

文献笔记

9.3-9.9

1. Time-sensitive POI Recommendation by Tensor Completion with Side Information

(1) 研究内容：使用张量完成模型范式，同时考虑社会以及空间对 POI 推荐的影响，以实现更有效的时间感知 POI 推荐

(2) 创新点及要点：

- 张量完成问题利用观察到的条目将用户项目时间张量分解为用户、项目和时间的低等级嵌入矩阵，这样就可以跟踪底层的低等级子空间结构，以填补时间感知推荐的缺失条目
 - 现有的张量完成方法忽略了 LBSNs 中可用的社会空间背景信息，这对 POI 推荐非常重要，因为人们倾向于与他们的朋友分享他们的偏好，同时近处的事物比远处的事物对用户更有影响力。
- 本文利用社交网络和 POI 位置的侧面信息来增强张量完成模型范式，以实现更有效的时间感知的 POI 推荐
 - 提出了一个基于 social Hausdorff 距离函数的正则化损失头，以优化重建的张量
 - 用位置熵来量化不同 POI 的受欢迎程度，以防止非常受欢迎的 POI 被过度代表，而压制其他更多样化的 POI 的出现
 - 为了解决负面采样的敏感性，通过将观察到的张量中所有未标记的条目视为负样本来训练整个数据的模型，同时重写损失函数来减少计算成本
- 如果两个用户是朋友，他们倾向于访问相同或在附近的 POI
- 在本文中，旨在利用 LBSN 中的社会空间信息作为侧面信息来改善基于神经张量完成的 POI 推荐。
 - 采用光谱法来获得初始用户、POI 和时间嵌入的粗略估计，以进行嵌入初始化
 - 用包括所有未标记数据在内的整个数据来训练神经张量模型，以消除负采样的缺点
 - 为了解决在所有张量项上直接计算最小二乘损失的高计算成本，通过重写这个损失函数，使计算和学习更有效率
- social Hausdorff 距离函数的反向传播公式
 - Hausdorff 距离度量测量两组点之间的距离。在模型设计中，希望最小化每对朋友的两组签到位置之间的 Hausdorff 距离，以执行社会同质性理论，但存在一些问题：

- * 原始 Hausdorff 距离度量的输入是位置，而不是本文中的神经张量模型的输出（即每个用户-POI-时间交互的概率）
- * 即使修改本文的 social Hausdorff 距离函数以允许 user-POI-time 交互概率，原始 Hausdorff 距离函数中的最小化算子 $\min(\cdot)$ 就其输入而言并不是一个平滑函数，因此它不支持反向传播
- 需要修改这个算子来支持反向传播学习。提出的 social Hausdorff 距离函数使模型能够整合社会关系和 POI 位置，以规范张量完成公式。
- 为了提高推荐的多样性，利用位置熵来推荐那些整体访问频率较低的 POI，考虑到它们通常比每个人都访问的地方更能揭示真实的用户社交偏好。
- 本文的贡献：
 - 提出了基于 social Hausdorff 距离的损失函数的反向传播公式，使模型能够利用社交图和 POI 位置来规范张量补全公式
 - 将社交空间损失头与位置熵 POI 权重相结合，以提高推荐的多样性
 - 用包括所有未标记数据的整个数据来训练模型，以克服张量补全中负采样的缺点，并重写最小二乘损失以降低时间复杂度
 - 使用频谱方法仔细初始化潜在用户/POI/时间嵌入，以避免在训练期间被困在静止点
- 实验所回答的问题：
 - 是否有必要将时间维度纳入推荐而不是考虑只是一个 user-POI 交互矩阵？
 - TCSS 能否在时间感知推荐方面优于现有最先进的张量补全模型？
 - 时间维度上的时间粒度如何影响推荐的性能？
 - 不同类别兴趣点的表现是否一致？
 - 消融研究：TCSS 的训练性能与使用负采样和其他初始化方法的产品相比如何？
 - 模型超参数如何影响结果质量？

(3) 问题描述:

- 使用 $\chi \in R^{I \times J \times K}$ 来表示 LBSN 的签到张量, 其中 I、J 和 K 分别表示用户数、POI 和时间单位（时间间隔）
- 集合 $\Omega = \{(i, j, k)\}$ 包含 χ 中所有条目的索引
- 给定一个社交图 $G = (V, E)$ ，每个节点 v_i 对应一个 LBSN 用户。若两个用户 v_i 和 $v_{i'}$ 是朋友，那么在 v_i 和 $v_{i'}$ 之间会有一条边 $e_{i,i'}$ 。用 l_j 来表示第 j 个 POI 的位置，它是一个经纬度的元组。
- 问题被表述为用社交图 $G = (V, E)$ 和 POI 的地理位置 $l_j | \forall j \in 1, \dots, J$ 将张量 χ 重构为 $\hat{\chi}$
- 目标是估计未观察到的条目 $\hat{\chi}_{i,j,k}$ 的值作为用户 i、POI j 和时间间隔 k 之间的分数，以推荐高分条目

