Xu H T, Li L D. Modeling and applications for temporal point processes//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Anchorage, USA, 2019: 3227-3228

[18] Zhang Y, Yang Q. A survey on multi-task learning. arXiv preprint arXiv:1707.08114, 2017

[19] Hadash G, Shalom O S, Osadchy R. Rank and rate: Multi-task learning for recommender systems//Proceedings of the 12th ACM Conference on Recommender Systems. Vancouver, Canada, 2018: 451-454

[20] Lu Y C, Dong R H, Smyth B. Why I like it: Multi-task learning for recommendation and explanation//Proceedings of the 12th ACM Conference on Recommender Systems. Vancouver, Canada, 2018: 4-12

[21] Tang H Y, Liu J N, Zhao M, Gong X D. Progressive layered extraction (PLE): A novel multi-task learning (MTL) model for personalized recommendations//Proceedings of the 14th

ACM Conference on Recommender Systems. Virtual Event, Brazil, 2020: 269-278

[22] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need//Proceedings of the 31st Conference on Neural Information Processing Systems. Long Beach, USA, 2017: 5998-6008

[23] 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, China, 2019: 1441-1450

[24] Srivastav N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958

[25] He K M, Zhang X Y, Ren S Q, Sun J. Deep residual learning for image recognition//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA, 2016: 770-778

[26] Ba J L, Kiros J R, Hinton G E. Layer normalization. arXiv preprint arXiv:1607.06450, 2016

[27] Bengio S, Vinyals O, Jaitly N, Shazeer N. Scheduled sampling for sequence prediction with recurrent neural networks//Proceedings of the 29th Conference on Neural Information Processing Systems. Montreal, Canada, 2015: 1171-1179

[28] Yang D Q, Zhang D Q, Zheng V W, Yu Z Y. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2015, 45(1): 129-142

[29] He X N, Liao L Z, Zhang H W, et al. Neural collaborative filtering//Proceedings of the 26th International Conference on World Wide Web. Perth, Australia, 2017: 173-182

[30] Kingma D P, Ba J L. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014

CHEN Cong, M. S. candidate. His main research interest is data mining.

ZHANG Wei, Ph. D., associate researcher. His main research interest is data mining.

WANG Jun, Ph. D., professor. His main research interest is computer vision.

《计算机学报》

Background

Session-based sequential recommendation studied in this paper is a specific example of recommendation applications and user behavior modeling. In this field, most existing studies aim to learn accurate user preference representations from anonymous user behavior sessions. Methodologies such as recurrent neural networks, convolutional neural networks, and graph neural networks could observe their successful appliance for the task. Among these studies, only a few have considered to utilize the temporal information associated with behavior sessions, which is promising to benefit the next item recommendation since additional knowledge could be leveraged. Nevertheless, they perform the next item and time prediction in a parallel manner, which is not very consistent with user behavior patterns that users tend to have the interaction intention first and then choose the appropriate time.

This paper studies the session-based sequential recommendation problem with two novel aspects. First, it develops a sequential multi-task learning method which learns to predict the next item first and then feeds it to the next time prediction. Through this way, the proposed model is welcomed for the advantage of coupling the two tasks more tightly. Second, a bidirectional time interval aware Self-attention approach is devised, paving the way to better employ contextual information of each position in user sessions. Comprehensive experiments on three publicly available datasets, i.e., Tianchi, Lastfm and Foursquare, have demonstrated the proposed model could outperform other strong competitors. Taking *NDCG*@5 as an example, it improves the best baseline by 13.51% in average. In addition, the contributions of sequential multi-task learning and the bidirectional time interval aware Self-attention mechanism are well validated.

The authors of this paper have already investigated the task of jointly predicting the next item and time in a parallel mode (please refer to the work[8]). As such, this work is a natural extension. Besides, the authors have a strong background of user generated data modeling. This is a broader field compared with session-based sequential recommendation and provides necessary techniques for it to achieve better performance.

This work is supported by the National Key Research and Development Program of China (No. 2019YFB2102600), the National Natural Science Foundation of China (No. 61702190 and No. 61672236) and the Foundation of Key Laboratory of Artificial Intelligence, Ministry of Education, China.