

周报

论文修改

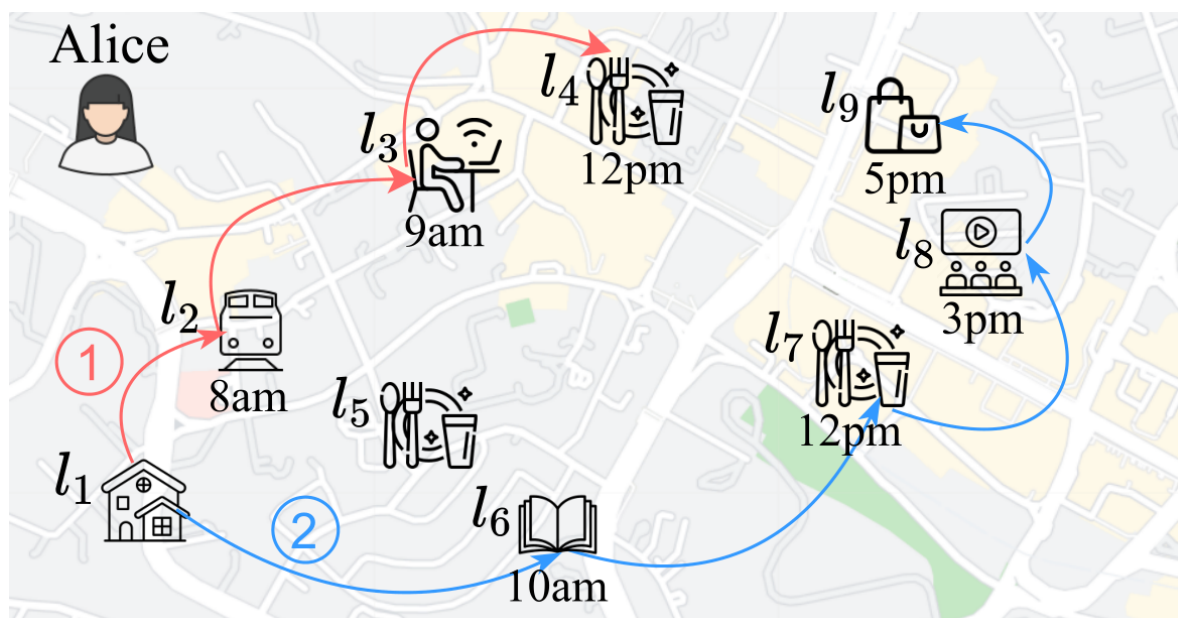
完成了对Introduction部分逻辑上问题的修改

完成了方法部分的修改，添加了地理相似度计算模块

论文阅读

Overview

论文认为，用户的下一个行为可能会受到过去和当前的行为外，还会受到多步的未来行为的影响，因为他们可能心里有活动计划。如下图所示，Alice 在轨迹 ① 中前往了距离公司 l_3 较近的餐厅 l_4 ，这说明她的行为受到最近的行为影响；另一方面在轨迹 ② 中，Alice 在离开图书馆 l_6 之后，并没有前往较近的餐厅 l_5 ，而是前往了较远的餐厅 l_7 ，之后去了电影院 l_8 和购物 l_9 ，这说明她在行动之前已经有了计划。



论文提出Context-aware Future Preference inference Recommender (CFPRec)，联合对用户过去、当前和未来的顺序行为进行建模，以获得更具表现性的偏好表示。此外，论文设计了一个未来的偏好提取器，以一种自融合的方式从过去的行为偏好中推断出隐含的未来偏好。

就是进行多步预测，再对多步预测的结果进行处理。

Problem Formulation

分别给定用户集合 $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$, POI 集合 $\mathcal{L} = \{l_1, l_2, \dots, l_{|\mathcal{L}|}\}$, POI 类别集合 $\mathcal{C} = \{c_1, c_2, \dots, c_{|\mathcal{C}|}\}$ 以及时间戳集合 $\mathcal{T} = \{t_1, t_2, \dots, t_{24}, t_{w_0}, t_{w_1}\}$ (t_{w_0}, t_{w_1} 分别表示工作日和节假日)。