(4) 模型:

- 模型框架：
 - 通过谱初始化方法将每个用户、POI 和时间间隔与嵌入相关联
 - 每个条目的值通过神经网络从相应的嵌入中学习
- 模型的新颖处：
 - 使用具有两个头的混合损失函数
 - * 加权 Hausdorff 距离头来利用社会空间信息
 - * 原始张量与重构张量在整个数据上的均方误差
 - 使用联合训练的策略来优化重建的张量
- 模型架构：
 - 图 1

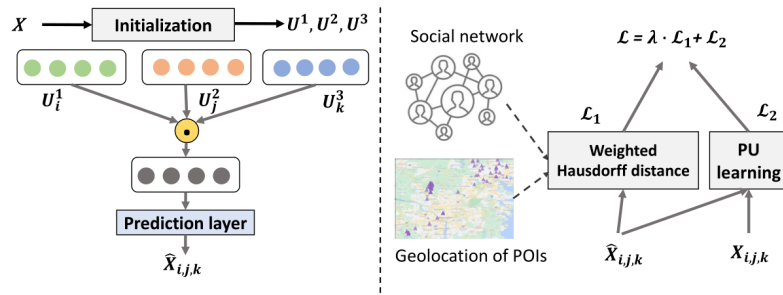


Fig. 2. The Model Framework

图 1:

- Embedding Initialization:
 - 张量分解模型通常使用梯度下降 (GD) 作为优化算法
 - 对于初始点的选择, 本文采用一种频谱方法, 用于获得初始用户、POI 和时间单位嵌入的粗略估计
 - * 将张量 χ “展开” 为三个子空间矩阵: 模式 1、模式 2 和模式 3 矩阵化 $A \in R^{I \times (JK)}$ 、 $B \in R^{J \times (IK)}$ 和 $C \in R^{K \times (IJ)}$
 - 其中 $A_{i,(j-1)K+k} = \chi_{i,j,k}$, $B_{j,(i-1)K+k} = \chi_{i,j,k}$ 和 $C_{k,(i-1)I+j} = \chi_{i,j,k}$
 - * 为了估计 rank-r 因子, 一种方法是探索 AA^T 、 BB^T 和 CC^T 的主要子空间
 - 将三个矩阵的对角元素置零, 然后使用所有非对角矩阵的 top-r 特征向量作为每个子空间中的估计因子
 - $U^1 = \text{eigen}((AA^T)|_{\text{offdiag}}, r) \in R^{I \times r}$
 - $U^2 = \text{eigen}((BB^T)|_{\text{offdiag}}, r) \in R^{J \times r}$
 - $U^3 = \text{eigen}((CC^T)|_{\text{offdiag}}, r) \in R^{K \times r}$
 - * 将所有对角线条目归零, 因为对角线条目对主要方向的影响太大, 应该降低权重
 - 这种初始化方法粗略估计的张量因子可以快速收敛

- Tensor Factorization Formulation

- 通过使用相应的嵌入向量计算每个条目 $X_{i,j,k}$ 的值, 对用户、POI 和时间间隔之间的三元交互进行建模
 - * 计算一个等于它们的元素乘积的向量:

$$\cdot \phi = (U_i^1, U_j^2, U_k^3) = U_i^1 \odot U_j^2 \odot U_k^3$$
 - * 预测 $\chi_{i,j,k}$ 的值, 将向量 $\chi(U_i^1, U_j^2, U_k^3)$ 通过参数 $h \in R^r$ 的密集层:

$$\cdot \hat{\chi}_{i,j,k} = h^T(U_i^1 \odot U_j^2 \odot U_k^3) = \sum_{t=1}^r h_t U_{i,t}^1 U_{j,t}^2 U_{k,t}^3$$
- 将 $\hat{\chi}_{i,j,k}$ 视为 $\chi_{i,j,k}$ 的概率, 即用户 i 在时间间隔 k 内访问 POI j
- 注意到 CP 模型是本文公式的一个特例。与 CP 相比, 本文的具有可学习参数 h 的模型更具表现力, 且可以捕获推荐系统中多个因素的更复杂的交互。
- 为了优化嵌入 U_1, U_2, U_3 和参数 h , 可以简单地最小化观察到的条目的平方误差 $(\chi_{i,j,k} - \hat{\chi}_{i,j,k})^2$ 。然而, 这个错误头本身并没有利用 LBSN 中丰富的社会空间信息
- 设计了一个具有两个损失头的混合损失函数: 一个是根据观察到的条目计算的预测误差头, 另一个是一种新颖的社交豪斯多夫距离函数, 用于测量 POI 的朋友子集彼此之间的距离

