

文献笔记

9.10-9.16

1.Spatial-Temporal Interval Aware Sequential POI Recommendation

(1) 研究内容: 改进目前的自注意力网络 (SAN), 以降低模型的参数规模或计算负担

(2) 创新点及要点:

- 利用两种轻量级方法, 时间感知位置编码器 (TAPE) 和间隔感知注意块 (IAAB), 通过分别考虑 POI 之间的时空间隔来推动 SAN, 不需要额外的参数也不需要高计算成本
 - TAPE 根据时间戳动态调整序列中的位置, 并通过正弦变换生成位置表示, 可以增强序列表示以反映所有 POI 之间的绝对顺序和相对时间接近度
 - IAAB 将缩放的时空间隔逐点添加到注意力图中, 可以促进注意力机制在时间条件的约束下重视所有 POI 之间的空间关系, 并提供更可解释的推荐
 - 将这两个模块集成到 SAN 中, 并提出了一个 Spatial-Temporal Interval-Aware 顺序 POI 推荐器, 即 STiSAN, 作为端到端部署
- 空间和时间信息是顺序 POI 推荐中的两个关键和互补因素
 - 空间信息可以描述两个 POI 之间的物理接近度, 尤其是当个体移动历史通常表现出空间聚集现象时
 - 时间信息可以反映 POI 之间的相对时间接近度, 对时间因素进行建模将有利于区分这些相同的序列并实现更细粒度的序列表示
- 本文的贡献:
 - 利用两种轻量级方法时间感知位置编码器 (TAPE) 和间隔感知注意块 (IAAB) 来分别考虑 POI 之间的相对时空接近度, 其中既不需要额外的参数也不需要显着的计算负担。
 - * 将时间戳编码为序列表示的 TAPE 可以反映相对时间接近度
 - * IAAB 明确地将时空关系引入注意力图中, 促使人们关注空间相关性
 - 将 TAPE、IAAB 集成到 SAN 中, 并提出时空区间感知顺序 POI 推荐框架 (STiSAN), 作为端到端部署, 以实现更细粒度的偏好建模并提供更可解释的推荐。
- 未来的工作: 将细致地探索自注意力学习并包含在时空关系矩阵中的顺序依赖关系之间的联系和差异

(3) 一些定义:

- 本文中用户每次 check-in 记为 $c^u = \langle u, p, g, t \rangle$, 分别表示用户, POI, 时间以及 POI 位置
- 用户 u 的 POI 访问序列记为 $S^u = c_1^u - > c_2^u - > \dots - > c_{|S^u|}^u$
- Relative Spatial-Temporal Proximity: 第 i 个和第 j 个 POI 的时空相对接近度记为 r_{ij} , 由地理区间 Δd_{ij} 和时间区间 Δt_{ij} 组成
- Top-K Sequential POI Recommendation: 为用户推荐有 K 个排名的 POI 列表: 记为 $TopK^u = Rec(S^u)$

(4) STiSAN:

- 整体架构:

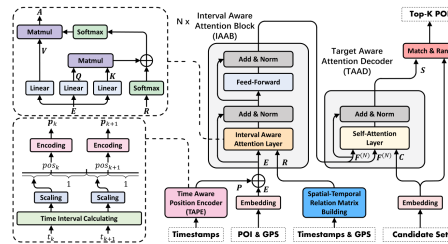


Fig. 3. The framework of STiSAN. The details of two significant components TAPE and IAAB are revealed in dashed boxes.

图 1:

- 遵循经典的编码器-解码器结构, 分别捕获顺序依赖性和改进用户的偏好表示。提出的两个轻量级模块 Time Aware Position Encoder 和 Interval Aware Attention Block 分别取代了 vanilla 位置编码和 self-attention 机制, 其中一种用于增强 POI 序列表示以反映相对时间接近度, 另一种关注局部 POI 之间的空间相关性
- Embedding Module:
 - 该模块将用户 u 的 POI 序列 S^u 作为输入, 并输出序列表示 E , 整体上与其他嵌入层相似
 - 其中较为特殊的一点是: 考虑到不同用户的序列长度不一致, 将较长的序列拆分为几个不重叠的长度为 n 的子序列。对于较短的序列在序列的头部重复添加一个“填充”签入, 直到长度增长到 n
- Time Aware Position Encoder (TAPE):
 - 核心思想: 根据 POI 序列中的时间间隔动态调整位置之间的差异, 并进一步反映相对时间接近度 (即将时间信息同步映射到位置信息上)
 - TAPE 将时间戳 t 依次作为输入, 计算出第 $k+1$ 个 POI 位置:

$$* pos_{k+1} = pos_k + \frac{\Delta t_{k,k+1}}{\Delta t} + 1$$

- * 由于时间戳在不同用户之间的分布不同,通过序列平均时间间隔 $\bar{\Delta t} = \frac{1}{n-1} \sum_{k=1}^{n-1} \Delta t_{k,k+1}$ 对间隔进行归一化
- * We also add an extra 1 to make model distinguish POIs sharing extremely small time intervals
 - 这句有点奇怪, 我的理解是将不同时间访问 POI 所导致的差异带入 POI 的位置信息中, 使得位置之间的差异符合时间相关性
- TAPE 将位置编码到 d 维空间中:
 - * $PE(pos, 2i) = \sin(pos/10000^{2i/d})$
 - * $PE(pos, 2i + 1) = \cos(pos/10000^{2i+1/d})$
- 序列中所有位置的表示矩阵表示为 P, 将其注入到序列表示中 $E = E + P$
- Spatial-Temporal Relation Matrix Building
 - 构建了一个时空关系矩阵 R 以反映时空相对接近度的归纳偏差
 - *