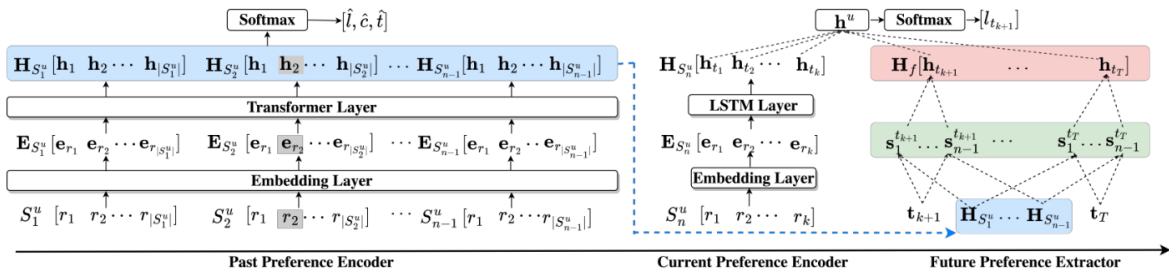
(check-in) check-in 序列为 $r = (u, l, c, g, t)$, 表示用户 u 在 t 时刻访问位置 l , l 的类别为 c , 地理位置为 g 。

(user trajectory) 用户轨迹是由特定用户的一系列时间顺序的签到记录来定义的, 即 $S^u = \{S_1^u, S_2^u, \dots, S_n^u\}$, 其中 $S_i^u = \{r_1, r_2, \dots, r_{|S_i^u|}\}$ 。用户 u 最新的轨迹可以表示为 $S_n^u = \{r_1, r_2, \dots, r_k\}$, 即当前行为 S_{cur}^u ; 用户历史行为 S_{past}^u 即 S_{cur}^u 之前的几天中的一系列行为 $\{S_1^u, \dots, S_{n-1}^u\}$ 。

(next POI recommendation) 给定用户 u 的历史行为 S_{past}^u 以及当前行为 S_{cur}^u , 通过对用户行为进行多步预测 (t_{k+1}, \dots, t_T), 预测用户 u 在下一时刻 t_{k+1} 的访问位置。

CFPRec

模型整体结构如下图所示:



为了从用户历史行为中学习用户偏好, 论文采用了双向 Transformer 结构, 主要基于两个原因:

1. 在训练过程中, Transformer 能够同时学习目标 POI 左右两侧的上下文信息, 对 POI 的表征能力强;

2. 其次，它在捕获非连续签到和聚合轨迹中最相关的行为的上下文相关性方面具有强大的能力。

Past Preference Encoder

具体来说，给定用户 u 的历史轨迹 $S_i^u = \{r_1, r_2, \dots, r_{|S_i^u|}\} \in S_{past}^u$ ，将每个 check-in 序列 $r = (u, l, c, g, t)$ 进行 Embedding：

$$\mathbf{e}_{r_i} = \mathbf{u} \oplus \mathbf{l} \oplus \mathbf{c} \oplus \mathbf{t}, \mathbf{e}_{r_i} \in \mathbb{R}^{5D}$$

其中 \oplus 表示连接操作； $\mathbf{u}, \mathbf{l}, \mathbf{c}$ 分别表示 user, location, category embedding； $\mathbf{t} \in \mathbb{R}^{2D}$ 为时间戳 embedding， $\mathbf{t} = \mathbf{t}_k \oplus \mathbf{t}_{w0}/\mathbf{t}_{w1}$ 。

经过 Embedding Layer 之后得到 $\mathbf{E}_{S_i^u} = [\mathbf{e}_{r_1}, \mathbf{e}_{r_2}, \dots, \mathbf{e}_{r_{|S_i^u|}}]$ ，并输入 Transformer Layer：

$$\begin{aligned} \mathbf{H}_{S_i^u} &= [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{|S_i^u|}] = \text{TransLayer}(\mathbf{E}_{S_i^u}) \\ &= \text{TransLayer}(\mathbf{E}_{S_i^u} \mathbf{W}_Q, \mathbf{E}_{S_i^u} \mathbf{W}_K, \mathbf{E}_{S_i^u} \mathbf{W}_V, \Delta^{dist}) \end{aligned}$$

$$\text{TransLayer}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \Delta^{dist}) = \left(F\left(\frac{\mathbf{Q}\mathbf{K}^T + \Delta^{dist}}{\sqrt{5D}}\right) \right) \mathbf{V}$$