- Social Hausdorff Distance

- 根据社会同质性理论, 社交网络倾向于形成具有相似属性或兴趣的节点集群
- 托布勒的第一地理定律指出, 近处的事物比远处的事物更相关
- 用 v_i 来表示 LBSN 中第 i 个用户的顶点, 形式上, 如果用户 $v_{i'}$ 是 LBSN 中另一个用户 v_i 的朋友, 则 $v_{i'}$ 的签到集倾向于与 v_i 的签到重叠或至少接近。
- 本文在损失函数中添加了一个正则化项, 以在重建的张量中强制执行这种社会空间集群结构。
- Hausdorff 距离是衡量两个点集彼此相距多远的度量。用 $S(v_i)$ 表示用户 v_i 将访问的潜在 POI, 并用 $N(v_i)$ 表示 v_i 的朋友访问过的 POI 集:
 - * $S(v_i) = \{j | \exists k \in \{1, \dots, K\}, \hat{\chi}_{i,j,k} > 0\}$
 - * $N(v_i) = \{j | \forall (v_i, v_{i'}) \in E : \exists k \in \{1, \dots, K\}, \hat{\chi}_{i',j,k} = 1\}$
- $S(v_i)$ 和 $N(v_i)$ 之间的平均 Hausdorff 距离 (AHD) 表示为:

$$\cdot d_{AH}(S(v_i), N(v_i)) = \frac{1}{|S(v_i)|} \sum_{j \in S(v_i)} \min_{j' \in N(v_i)} d(j, j') + \frac{1}{|N(v_i)|} \sum_{j' \in N(v_i)} \min_{j \in S(v_i)} d(j, j')$$
- 图 2

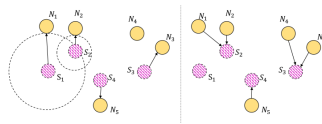


Fig. 3. Average Hausdorff Distance (Arrows Highlight Nearest Neighbors)

图 2:

- 图 2 说明了 AHD 计算, 粉色虚线圆圈代表用户 v_i 可能感兴趣的那些潜在 POI, 黄色圆圈是 v_i 的朋友签入的真实 POI。对于每个粉红色虚线节点 $j \in S(v_i)$, $\min_{j' \in N(v_i)} d(j, j')$ 是 j 与其在 $N(v_i)$ 中的最近邻居之间的距离
- 为了用社会同质性和托布勒第一地理定律规范损失函数, 则需要最小化 $S(v_i)$ 和 $N(v_i)$ 之间的 AHD。对此有两个问题:

- * 本文的神经张量模型估计了每个 user-POI 时间交互的概率, 而不是 POI 的位置。该函数必须相对于神经张量模型的输出是可学习的。 $S(v_i)$ 的潜在位置可能与 LBSN 中的整个 POI 集一样大, 需要推广 AHD 定义以允许 $S(v_i)$ 是具有发生概率的位置集

- * 最小化函数 $\min(\cdot)$ 就其输入而言不是一个平滑函数, 因此它不支持反向传播

- 设计了一个基于 AHD 思想的加权 Hausdorff 损失函数:

$$\begin{aligned}
 * d_{WH}(S(v_i), N(v_i)) &= \frac{1}{|A+\epsilon|} \sum_{j \in S(v_i)} p_{i,j} \min_{j' \in N(v_i)} d(j, j') + \frac{1}{|N(v_i)|} \sum_{j' \in N(v_i)} M_\alpha [p_{i,j} d(j, j') + \\
 &\quad (1 - p_{i,j} d_{max})] \\
 \cdot p_{i,j} &= 1 - \prod_{k=0}^K (1 - \hat{\chi}_{i,j,k}) \text{ 估计用户 } v_i \text{ 将检查第 } j \text{ 个 POI 的概率} \\
 \cdot A &= \sum_{j \in S(v_i)} p_{i,j} \text{ 是取加权距离平均的归一化因子}
 \end{aligned}$$

- Diversifying Recommendation by Location Entropy

- 理想情况下, 用户 v_i 希望看到与 v_i 当前兴趣非常匹配但尚未被很多人访问过的推荐 POI
- 使用位置熵的概念来衡量 POI 的受欢迎程度。设 $\Phi_{i,j} = \{ \langle i, j, k \rangle \mid \chi_{i,j,k} = 1 \}$ 为用户 i 在 POI j 的所有签到的集合, 设 $\Phi_j = \{ \langle i, j, k \rangle \mid \chi_{i,j,k} = 1 \}$ 是所有用户在 POI j 的签到集合, 那么位置熵定义为:

$$\begin{aligned}
 * \varepsilon_j &= - \sum_{i: |\Phi_{i,j}| > 0} \frac{|\Phi_{i,j}|}{|\Phi_j|} \log \frac{|\Phi_{i,j}|}{|\Phi_j|} \\
 \cdot \varepsilon_j &\text{ 的高值意味着 POI } j \text{ 被不同的用户访问}
 \end{aligned}$$