$$R = \begin{bmatrix} r_{11} & \dots & 0 \\ & \ddots & \vdots \\ r_{n1} & & r_{nn} \end{bmatrix}$$
 - * 这里将 R 设置为下三角形使得模型在每步 i 预测第 i + 1 个 POI 时只能关注之前的 i 个 POI, 以防止信息泄漏
 - 第 i 个和第 j 个 POI 之间的时空关系表示为 r_{ij}
 - * 定义 $\hat{r}_{ij} = \Delta t_{ij} + \Delta d_{ij}$ 由相应的时间和地理区间组成
 - * 认为精确间隔在超过某个阈值后无用, 而后分别通过最大时间和地理间隔阈值 k_t 、 k_d 裁剪间隔
 - $\Delta t_{ij} = \min(k_t, |t_i - t_j|)$
 - $\Delta d_{ij} = \min(k_d, \text{Haversine}(g_i, g_j))$
 - * 认为关系应该与它们的距离成反比, 得 $r_{ij} = \hat{r}_{max} - \hat{r}_{ij}$
- Interval Aware Attention Block (IAAB):
 - 推动模型在局部 POI 之间附加重要的空间信息并提供更可解释的推荐
 - IAAB 交替使用间隔感知注意层和前馈网络以及残差连接和层归一化
 - Interval Aware Attention Layer:
 - * 注意力层以序列表示 E 和关系矩阵 R 作为输入
 - * 通过三个不同的矩阵 W_Q 、 W_K 、 W_V 将序列表示 E 转换为 query, key, value 矩阵
 - $Q = EW_Q, K = EW_K, V = EW_V$
 - * 然后, 该层通过逐点加法将注意力图 (即顺序依赖) 与关系矩阵显式组合
 - $A = \text{Softmax}(\frac{QK^T}{\sqrt{d}} + R)V$
 - 注意在添加归一化之前使用 Softmax 缩放 R

- * 通过这种方式, 本文的 IAAB 利用时空关系为注意力图提供了修正, 从而增强了模型考虑局部 POI 之间相对空间接近度的能力
- * 显式组合不是将关系嵌入到高维空间中, 而是提高了模型的可解释性
- Feed-Forward Network:
 - * 采用 2 层逐点前馈网络来编码不同维度之间的相互作用, 并赋予注意力结果非线性
 - * 由两个不同的线性层和激活函数 ReLU 组成
 - $F = \max(0, AW_1 + b_1)W_2 + b_2$
- Residual Connection and Layer Normalization:
 - * 多层神经网络可以帮助模型捕获输入的层次特征。但随着网络的深入, 准确率趋于饱和, 然后迅速下降。该问题是由更多层的累积训练误差引起的, 而非过度拟合
 - * 本文在 STiSAN 中堆叠 $N = 4$ 个间隔感知注意块, 结合残差连接和层归一化, 以稳定和加速训练过程
 - * 假设输入是一个向量 x :
 - $x = x + \text{Layer}(\text{LayerNorm}(x))$
 - 其中 $\text{Layer}(\cdot)$ 可以是注意力层或前馈层, 并且归一化的执行方式为 $\text{LayerNorm}(x) = \alpha \odot \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta$
- Target Aware Attention Decoder (TAAD)
 - 该模块用来改进用户对候选者偏好的表示, 而非直接将第 N 个 IAAB $F^{(N)}$ 的输出与候选 POI 匹配, 即在这个模块引入用户的个人偏好
 - $S = \text{TAAD}(F^{(N)}|C) = \text{Attn}(C, F^{(N)}, F^{(N)})$
 - $\text{Attn}(C, F^{(N)}, F^{(N)}) = \text{Softmax}(\frac{CF^{(N)T}}{\sqrt{d}})F^{(N)}$
- Matching and Ranking
 - 用户在步骤 i 的偏好向量是 S_i , C_j 是 POI j 的表示向量
 - 计算候选 POI j 的偏好分数
 - * $y_{i,j} = f(S_i, C_j)$
 - 在匹配所有候选 POI 并对相应的偏好分数进行排序后, 模型推荐用户 u 将来可能访问的 Top-K POI 列表。
- Model Training
 - 二元交叉熵损失函数被广泛用于优化顺序推荐, 而为了高效训练, 往往只从所有未访问的 POI 中随机抽取一个负样本, 无法充分有效利用大量负样本
 - 对于每个目标 POI o_i , 检索周围最近的 L 个 POI 它作为负样本, 引入加权二元交叉熵损失函数来优化模型
 - * $\text{Loss} = - \sum_{S^u \in S} \sum_{i=1}^n (\log \sigma(y_i, o_i) + \sum_{l=1}^{Lw_l \log(1 - \sigma(y_{i,l}))})$
 - * 其中 $w_l = \frac{\exp(y_{i,l}/T)}{\sum_{l=1}^L \exp(y_{i,l}/T)}$ 是负 POI l 的权重

(5) 实验部分:

- 实验所探究的问题:
 - STiSAN 能否提供与几个最先进的 baseline 相比更出色的性能?
 - 在本文的框架下, 两个提议的组件 TAPE 和 IAAB 的有效性如何?
 - TAPE 和 IAAB 能否有效地应用于 vanilla self-attention 网络?
 - STiSAN 对不同稀疏度级别的敏感度如何?
- 数据集:
 - Gowalla5、Brightkite6、Weeplaces7 和长春城市交通数据集
 - 这里删掉了访问少于 20 个 POI 的用户和交互次数少于 10 次的 POI
 - 目前来看绝大多数的 POI 论文中都有删掉冷数据的操作
- 评价指标:
 - 命中率:

$$* HR@k = \frac{\sum_{|Eval|} |TopK_k \cap trg|}{|Eval|}$$
 - Normalized Discounted Cumulative Gain(NDCG):

$$* NDCG@k = \frac{1}{D} \sum_{i=1}^k \frac{2^{|TopK_k \cap trg|}}{\log_2(i+1)}$$
- Overall Performance : 本文所认为分析的一些问题:
 - 对顺序依赖或高阶交互信息的建模不足
 - 无法处理长序列以及对地理信息的建模不足
 - 垂直卷积操作有助于模型聚合全局序列信息, 避免了 RNN 中的梯度消失问题
 - 基于 self-attention 的方法由于很强的捕获全局序列信息的能力具有更稳定和相对更好的性能
- Ablation Study:
 - 时间感知位置编码器被证明有助于增强序列表示, 即 TAPE 在帮助模型捕获 POI 之间的相对时间接近度方面的有效性, 空间和时间信息在表示序列中都是关键的
 - Interval Aware Attention Block 可以通过修正提供更准确的结果, IAAB 可以更关注空间相关性, 即实现更合理的注意力权重分配
 - TAPE 和 IAAB 在处理较长序列的相对时空接近度方面性能较为优异
 - 通过自注意力机制学习并包含在时空间隔中的顺序依赖关系具有一些相似性并且可以相互完成
 - Target-Aware Attention Decoder 仅在某些情况下有用。可能是 TAAD 忽略了输入序列中最后一个 POI 与候选 POI 之间的空间间隔
- Extensibility and Interpretability of TAPE:
 - 可视化一个用户连续访问的 POI 之间的时间间隔, 其分布是不均匀的并且变化很大, TAPE 会部分地加强或分散对当前位置的注意力, 并对前一个位置进行相应的反操作

- 即两个连续 POI 之间的时间间隔越小，注意力权重就越相似，反之亦然，证明了自注意力机制可以有效地捕获相对时间接近度
- 本文认为 TAPE 是一种用于增强序列表示的有意义、实用且当然是轻量级的方法
- Extensibility and Interpretability of IAAB:
 - IAAB 可以成为 SAN 考虑空间因素的轻量级替代方案
 - IAAB 可以重点关注这些重要的 POI，尤其是那些分布在该序列中更靠前位置的 POI
- Sensitivity w.r.t Sparsity levels:
 - STiSAN 在所有稀疏级别上都更优
 - 随着数据集变得越来越密集，所有方法的性能都会先增加后降低

(6) 一些相关知识:

- Attention Mechanism
 - 核心思想是促使模型更加重视输入中更相关的部分
 - 自注意力的固有缺点是全局权重平均抑制了局部 POI 之间的空间关系
- Positional Representation
 - 本文利用时间感知位置编码器 (TAPE) 将 POI 之间的相对时间接近度编码为序列表示，其中既不需要额外的可学习参数，也不需要大量的计算负担
- Sequential POI Recommendation
 - 本文的框架作为端到端部署，利用轻量级方法（即没有额外的可学习参数和可忽略的计算复杂度）TAPE 和 IAAB 来考虑时空因素
 - 一方面，TAPE 主要侧重于通过编码时间间隔来增强序列表示
 - 另一方面，IAAB 促进注意力机制，重视空间关系并提供可解释的建议

2.Next Point-of-Interest Recommendation with Inferring Multi-step Future Preferences

(1) 研究内容：考虑未来行为对选择 next-POI 决策的影响，即因为用户可能经常会考虑活动计划，用户的下一次 POI 访问也可能会受到他们未来多步骤行为的影响。

(2) 创新点及要点：

- 提出了一种新颖的上下文感知的未来偏好推理推荐器 (CFPRec)，以自我组合的方式帮助推断用户的未来偏好
 - 由于用户每日签到的周期性特征，从学习到的过去偏好中巧妙地推导出多步骤的未来偏好，从而隐含地模仿用户在下次访问前的活动计划
 - 推断出的未来偏好与当前偏好无缝整合，以实现更具表现力的用户偏好学习
- 关于 CFPRec:

- 由过去和当前的偏好编码器以及未来偏好提取器组成
- 采用了 Transformer 层和 LSTM 层分别对用户的过去和当前偏好进行建模
- 设计了三个辅助目标来监督过去偏好的学习。用户日常签到行为的周期性属性, 促使设计了通过自组合方式进行隐式未来偏好推断的提取器, 即多步骤的未来偏好是通过两层注意力聚合从学习到的过去偏好中提取的, 减轻了缺乏明确未来行为的挑战
- CFPRec 为下一个 POI 的预测编码了左边和右边的环境 (即过去/现在和未来的偏好)
- CFPRec 的框架:
 - 过去的偏好编码器, 对用户过去的长期轨迹进行建模, 以捕捉非连续签到行为之间的上下文感知相关性
 - * 双向 Transformer 作为过去的偏好编码器
 - * 在模型训练过程中, Transformer 编码器能够对目标 POI 的左右上下文进行建模, 从而能够学习到更具表现力的 POI 表征
 - * 在捕捉非连续签到之间的上下文关联性和在轨迹中自动聚合最相关的行为方面显示出强大的能力
 - 当前偏好编码器, 通过对最近的顺序签到建模来学习用户当前的偏好
 - * 采用一个 LSTM 单元, 从当前的签到行为中表征用户的时间意识的顺序依赖性
 - 未来偏好提取器, 配备两层注意力聚合, 用于推断多步未来偏好
 - * 用户日常签到行为的周期性特性启发作者推导出一个提取器, 该提取器配备两层注意力聚合, 以自集成的方式推断多步隐式未来偏好, 从而隐式模仿用户下次访问前的活动计划
- 主要贡献:
 - 第一个研究多步骤的未来偏好在下一个 POI 推荐中的影响
 - CFPRec 联合模拟了用户过去、现在和未来的连续行为, 以获得更具表现力的偏好表示。特别是, 设计了一个未来偏好提取器, 以自我组合的方式从学习的过去偏好中推断出隐含的未来偏好