其中 $F(\cdot)$ 为 softmax; $\mathbf{h}_i \in \mathbb{R}^{5D}$; $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{5D \times 5D}$ 为可学习矩阵; $\Delta^{dist} \in \mathbb{R}^{|S_i^u| \times |S_i^u|}$ 为空间关系矩阵, 位置 l_i, l_j 之间的空间关系计算如下:

$$\Delta_{i,j}^{dist} = (1 + \text{dist}(l_i, l_j))^{-1}$$

Past Preference Encoder 在从长期的序列行为中, 捕获用户的过去的偏好方面发挥了重要的作用。同时, 论文设计了三个辅助任务来进一步监督过去的偏好学习, 并帮助获取更准确的偏好表征。

对于序列 S_i^U , 随机 mask 20% 的 check-in 记录, 获得 $r_m = (u, l_m, c_m, g_m, t_m)$, 对应的 embedding 和隐藏状态分别用 \mathbf{e}_m 和 \mathbf{h}_m 表示。接着, 论文通过 softmax 解码隐藏向量 \mathbf{h}_m , 对原始位置 l 、类别 c 和时间 t 进行多任务预测:

$$\hat{l} = F(\mathbf{h}_m \mathbf{W}_l), \hat{c} = F(\mathbf{h}_m \mathbf{W}_c), \hat{t} = F(\mathbf{h}_m \mathbf{W}_t)$$

其中 $\mathbf{W}_l \in \mathbb{R}^{5D \times |\mathcal{L}|}$, $\mathbf{W}_c \in \mathbb{R}^{5D \times |\mathcal{C}|}$, $\mathbf{W}_t \in \mathbb{R}^{5D \times |\mathcal{T}|}$ 分别为三个任务的可学习矩阵。辅助任务的损失函数计算如下:

$$\begin{aligned} L^{aux} &= L_l^{aux} + L_c^{aux} + L_t^{aux} \\ &= - \sum_{k \in N_m} \log(\hat{l}_k) + \log(\hat{c}_k) + \log(\hat{t}_k) \end{aligned}$$

其中 N_m 为经过 mask 的记录。

Current Preference Encoder

论文通过使用一个 LSTM 单元来获取用户对当前签到行为的时间感知顺序依赖关系。与之前类似地，首先将用户行为轨迹输入到 Embedding Layer，获得当前轨迹的特征表示 $\mathbf{E}_{S_n^u} = [\mathbf{e}_{r_1}, \mathbf{e}_{r_2}, \dots, \mathbf{e}_{r_k}]$ 输入 LSTM Layer:

$$\mathbf{h}_{t_i} = \text{LSTM}(\mathbf{e}_{r_i}, \mathbf{h}_{t_{i-1}}), i \in \{1, 2, \dots, k\}$$

其中 $\mathbf{h}_{t_i}, \mathbf{h}_{t_{i-1}}$ 分别为 t_i, t_{i-1} 时刻的隐藏层状态。最后得到 LSTM 层的输出 $\mathbf{H}_{S_n^u} = [\mathbf{h}_{t_1}, \mathbf{h}_{t_2}, \dots, \mathbf{h}_{t_k}]$

Future Preference Extractor

就像之前提到的，论文期望通过融合未来的行为捕获更准确的用户偏好。诚然，在实际的 POI 推荐中，用户的未来行为是不可知的，仍有一些基于双向的方法在模型训练中编码了过去和未来的行为，即对目标 POI 的左右上下文进行建模，但在预测任务中，由于无法获得未来行为，限制了捕获用户偏好的能力。

这篇论文根据用户日常行为的周期性提出了一个提取器，包含两层的 Attention，通过自融合的方式推断隐含的未来偏好行为。

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（即周期属性），因此论文设计了一个时间感知的注意力器，根据时间背景来识别在过去的轨迹中与未来最相关的行为。

具体来说，在之前的步骤中，已经将用户历史轨迹 $S_i^u \in S_{past}^u$ 编码为 $\mathbf{H}_{S_i^u} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{|S_i^u|}]$ ，其中，每个隐藏向量 \mathbf{h}_i 表征了用户的时空感知动态 POI 偏好和静态活动偏好。