- 最终用户 v_i 的社交 Hausdorff 距离变为:

$$\begin{aligned}
 * d_{WH}(S(v_i), N(V_i)) &= \frac{1}{|A+\epsilon|} \sum_{j \in S(v_i)} p_{i,j} e_j \min_{j' \in N(v_i)} d(j, j') + \frac{1}{|N(v_i)|} \sum_{j' \in N(v_i)} e_{j'} \\
 &\quad M_\alpha [p_{i,j} d(j, j') + (1 - p_{i,j} d_{max})]
 \end{aligned}$$

- 最终的社交 Hausdorff 损失函数定义为所有用户的社交 Hausdorff 距离之和:

$$* \mathcal{L}_1 = \sum_{v_i \in V} d_{WH}(S(v_i), N(v_i))$$

- Learning with Whole Data

- 大多数现有的张量完成模型都是为了最小化原始数据张量 χ 和重建张量 $\hat{\chi}$ 之间的差异而制定的
- 在本文中, 在整个数据上而不是通过负采样来训练模型。然后最小化所有条目的均方误差

$$* \mathcal{L}_2 = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K w_{i,j,k} (\chi_{i,j,k} - \hat{\chi}_{i,j,k})^2$$

· 其中 $w_{i,j,k}$ 是类平衡的入口权重:

$$w_{i,j,k} = \begin{cases} w_+, & \text{if } \text{entry}(i,j,k) \in \Omega \text{ is positive} \\ w_-, & \text{if } \text{entry}(i,j,k) \in \Omega \text{ is unlabeled} \end{cases} \quad (0.1)$$

· 该式的时间复杂度为 $O(I \times J \times K)$

– 重写损失函数以降低时间复杂度:

$$\begin{aligned} * \mathcal{L}_2 = & \sum_{(i,j,k) \in \Omega_+} ((w_+ - w_-) \hat{\chi}_{i,j,k}^2 - 2\chi_{i,j,k} \hat{\chi}_{i,j,k}) + w_- \sum_{r_1=1}^r \sum_{r_2=1}^r h_{r_1} h_{r_2} \left(\sum_{i=1}^I U_{i,r_1}^l U_{i,r_2}^l \right) \\ & \left(\sum_{j=1}^J U_{j,r_1}^2 U_{j,r_2}^2 \right) \left(\sum_{k=1}^K U_{k,r_1}^3 U_{k,r_2}^3 \right) \end{aligned}$$

• Loss Function

– 将社交空间损失头 L_1 和最小二乘损失头 L_2 组合成最终的损失函数 L 来训练模型:

$$* L = \lambda L_1 + L_2$$

– 使用联合训练的策略通过直接最小化最终损失 L 来优化神经网络模型中的参数

(5) 实验:

• 数据集:

- Gowalla, November 2010 to May 2011
- Yelp
- Foursquare, April 2012 to January 2014
- GMU-5K

• 评测指标:

- hit ratio (Hit)
- mean reciprocal rank (MRR)

• Result:

– 图 3

TABLE I
RESULTS COMPARISON

Model		Gowalla		Yelp		Foursquare		GMU-5K	
		Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR
Matrix completion	MCCO	0.3309	0.1804	0.3951	0.1847	0.2880	0.1367	0.3321	0.2192
	PureSVD	0.3779	0.2107	0.4307	0.2087	0.5584	0.2856	0.6651	0.3772
POI recommendation	STRNN	0.3203	0.1908	0.3294	0.1363	0.2586	0.1288	0.4847	0.3481
	STAN	0.5239	0.3311	0.5112	0.3345	0.4723	0.3215	0.6328	0.2667
	STGN	0.5230	0.4127	0.4880	0.2485	0.5101	0.3185	0.4646	0.2810
	LFBCA	0.3513	0.2470	0.3945	0.1575	0.3541	0.1909	0.3828	0.1963
Tensor completion	CP	0.4544	0.2683	0.5522	0.2479	0.5156	0.2955	0.7072	0.4917
	Tucker	0.3742	0.2168	0.5902	0.2701	0.5585	0.3188	0.6989	0.4864
	P-Tucker	0.8162	0.3793	0.6255	0.2627	0.7843	0.3401	0.7473	0.3981
	NCF	0.7891	0.3453	0.6574	0.3057	0.8079	0.4279	0.8077	0.4629
	NTM	0.6181	0.3074	0.1719	0.0839	0.7699	0.3886	0.7897	0.3493
	CoSTCo	0.7571	0.2806	0.5565	0.2106	0.8336	0.3252	0.7465	0.4420
	TCSS	0.9177	0.6206	0.7276	0.3408	0.9298	0.6133	0.9598	0.6376