(3) 问题定义:

- 令 U 表示一组用户; L 表示一组 POI (即位置); C 是一组 POI 类别; $T = t_1, t_2, \dots, t_{24}, t_{w_0}, t_{w_1}$ 是一组时隙 (将一天映射为 24 个时隙) 以及工作日 (t_{w_0}) 和周末 (t_{w_1}), 签到记录 $r = (u, l, c, g, t)$,
- S^u 表示用户 u 的轨迹, 本文将一天内按时间顺序排列的一组签到记录记为 S_{cur}^u (即短期行为), 过去的行为 S_{past}^u (即长期行为) 为过去几天 (例如 7 或 14 天) 轨迹
- Research Problem:
 - 给定用户 u 的过去行为 S_{past}^u 和当前行为 S_{cur}^u , 目标是通过推断用户 u 在 t_{k+1}, \dots, t_T 的多步未来偏好, 推荐一组 POI 供用户 u 在下一个时间戳 t_{k+1} 访问

(4) CFPRec:

- 整体模型:

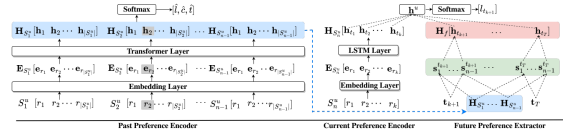


Figure 2: The overall architecture of our proposed CFPRec, which consists of past and current preference encoders as well as the future preference extractor; and the rectangles in grey, pink and green color represent the masked record, inter-sequence attention aggregation and intra-sequence attention aggregation, respectively.

图 2:

- CFPRec 由三个模块组成:

- 过去偏好编码器, 捕捉非连续签到行为之间的上下文感知相关性
- 当前偏好编码器, 学习用户当前的偏好
- 未来偏好提取器, 配备两层注意力聚合, 用于推断多步未来偏好

- Past Preference Encoder

- 本文采用双向 Transformer 作为过去的偏好编码器
 - * 在模型训练过程中, Transformer 编码器能够对目标 POI 的左右上下文进行建模, 从而可以学习 POI 更具表现力的表示
 - * 可以捕获非连续签到之间的上下文感知相关性, 并在轨迹内自动聚合最相关的行为
- 提出了一个扩展 Transformer 编码器来捕获用户过去的偏好, 该偏好编码用户对 POI 的时空感知动态偏好和关于活动的静态偏好
- 将每个签到行为 r_i 表示为: $e_{r_i} = u \oplus l \oplus c \oplus t$
 - * 其中 t 是对时隙和工作日/周末进行编码, 即 $t = t_k \oplus t_{w_0} / t_{w_1}$
- S_i^u 被嵌入层转换为 $E_{S_i^u}$, 然后输入到 Transformer 层
 - * $H_{S_i^u} = [h_1, h_2, \dots, h_{|S_i^u|}] = \text{TransLayer}(E_{S_i^u}) = \text{TransLayer}(E_{S_i^u} W_Q, E_{S_i^u} W_K, E_{S_i^u} W_V, \Delta^{dist})$
 - * $\text{TransLayer}(Q, K, V, \Delta^{dist}) = (F(\frac{QK^T + \Delta^{dist}}{\sqrt{5D}}))V$
- 过去偏好编码器在从长期顺序行为中捕获用户过去偏好方面发挥着重要作用, 本文设计了三个辅助任务来进一步监督过去的偏好学习并帮助捕获更准确的偏好表示
 - * 给定一个序列 S_i^u , 随机屏蔽 20% 的签到记录, 对应的嵌入向量和隐藏状态分别用 e_m 和 h_m 表示, 通过 softmax 层解码隐藏向量 h_m , 对原始位置 l 、类别 c 和时间 t 进行多任务预测:
 - $\hat{l} = F(h_m W_l), \hat{c} = F(h_m W_c), \hat{t} = F(h_m W_t)$
 - * 辅助任务的损失函数为 (负对数似然)
 - $L^{aux} = L_l^{aux} + L_c^{aux} + L_t^{aux} = -\sum_{k \in N_m} \log(\hat{l}_k) + \log(\hat{c}_k) + \log(\hat{t}_k)$