接着，使用未来时间 embedding \mathbf{t}_f 作为查询向量（query）关注过去轨迹的每个隐藏状态，得到未来时间轨迹 embedding $\mathbf{s}_i^{t_f}$ ：

$$\mathbf{s}_i^{t_f} = \sum_{i=1}^{|S_i^u|} \alpha_i \mathbf{h}_i, t_f \in \{t_{k+1}, t_{k+2}, \dots, t_T\}$$

$$\alpha_i = \frac{\exp(\mathbf{t}_f^T \mathbf{h}_i)}{\sum_{i'=1}^{|S_i^u|} \exp(\mathbf{t}_f^T \mathbf{h}_{i'})}$$

因此， $\mathbf{s}_i^{t_f}$ 能够从过去的轨迹 S_i^u 中提取与未来时间戳 t_f 最相关的行为来表示潜在的未來行为。

Inter-sequence Attention Aggregation

序列内注意层（Intra-sequence）强调了未来时间上下文背景对轨迹内相关行为的影响。此外，用户的日常顺序偏好也在不断变化，因此，论文提出了一个序列间注意层（Inter-sequence）来模拟不同轨迹间的序列偏好演变过程。

使用动态的 \mathbf{u} 作为查询向量（query）关注每一条轨迹偏好：

$$\mathbf{h}_{t_f} = \sum_{i=1}^{n-1} \beta_i \mathbf{s}_i^{t_f}, t_f \in \{t_{k+1}, t_{k+2}, \dots, t_T\}$$

$$\beta_i = \frac{\exp(\mathbf{u}^T \mathbf{s}_i^{t_f})}{\sum_{i'=1}^{n-1} \exp(\mathbf{u}^T \mathbf{s}_{i'}^{t_f})}$$

其中 \mathbf{h}_{t_f} 为 t_f 时刻的未来偏好。

现在，未来偏好提取器通过自融合实现了推断多步的未来偏好，即 $\mathbf{H}_f = [\mathbf{h}_{t_{k+1}}, \mathbf{h}_{t_{k+2}}, \dots, \mathbf{h}_{t_T}]$ ，这也继承了用户过去的偏好。

Model Training and Complexity Analysis

将用户当前偏好和未来偏好的隐藏层信息整合得到 $\mathbf{H}_{S_n^u, f} = [\mathbf{H}_{S_n^u}, \mathbf{H}_f]$, 这使得 CFPRec 能够在训练和预测过程中对目标 POI 的左右上下文进行建模。用户的最终偏好计算如下:

$$\mathbf{h}^u = \sum_{i=1}^T \omega_i \mathbf{h}_{t_i}, \mathbf{h}_{t_i} \in \mathbf{H}_{S_n^u, f}$$
$$\omega_i = \frac{\exp(\mathbf{u}^T \mathbf{h}_{t_i})}{\sum_{i'=1}^T \exp(\mathbf{u}^T \mathbf{h}_{t_{i'}})}$$

得到的用户偏好 \mathbf{h}^u , 使用 softmax 得到对不同 POI 的访问概率:

$$\hat{y} = F(\mathbf{h}^u \mathbf{W}_o)$$

其中 $\hat{y} \in \mathbb{R}^{|\mathcal{L}|}$ 表示预测概率; $\mathbf{W}_o \in \mathbb{R}^{5D \times |\mathcal{L}|}$ 为可学习矩阵。损失函数计算如下:

$$L^{poi} = - \sum_{i \in \mathcal{N}} \log(\hat{y}_i)$$

其中 \mathcal{N} 为训练集样本。最后与前面的辅助任务结合得到最终的损失函数:

$$L = L^{poi} + \eta L^{aux}$$

算法流程图如下所示:

Algorithm 1: Training of the CFPRec

Input: $\mathcal{U}, \{S^{u_1}, S^{u_2}, \dots, S^{u_{|\mathcal{U}|}}\}, max_iter, \eta, D, T, \gamma$ **Output:** Trained model $\{\Theta\}$

```
1 Initialize model parameters  $\Theta, \mathcal{N} = \emptyset$ ;
  // Training instance construction
2 foreach  $u \in \mathcal{U}$  do
3   foreach  $S_i^u \in S^u$  do
4      $S_{past}^u = \{S_1^u, S_2^u, \dots, S_{n-1}^u\}, S_{cur}^u = S_n^u$ 
5      $\{S_{past}^u, S_{cur}^u\} \mapsto \mathcal{N}$ 
  // Update parameters  $\Theta$ 
6 foreach  $iter \in \{1, 2, \dots, max\_iter\}$  do
7   Select instance  $\{S_{past}^u, S_{cur}^u\}$  from  $\mathcal{N}$ 
8   Update  $\Theta$  by minimizing the objective in Eq.(12)
```