图 3:

• Ablation Study:

- Social Hausdorff distance

- * Self-Hausdorff 使用用户 v_i 已经访问过的 POI 集合替换公式中的 $N(v_i)$ ，以消除损失中的社会影响
- * Zero-out 仅使用 L_2 进行训练，忽略与用户 v_i 的最近 POI 的距离大于阈值 σ 的任何 POI
- Initialization
 - * 验证谱方法初始化潜在因子的有效性
 - 朴素随机初始化, 将 U_i^1 、 U_j^2 和 U_k^3 与随机向量相关
 - one-hot 编码代替了本文的初始化方法, one-hot 编码的变体使用 NCF 的方法, 索引每个用户 i (分别是 POI j 或时间 k), 其对应位置为 1, 其他为 0, 然后使用可学习的嵌入层将此高维向量转换为嵌入
- Learning on Whole Data
 - * 引入了一种变体, 该变体采用负采样策略来随机采样一些负条目, 其数量等于观察到的条目的数量
- 图 4

TABLE II
ABLATION STUDY

Model Variants	Gowalla		Yelp		Foursquare		GMU-5K	
	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR
Random initialization	0.8833	0.6107	0.6892	0.3275	0.9163	0.6095	0.9045	0.5196
One-hot initialization	0.8688	0.6036	0.6704	0.3061	0.8851	0.5613	0.8968	0.4926
Remove G_2 ($\lambda = 0$)	0.8442	0.5763	0.6308	0.2896	0.8670	0.5115	0.8421	0.4854
Negative sampling	0.8549	0.4098	0.5637	0.2218	0.8917	0.4348	0.9231	0.5550
Self-Hausdorff	0.8614	0.5858	0.6478	0.3029	0.8783	0.5354	0.8654	0.5071
Zero-out	0.8574	0.5571	0.6538	0.3064	0.8321	0.5248	0.8047	0.5012
Full-Fledged TCSS	0.9177	0.6206	0.7276	0.3408	0.9298	0.6133	0.9598	0.6376

图 4:

- Analysis on POI Types and Time Granularity

- Effect of POI Category
 - * 研究推荐性能如何根据不同类型的 POI 而变化
 - * 图 5

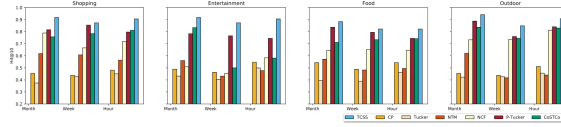


Fig. 4. Hit@10 on Different Categories

图 5:

- * 图 6

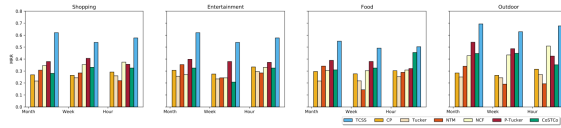


Fig. 5. MRR on Different POI Categories

图 6:

- Effect of Time Granularity
 - * 时间粒度变化时的模型性能
 - * 图 7

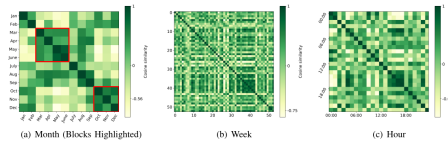


图 7:

- * 图 8

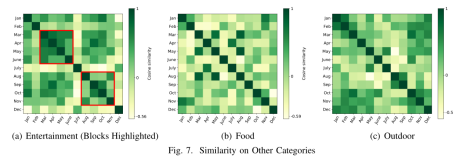


图 8:

- Setting of weights
 - 由于负面条目的数量远大于正面条目的数量，因此给予正面条目更多的权重以平衡它们在训练期间的重要性
 - 图 9

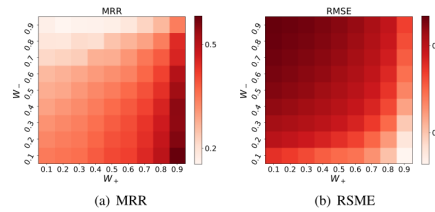


Fig. 8. Effect of Different Weight Combinations (Gowalla)

TABLE III
PERFORMANCE WITH DIFFERENT (w_+, w_-)

(w_+, w_-)	RMSE		Hit@10	MRR
	Positive	Negative		
(0.9, 0.1)	0.4254	0.1627	0.9061	0.5875
(0.95, 0.05)	0.4197	0.1601	0.9077	0.6008
(0.99, 0.01)	0.4148	0.1577	0.9177	0.6206
(0.995, 0.005)	0.4163	0.1593	0.9198	0.6198
(0.999, 0.001)	0.4171	0.1602	0.9184	0.6039