- Current Preference Encoder

- 使用 LSTM 单元来表征用户对当前签到行为的时间感知顺序依赖性

- 给定用户 u 当前的轨迹 S_n^u 嵌入的当前轨迹 $E_{S_n^u}$ 被输入 LSTM 层:
 - * $h_{t_i} = LSTM(e_{r_i}, h_{t_{i-1}})$
- LSTM 层最终输出了针对用户当前偏好的隐藏状态序列 $H_{S_n^u} = [h_{t_1}, h_{t_2}, \dots, h_{t_k}]$ 。
- Future Preference Extractor
 - 核心思想: 用户可能经常下一个动作之前进行活动计划, 因此融合未来的行为将有助于捕获更准确的用户偏好 w.r.t 和下一次 POI 访问
 - 本文推导出一个提取器, 该提取器配备两层注意力聚合, 以自集成的方式推断多步隐式未来偏好, 从而隐式模仿用户下次访问前的活动计划
 - Intra-sequence Attention Aggregation:
 - * 由于用户通常在相同的时间上下文中表现出类似的偏好 (即周期性属性), 因此设计了一种时间感知注意力来识别过去轨迹中与未来时间上下文最相关的行为
 - * 用户过去的轨迹 S_i^u 通过过去的偏好编码器已被编码到隐藏状态矩阵 $H_{S_i^u}$, 其中每个隐藏向量 h_i 表征用户的时空感知动态 POI 偏好和静态活动偏好
 - * 未来时间嵌入 t_f 被用作关注过去轨迹的每个隐藏状态的查询向量, 因此未来时间轨迹嵌入 ($s_i^{t_f}$) 表示为:
 - $s_i^{t_f} = \sum_{i=1}^{|S_i^u|} \alpha_i h_i$
 - $\alpha_i = \frac{\exp(t_f^T h_i)}{\sum_{i=1}^{|S_i^u|} \exp(t_f^T h_{i'})}$
 - Inter-sequence Attention Aggregation:
 - * 序列内注意力层强调未来时间上下文对轨迹内相关行为的影响
 - * 提出了一个序列间注意力层来模拟跨不同轨迹的顺序偏好进化过程
 - * 获得多个关于 t_f 的未来时间轨迹嵌入之后, 使用动态用户嵌入 u 作为查询向量来关注相关的轨迹偏好
 - $h_{t_f} = \sum_{i=1}^{n-1} s_i^{t_f}$
 - $i = \frac{\exp(u^T s_i^{t_f})}{\sum_{i'=1}^{n-1} \exp(u^T s_{i'}^{t_f})}$

(5) 实验部分:

- 实验所研究的问题:
 - CFPRec 是否优于最先进的 baseline?
 - CFPRec 的不同组件如何影响其性能?
 - CFPRec 的关键超参数如何影响其性能?
- 数据集:
 - Singapore (SIN), New York City (NYC) and Phoenix (PHO)
 - 删除了少于 10 次交互的 POI, 并过滤掉了少于 5 个轨迹的非活动用户和少于 3 个签到记录的轨迹
- 评价指标:
 - HitRate@K (HR@K), NDCG@K

- Comparison Results; 基于 RNN 的方法的比较讨论如下
 - 建模顺序行为中利用注意力机制可以提高下一个 POI 推荐的质量
 - 联合建模用户的时空感知活动和位置偏好有利于捕捉用户下次 POI 访问的偏好
 - 对过去和当前的顺序行为进行建模可以获得更准确的 next-POI 推荐，忽略对未来偏好建模会阻碍其用户偏好学习

3.A Survey on Deep Learning Based Point-Of-Interest (POI) Recommendations

(1) Introduction

- POI 推荐技术本质上是利用用户的历史签到和其他多模式信息来推荐适合用户的下一组 POI
- 来自自然语言处理 (NLP) 的一些模型也已用于 POI 推荐中人类移动性的复杂建模
- 本文的主要目的：
 - 展示如何使用不同的技术来处理不同的特征
 - 确定每种技术的优缺点模型
 - 提出潜在差距和未来研究机会的总结
- 本文的主要贡献：
 - 根据应用目标对所有 POI 推荐模型进行分类
 - 概述了该领域中使用的所有数据集的特征，并讨论了它们的优势和局限性
 - 根据不同的深度学习范式对 POI 推荐模型进行分类，并比较它们的优劣
 - 确定了影响 POI 建议的不同因素（即社会影响、顺序效应等），并对所有模型涵盖的每个因素提供表格分析
 - 根据性能指标对所有最新技术进行了比较
 - 确定了现有工作的不足，并为 POI 研究提供了全面的未来建议

(2) Network Architecture Preliminaries

- Feed-Forward Network

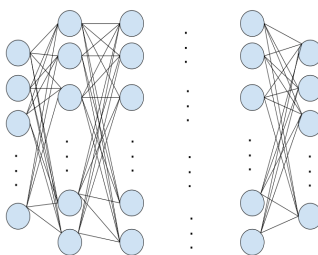


Figure 1: Feed-Forward Network

图 3:

- 神经节点堆叠在层中，其中一层中的每个节点都连接到下一层中的所有节点
- 加权连接结合了一层的特征，并通过非线性函数（例如 ReLU、Sigmoid、tanh 等）将它们传递给后续层，堆叠神经元层显著增加了网络的表现力
- 前馈网络可以捕捉特征内高度复杂的关系，但它们过高的表示能力通常会导致训练数据过拟合，从而导致泛化能力差
- 随着层数的增加，模型的大小会急剧增加，这使得它们更难训练和部署
- 前馈网络没有明确的空间和顺序特征处理能力，这限制了它们在时空模型中的使用
- Convolutional Neural Network(CNN)

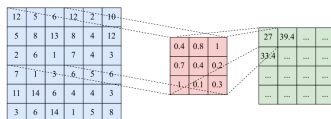


Figure 2: CNN Layer Workflow

图 4:

-
- 适合从给定输入捕获空间特征
- 卷积过滤器和池化层系统地用于分层处理输入。输入的子集通过卷积过滤器逐渐引导，池化层用于缩小转换后的特征
- 上述过程有助于 CNN 获得空间感知，同时保持参数数量显著低于前馈网络
- 不考虑输入中的任何顺序或时间依赖性
- Recurrent Neural Network(RNN)

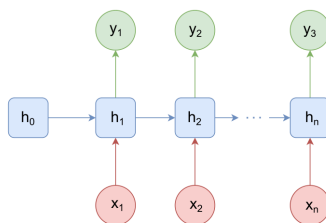


Figure 3: RNN Model

图 5:

-
- 循环神经网络 (RNN) 通过获取输入序列，然后通过使用隐藏状态来学习输入序列的顺序模式
- 图中 x_i 、 y_i 和 h_i 分别代表输入、输出和隐藏状态
- 隐藏状态捕获输入序列的上下文信息即捕获序列模式
- RNN 存在梯度爆炸和梯度消失问题，因此无法捕捉长期偏好

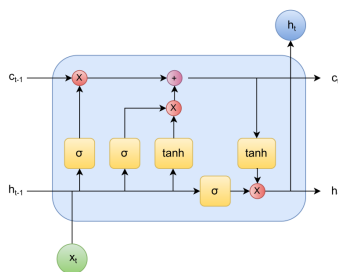


Figure 4: Basic LSTM Cell

图 6:

- Long-Short Term Memory
 -
 - LSTM 使用门机制，能够捕捉长期偏好
 - 图中隐藏状态为 h_t 单元状态 c_t
 - 连续 LSTM 单元之间的信息流通过三个门控制: 输入门; 遗忘门和输出门。训练速度较慢，且需要大量的训练数据
- Gated Recurrent Unit
 - 只使用两个门，即重置和遗忘门
 - 当训练数据较少时，基于 GRU 的模型可以训练得更快，且效果更好
 - RNN 或 LSTM 等序列模型按序列的逻辑顺序处理输入，但这种方案往往会丢失较长序列中的特征，从而导致模型性能不佳
- Attention Mechanism
 - 通过在显著输入区域模仿人类焦点对输入的关键部分给予更高的关注，这反过来又有助于将复杂的输入分解为易于处理的简单部分
 - 注意力机制目前已经引入了自注意力机制，其可以提高性能并允许并行处理输入
 - 关键思想是将输入映射到查询、键和值向量。通过取值向量的加权和来计算输出，其中权重由查询和键值的函数确定
- Generative Adversarial Network

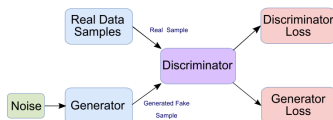


Figure 5: Generative Adversarial Network

图 7:

-
- 生成对抗网络（GAN）是一种特殊形式的生成机器学习框架，其中两个不同的网络以不同的目标相互竞争

- 其中一个称为“生成器”，另一个称为“判别器”。生成器网络生成候选者，而鉴别器网络尝试评估这些候选者
- 生成器试图通过创建新的候选样本来“欺骗”鉴别器，而鉴别器则试图将这些样本与真实数据分布区分开来。

(3) Dataset Description

- 本文所使用的数据集有:Foursquare;Gowalla;Brightkite;Yelp;Weeplaces;Instagram;Twitter;Other datasets
- 数据集中记录了 LBSN 中的 user-POI 和 user-user 关系。User-POI 数据通常包含用户签到信息，包括时间戳、位置和语义特征
 - POI 语义特征包括 POI 的类别和包含在用户 LBSN 帖子中的标签、POI 的创建日期、地理位置（纬度、经度）、签到次数、签到的用户数、签到半径等
 - 用户语义特征包含帖子、朋友、签到等的数量
 - Foursquare、Gowalla、Weeplace 等数据集也包含用户-用户关系作为多对多模式以保持上下文中的社会影响

(4) POI Recommendation Models

- 将模型分为六个主要类别。基于 RNN 的模型;LSTM 模型;GRU 模型;图嵌入模型;GAN 模型和其他模型
- RNN based models
 - 该模型可以将一个 POI 序列映射到另一个 POI 序列（连续 POI 推荐）或仅一个 POI（下一个 POI 推荐）
 - RNN 非常适合分析时间序列数据
- LSTM models
 - Basic LSTM
 - * 大多数模型使用编码器-解码器方案，编码器通常从用户的签到列表和其他属性中收集并使用信息，解码器预测 next-POI 或 POI 序列
 - Bi-LSTM
 - * 双向 LSTM (Bi-LSTM) 通过考虑输入的两个方向使模型从先前的输入中获取顺序信息，并捕获来自输入后面部分的信息
 - Modified LSTM
 - * 基本思想是通过修改基本的 LSTM 来更好地捕捉用户的短期和长期偏好
 - Self-Attention
 - LSTM 有助于捕捉访问 POI 的短期和长期影响
- GRU
 - RNN 和 LSTM 都容易出现冷启动问题。大多数 RNN 模型依赖于最后一个隐藏层，这限制了从隐藏层学习用户信息

- RU 需要学习的参数更少，但它有额外的门，以补偿上述问题，同时解决梯度爆炸和消失的问题
- Graph Embedding
 - 图嵌入可以学习数据空间的低维关键特征
- GAN
 - 两个神经网络模型相互竞争以提供更好的预测
- Other models(deep learning POI recommendation models with hybrid architectures)
- Summary of different paradigms of deep learning models:
 - 序列信息捕获一直是 POI 推荐模型的核心