Experiments

Datasets

论文的数据集为 Foursquare, 从中选择了三个城市: Singapore (SIN), New York City (NYC), Phoenix (PHO)

	# User	# POI	# Check-in	Density
SIN	8,648	33,712	355,337	0.12%
NYC	16,387	56,252	511,431	0.06%
PHO	2,946	7,247	47,980	0.22%

Baselines

MostPop BPRMF ST-RNN ATST-LSTM MCARNN PLSPL iMTL CTLE

Results

	SIN				NYC				PHO			
	HR@5	HR@10	NDCG@5	NDCG@10	HR@5	HR@10	NDCG@5	NDCG@10	HR@5	HR@10	NDCG@5	NDCG@10
MostPop	0.0125	0.0293	0.0106	0.0151	0.0139	0.0235	0.0101	0.0145	0.0160	0.0223	0.0114	0.0131
BPRMF	0.0352	0.0525	0.0222	0.0380	0.0381	0.0468	0.0239	0.0339	0.0487	0.0585	0.0256	0.0304
ST-RNN	0.0959	0.1370	0.0655	0.0794	0.1347	0.1826	0.0593	0.1303	0.1240	0.2028	0.0802	0.1229
ATST-LSTM	0.1296	0.1933	0.1027	0.1476	0.1667	0.2031	0.0912	0.1638	0.1579	0.2377	0.1033	0.1385
MCARNN	0.1608	0.1862	0.1169	<u>0.1591</u>	0.1835	0.2397	0.1036	<u>0.1870</u>	0.1905	0.2726	0.1264	0.1617
iMTL	0.1505	0.1801	0.1051	0.1423	0.1798	0.2422	0.0989	0.1861	0.1830	0.2747	0.1301	0.1632
PLSPL	0.1447	0.1719	0.1126	0.1384	0.1741	0.2413	0.0961	0.1825	0.1775	0.2569	0.1285	0.1538
CTLE	<u>0.2041</u>	<u>0.2784</u>	<u>0.1315</u>	0.1556	<u>0.2421</u>	<u>0.3205</u>	<u>0.1513</u>	0.1841	<u>0.2632</u>	<u>0.3605</u>	<u>0.1995</u>	<u>0.2068</u>
CFPRec	0.2310	0.3085	0.1588	0.1836	0.2771	0.3606	0.1971	0.2190	0.3421	0.4253	0.2432	0.2730
Improve	13.2%	10.8%	11.2%	15.4%	14.5%	12.5%	30.3%	19.0%	30.0%	17.5%	21.9%	32.0%

Ablation Study

1. 去除 past preference encoder 中的辅助任务；
2. 去除 future preference extractor；
3. 只进行单步未来偏好预测；
4. 将 future preference extractor 中的 Attention 替换为平均池化；
5. 去除 past preference encoder 和 future preference extractor

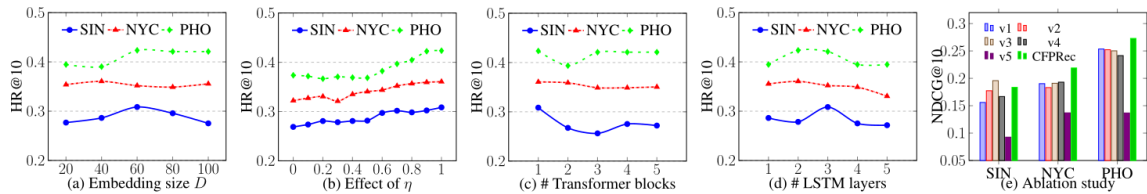


Figure 3: Parameter sensitivity analysis and ablation study of CFPRC on the three datasets.

Conclusion

论文最主要的思路，也就是对目标 POI 的左右上下文进行建模，这一点其实并不复杂，但关键是如何表示用户在未来的偏好需求。这篇论文给出的方法是通过区分当前轨迹和历史轨迹，从历史轨迹中学习规律，作为未来偏好的依据。对于位置，时间，类别的处理，这篇论文单独提出来 3 个辅助任务，这样的方法就消融实验的结果来看似乎也算可行。

下周计划

阅读一篇论文

精读小论文，排查修改语法、格式、单词拼写等小错误

确定小论文目标期刊，完成收尾工作进行投稿