图 9:

- Training Efficiency
 - 验证初始化方法和重写的损失函数的训练效率
 - 图 10
- Parameter Sensitivity

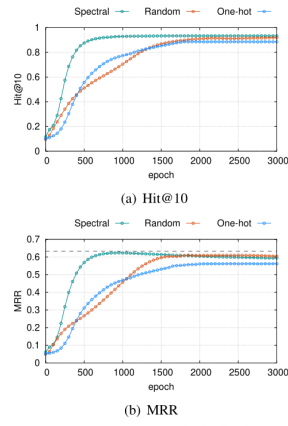


Fig. 9. Effectiveness of Initialization

TABLE IV
TRAINING TIME (Q_E EPOCH)

Methods	Gowalla	Yelp	Foursquare
Original Loss: Eq (14)	2.29×10^5 s	2.49×10^5 s	4.51×10^5 s
Negative Sampling	30.12 s	11.03 s	30.61 s
Rewritten Loss: Eq (15)	0.13 s	0.11 s	0.17 s

图 10:

— 张量秩 r 的值（即嵌入的长度）在张量完成中起着至关重要的作用

* 图 11

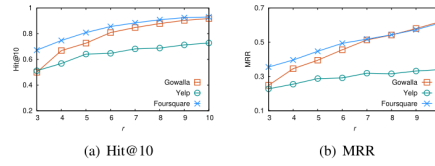
Fig. 10. Varying r

图 11:

— λ 的值表示社会豪斯多夫距离在损失函数 $L = \lambda L_1 + L_2$ 中的权重

* 图 12

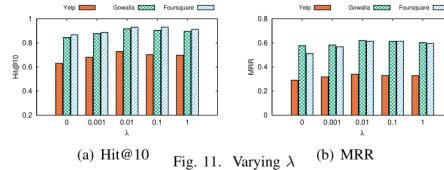
Fig. 11. Varying λ

图 12:

• A Case Study

— 进行一个研究案例来调查 TCSS 的推荐分数

* 图 13

— 研究了不同方法的分数如何沿时间维度变化

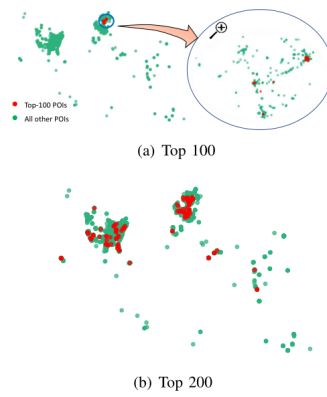


Fig. 12. POIs with High Scores

图 13:

* 图 14

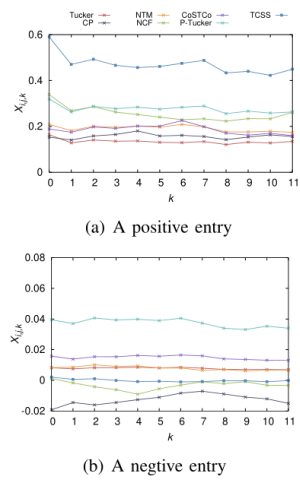


Fig. 13. Score Along the Time Dimension

图 14:

2. Modeling Spatio-temporal Neighbourhood for Personalized Point-of-interest Recommendation

- (1) 研究内容：利用图结构将用户和地点一起建模，从而捕捉用户对地点的个人和动态偏好，同时关注用户对位置的距离/时间的接受程度
- (2) 创新点及要点：
 - 将带有时间信息的知识图谱（称为 TKG）引入到 POI 推荐中，包括带有时间戳的用户和地点
 - 克服现有的基于序列和基于 KG 的推荐方法的局限性

- 在 TKG 的基础上, 提出了一个空间-时间图卷积注意力网络 (STGCAN), 一个通过动态捕捉空间-时间邻域来学习用户对 TKG 的偏好的新型网络
- 在 STGCAN 中, 在 TKG 上构建接受场, 在每个时间戳上分别聚合用户和位置的邻域。而将空间-时间间隔作为用户对距离/时间的接受度来衡量, 并进行 self-attention
 - 考虑了用户对位置的个人偏好, 并开发了关注机制来衡量每个用户对距离和时间的接受程度
- 本文中考虑的 POI 推荐的影响因素:
 - 个人偏好: 个人对同一地点的不同偏好
 - 动态偏好: 用户对同一场所不同时间的偏好发生的变化
 - 个人对距离/时间的接受程度: 对于相同距离/时间的场所, 不同的人可能对其有不同的接受程度
- 本文的贡献:
 - 提出了一个空间-时间图卷积注意力网络, 这个网络可以在 TKG 上学习用户/位置之间的动态关联性, 以模拟用户的 POI 兴趣。作者认为 STGCAN 是第一个探索 TKGs 来学习用户在 POI 推荐中的个人偏好的模型
 - 在 STGCAN 中, 考虑了用户对位置的个人偏好, 并在 TKG 上分别构建了用户/位置感知域, 以捕捉用户/位置的邻域来学习表征
 - 在 STGCAN 中, 开发了一种注意力机制, 可以衡量每个用户对距离和时间的接受程度, 从而增强 POI 推荐