(5) Influential Factors

- 影响 POI 推荐的一些常见因素，如顺序效应、地理影响、语义效应、社会影响、时间影响等；这些因素源自用户决策的行为
- Sequential Effect
- Geographical Influence
 - 人们倾向于访问离他很近的地方，或者离他已经去过的地方很近的地方
 - 在区域内签到位置的用户有相对较大的概率访问附近的地方
 - 用户倾向于去商店、市场或访问靠近他们居住地的野餐地点
 - 在参观了某个旅游景点后，人们往往会去附近的餐馆或商场
 - 空间邻近度是预测用户下一个位置的一个值得关注的问题
- Semantic Effect
 - 每个 POI 都有一些属性，如果两个 POI 包含相似的属性，则它们在语义上彼此接近
 - 每个人也有自己的偏好，一个人想要访问那些与其偏好相匹配的 POI
 - 从用户的签到列表中，可以捕获用户偏好并尝试预测那些在语义上与之之前签到非常相似的 POI
- Social Influence
 - 一个人的决定很大程度上取决于他/她的社会地位、朋友、邻居、文化等
 - 社会影响会影响一个人访问 POI 的兴趣。当他/她的朋友对该 POI 给予好评时，用户访问该 POI 的概率就会增加
 - 为了解决冷启动问题，新用户的社交圈可能非常有益，因此模型可以通过推荐用户社交圈的相同兴趣点来了解用户的偏好
- Temporal Influence
 - 用户活动受时间影响
 - 用户在一天中的不同时间显示不同的签到偏好，并且倾向于在连续时间而不是非连续时间具有相似的偏好

(6) Shortcomings and Challenges

- Data Sparsity
 - 数据稀疏性是构建位置推荐系统的最关键问题之一
 - 从用户的角度来看,一个人在他/她的一生中只去过很少的地方,而不是访问大量的 POI
 - 探索不同 POI 的成本明显高于探索其他推荐领域的不同选项,这进一步加剧了数据稀疏性问题
- Cold Start
 - 当用户加入 LBSN 网络时,缺乏对该用户的适当表征会导致初始推荐性能不佳
 - 当创建一个新的 POI 进行探索时,由于缺乏与该 POI 相关的历史轨迹,它在推荐方面严重落后于现有的 POI
- Scarcity of benchmark dataset
 - 已有数据集的庞大规模使得其无法同时使用,大多数先前的工作仅使用这些数据集的一个子集来评估所提出的模型
 - 最近研究表明,除了签到之外,来自文本描述、用户评论等其他模式的数据可以提高推荐性能。而现有的主流数据集不包含这些数据
- Lack of online learning
 - 日常用户正在生成大量新的签到数据,其中包含有关用户偏好变化的重要信息。因此可以使用新数据更新模型的在线学习策略具有一定的研究前景
- Privacy Preserved POI Recommendation
 - 将联邦学习运用于 POI 推荐
- Recommendation for Social Groups
 - 对社交群体的推荐与个性化 POI 推荐不同,因为每个群体成员可能对选择 POI 有不同的偏好。在为的一组用户推荐 POI 时,社交方面变得尤为重要

4. An Experimental Evaluation of Point-of-interest Recommendation in Location-based Social Networks

(1) INTRODUCTION

- POI 推荐所面临与其他推荐系统所不同的挑战:
 - 丰富的上下文
 - * 用户的移动偏好受地理距离的影响,通常在少数活动区域内访问 POI
 - * 用户可能每天都访问相同的 POI
 - * 用户的偏好与时间有关
 - * 用户访问偏好可能会受到影响通过他们的社会关系
 - 数据稀缺问题

- * POI 推荐的数据稀缺性问题比其他推荐问题严重得多
- * 用户访问的 POI 数量通常只是所有 POI 的一小部分
- 设计了一个评估程序来评估 12 个具有代表性的 POI 推荐模型, 旨在从多个方面获得 POI 推荐模型的总体图景
- 即在不同来源、不同稀疏度的数据集以及具有不同历史数据大小的用户上试验这些模型, 为这些 POI 推荐方法的相对优点以及这些模型的适用场景提供了新的见解
- 评估了 POI 推荐中用户偏好建模的不同推荐技术, 以及上下文信息的建模方法以提供有关哪种方法对每个组件表现更好的见解

(2) POI RECOMMENDATION

- 展示了 LBSN Gowalla、Foursquare 和 Yelp 数据集的一些统计分析结果
 - 空间影响: 将用户的连续签到视为 POI 之间的转换, 并计算用户的转换距离分布, 通过分析三个数据集的过渡距离的累积分布函数 (CDF) 可以看出用户倾向于访问附近的 POI
 - 社会影响: 计算朋友之间和随机抽样的非朋友之间的常见签到 POI 的数量, 可以看出大多数朋友在他们的签到 POI 上有小的重叠, 但重叠明显大于非朋友
 - 时间影响: 一方面, 两个用户可能在时间上表现不同。另一方面, 不同的 POI 有不同的开放时间和高峰时间。因此随着时间的推移, 它们的签到模式也不同
- 感觉此处整体来看没有问题, 不过感觉可能有一些笼统, 是否可以在此基础上对三大类更进一步细分小类
- 值得注意的一点是, 本文在分析模型时都大体上根据这三类进行分析

(3) MODELS FOR EVALUATION

- 根据模型的推荐技术对其进行分组
- Matrix Factorization Models
 - 矩阵分解 (MF) 将签到矩阵 $C \in R^{M \times N}$ 分解为用户矩阵 $U \in R^{M \times K}$ 和 POI 矩阵 $L \in R^{N \times K}$, 其中 M、N 和 K 是用户数, POI 和潜在因素, 每个用户 i 和 POI j 的潜在特征由 u_i 和 l_j 表示
 - 用户 i 对 POI j 的推荐分数因此建模为内积
 - * $\hat{C}_{ij} = u_i l_j^T$
 - 目标函数表示为:
 - * $\min_{U, L} \|C - UL^T\|_F^2 + \lambda_1 \|U\|_F^2 + \lambda_2 \|L\|_F^2$
 - 模型主要有: LRT; IRenMF; GeoMF; RankGeoFM; ASMF
- Poisson Factor Models
 - 泊松因子模型 (PFM) 是一种概率模型, 将 user-POI 签入矩阵 C 分解为 C Poisson(UL^T)
 - 模型主要有: MGMPFM; GeoPFM