(3) 问题描述:

- 模型的输入是用户的历史访问序列 Y 和位置 GPS 信息, 以及 TKG G ; 输出是一个位置概率矩阵 P 。 Y 中的每个序列都包含三个属性 u_i, l_i, t_i , 表示 u_i 在时间 t_i 访问 l_i
- Definition of Temporal Knowledge Graph: TKG G 表示为四元组 $G = (e_u, r, e_l, t) | e_u \in E_u, e \in R, e_l \in E_l, t \in T$, 其中 e_u 和 e_l 表示用户实体和位置实体; 关系 r 表示“访问”关系的强度; t 是相关性发生时的时间戳
- Definition of Receptive Field: 将 $H_{U/L,U}^{(D)}$ 表示为具有 D 深度的用户/位置感受野; $e_{u'}/l_{l'}$ 作为用户/位置邻居实体; $N^{u/l}$ 作为邻居实体集

(4) STGCAN:

- STGCAN 有四个主要层
 - 嵌入层, 将输入编码为潜在嵌入
 - 用户邻居聚合层, 聚合 TKG 上的用户邻居实体
 - 位置邻居聚合层, 聚合 TKG 上的位置邻居实体
 - 时空间隔感知注意层, 通过注意力机制将用户/位置的表示与时空间隔相结合
- 整体架构:
 - 图 15

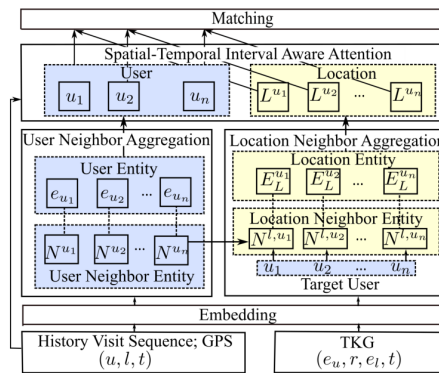


Figure 3: The architecture of the proposed STGCAN model.

图 15:

- Embedding Layer
 - 将历史访问序列 Y 的原始时间戳映射为 $24 * 7$ 小时（一周内的一小时间隔）
 - 将用户 U 、位置 L 、时间戳 T 的集合编码为低维潜在嵌入度量: $E_U \in R^{N \times E}$, $E_L \in R^{M \times E}$, $E_T \in R^{K \times E}$, 其中 E 是嵌入大小的维度
- Neighbour Aggregation Layer
 - 基于 TKG 上的时间信息，将每个时间戳的用户和位置实体各自的邻域与构建的用户/位置接收域进行聚合
 - User Neighbour Aggregation Layer
 - * 用于聚合 TKG 上的用户邻居实体以进行用户表示
 - * 对于 G 中的四元组 (e_u, r, e_l, t) ，将 e_u 的用户邻居实体 $e_{u'}$ 定义为在时间 t 之前与 e_u 共享相同位置实体的实体，形成四元组 $(e_{u'}, r, e_l, t')$
 - * 通过时间戳选择最近的 K 个邻居实体，公式化为 $N_t^u = e_t^{u',1}, e_t^{u',2}, \dots, e_t^{u',K}$
 - * 图 16 给出了目标用户实体及其用户邻居实体在 \mathcal{G} 上的三深度用户感受野上的说明性示例

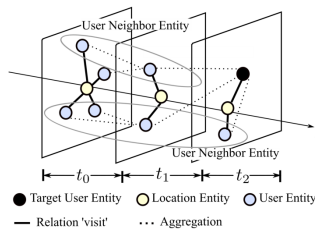
Figure 4: A target user entity (black) with a three-depth user-receptive-field on G_t . Firstly, the purple entities in G_{t_0} are aggregated to update the purple entities in G_{t_1} . Then, those purple entities in G_{t_1} are aggregated to update the target black entity in G_{t_2} .