- Link-based Models
 - 构建一个图来对 LBSN 用户及其关系进行建模。在图中，用户偏好和社会影响都由不同类型的边建模
 - 具有相似签到行为的用户被链接以建模“相似关系”，并添加表示“友谊关系”的边以连接图中的朋友
 - 模型有 LFBCA
- Hybrid Models
 - 混合模型结合了两种或多种推荐方法的输出，每种方法都对用户偏好或一种上下文信息进行建模
 - 模型主要有:USG;iGSLR;LORE;GeoSoCa

(4) EVALUATION SETTING

- Datasets
 - 在实验中使用三个数据集: Gowalla dataset;Foursquare dataset;Yelp dataset
- Evaluation Metrics
 - 在实验中使用四种评价指标: precision (Pre@K), recall (Rec@K), normalized discounted cumulative gain (nDCG@K) and mean average precision (MAP@K)
- Performance Evaluation Procedure
 - Evaluation on Different Types of Data
 - * Different datasets
 - 在 Gowalla、Foursquare 和 Yelp 数据集上评估所有模型
 - 令 K 从 5、10、20 变为 50
 - 实验显示对于每个模型，Gowalla 和 Foursquare 数据的准确性相似，而所有模型在 Yelp 上的表现都较差
 - 而后是各个模型的表现及对其性能的分析:
 - 对于混合模型，最好利用用户偏好建模之上的其他信息来进行 POI 推荐
 - 将用户的签到建模为隐式反馈在 POI 推荐中更合适
 - 对用户倾向的假设会影响模型在数据集上的表现
 - 学习地理影响可能将改进 MF 模型
 - 对于基于链接的模型，用户与其空间行为的关系也应该被视为“链接”
 - 联合学习将是比单独建模更好的利用上下文信息的方法
 - * Data density
 - 通过随机消除签入矩阵的非零条目（以使其更稀疏）或随机将数据从调整集移动到训练集（使其更密集）从而研究训练数据密度的影响
 - 从具体实验来看数据密度与模型效果成正比，个人认为可以具体分析比较各个模型效果的增减幅度随数据密度的变化

- Evaluation for Different Types of Users
 - * 研究不同类型的用户属性对推荐模型的影响
 - * Number of check-in POIs of users
 - 签到 POI 的数量预计会影响推荐模型的准确性
 - 根据训练数据中签到 POI 的数量将用户分组
 - 使用所有用户训练模型，并分别对不同的用户组进行评估
 - 实验考虑了模型对于活跃用户和冷启动用户性能
 - 实验考虑签到 POI 数量的增加，不同模型性能的增幅
 - * Activity range of users
 - 由于地理因素在 POI 推荐中很重要，设计实验来研究用户活动范围对推荐模型的影响
 - 根据用户签到的相邻 POI 之间的平均距离将用户分组
 - 实验分析: 更大范围的用户活动会对地理增强型 POI 推荐产生副作用。地理模型可能不适用于具有广泛活动范围的用户
 - * Tourist Users
 - 评估特殊类型用户（即游客）的模型
- Evaluation for Different Modeling Methods
 - * 对于混合 POI 推荐模型来说其不同之处在于如何对用户偏好和每种类型的上下文信息进行建模，特别是地理和社会影响
 - * 本节评估这些方法的单个组件以评估其性能
 - Comparing geographical modeling methods
 - Comparing social modeling methods
 - Comparing user preference modeling methods
- Scalability Evaluation
 - * 评估模型的训练和查询（即推荐）可扩展性
 - * Time complexity analysis
 - 分析了不同模型的训练和查询时间复杂度
 - * Training scalability
 - 这里的训练可扩展性是由模型的训练时间是否受训练集的大小所影响或训练时间随着训练数据的大小所增长的速度而决定的
 - * Querying scalability
 - 查询时间表示计算用户 u 在所有 POI 上的推荐分数的平均时间

(5) Summary of New Insights

- 基于隐式反馈模型和考虑地理信息是两个提升模型效果的属性
- 提出了对密集数据和稀疏数据有较好效果的模型, 提出了对稀疏性变化不敏感的模型
- 顺序建模在面对用户拥有更多 POI 的情况时，增幅最大

- 对于较大的用户活动范围，所有地理模型的准确性都会降低
- 幂律分布是地理建模的有效解决方案
- 建模个性化二维签到分布更适合活跃用户，且个性化的二维分布需要更多的数据来精准捕捉用户的行为

(6) POI Recommendation

- Collaborative Filtering
- Link-based Methods
- Factorization Models
- Probabilistic Models

(7) Other POI Recommendation Problems

- Next POI Recommendation
 - 给定用户及其当前位置，next-POI 推荐旨在推荐用户在下一个时间间隔（例如，在接下来的 6 小时内）可能访问的新 POI
 - 大多数下一个 POI 推荐模型使用连续签到之间的顺序信息来推荐。此外，为位置预测开发的模型也可以应用于下一个 POI 推荐
- Time-aware POI Recommendation
 - 考虑到用户偏好随时间而变化，给定用户一个时间，时间感知 POI 推荐会向用户返回当时最有可能访问的新 POI
- In-town/Out-of-town POI Recommendation
 - 给定用户的家乡和当前位置或城市，in-town/out-of-town POI 推荐向用户返回新的 POI
 - 当用户在城内或外地时使用不同的推荐策略