图 16:

- * 为了区分 K 个相邻实体与目标用户的相关性， \mathcal{G} 上的关系 r 由时空评分函数 I 重新定义

* 函数 I 进一步分为 I_s 和 I_t ，分别计算空间分数和时间分数

* 最终的时空分数 $\lambda_{st}^{u,u'}$ 输出为:

$$\lambda_{st}^{u,u'} = I = I_s + I_t = I_t\{(e_u, r, e_l, t), (e_{u'}, r, e_l, t')\} = \exp(-\gamma|t - t'|)$$

* 获得了 K 个用户邻居实体，它们的 softmax 归一化时空分数具有点生成。目标用户表示由 K 个相邻实体通过线性变换集成，公式如下：

$$\begin{aligned} \cdot \alpha_{st}^{u,i} &= \frac{\exp(\lambda_{st}^{u,i})}{\sum_{i=1}^K \exp(\lambda_{st}^{u,i})} \\ \cdot e_t^u &< -e_t^u + \sigma\left(\sum_{i=1}^K \alpha_{st}^{u,i} \cdot e_t^{u',i}\right) \end{aligned}$$

* 目标用户的表示将因此更新，聚合过程将在 $H_U^{(D)}$ 的每个深度重复：

$$\cdot e_t^u[h+1] = e_t^u + \sigma\left(\sum_{i=1}^K \alpha_{st}^{u,i} \cdot e_t^{u',i}[h]\right)$$

– Location Neighbour Aggregation Layer

* Location-receptive-field 用于聚合 TKG 上的相邻实体以进行位置表示

* 根据用户对位置的个人喜好，位置邻居将由用户和他的邻居选择

* 对于用户实体 e_u ，将 e_l 的位置邻居实体定义为 $e_{l'}$ ，作为在时间 t 之前与用户 u 的用户邻居实体 $e_{u'}$ 相连的位置实体，公式化为 $(e_{u'}, r, e_{l'}, t')$ ，连接关系可以理解为语义上的“访问”。

* 通过排序的时间戳将每个用户的用户邻居实体 N^u 连接起来，公式化为 $N^u = \text{concat}(N_t^u | i \in T)$

* 选择具有最新时间戳 \hat{t} 且不大于 t 的位置实体作为邻居位置实体，通过下界函数： $\hat{t} = \text{lower_bound}(t)$

* 因此每个位置将同时有 K 个位置邻居实体用户 u ，公式化为 $N_t^{l,u} = e_{t_1}^{l'_1,u}, e_{t_2}^{l'_2,u}, \dots, e_{t_K}^{l'_K,u}$

* 图 17 通过 \mathcal{G} 的三深度位置感受野给出了一个位置实体及其邻居实体的说明性示例

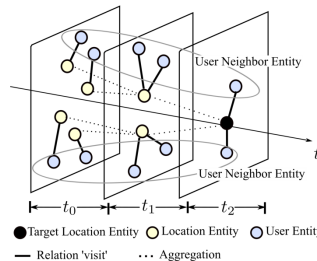


Figure 5: A target location entity (black) with a three-depth location-receptive-field on G_t . Firstly, the yellow entities in G_{t_0} are aggregated to update the yellow entities in G_{t_1} . Then, those yellow entities in G_{t_1} are aggregated to update the target black entity in G_{t_2} . User neighbour entities (purple) help to choose the yellow entities.

图 17:

* 为了区分 K 个位置相邻实体与目标位置的相关性， \mathcal{G} 上的关系 r 在用户感受野中通过相似的时空评分函数 I 进行评分。在这里，将 I_t 中的指数替换为倒数运算以增强时间间隔奖励，并利用 I_s 中的 GPS 坐标来计算 harsine 距离。位置实体的时空分数计算为：

$$\begin{aligned}
& \cdot \lambda_{st}^{l,u} = I = I_s + I_t \\
& \cdot \begin{cases} I_s = I_s(GPS_l, GPS_{l'_u} = \frac{1}{\theta + H(GPS_l, GPS_{l'_u})}) \\ I_t = I_t\{(e_u, r, e_l, t), (e_{u'}, r, e_{l'}, t')\} = \frac{1}{\theta + |t - t'|} \end{cases} \quad (0.2)
\end{aligned}$$

* 目标位置表示由其自身及其 K 个与用户感受野中类似的组合功能相邻实体的表示：

$$\begin{aligned}
& \cdot \alpha_{st}^{l,u} = \frac{\exp(\lambda_{st}^{l,u})}{\sum_{i=1}^K \exp(\lambda_{st}^{l,u})} \\
& \cdot e_t^{l,u} < -e_t^{l,u} + \sigma(\sum_{i=1}^K \alpha_{st}^{l,u} \cdot e_u^{l'_u}) \\
& \cdot e_t^{l,u}[h+1] = e_t^{l,u} + \sigma(\sum_{i=1}^K \alpha_{st}^{l,u} \cdot e_u^{l'_u}[h])
\end{aligned}$$

• Spatial-Temporal Interval Aware Attention Layer

- 将空间距离和时间间隔测量为时空间隔的注意层，并分配权重组合内部两者的相对比例用户和用户之间的表示
- 通过每个序列之间的空间距离 Δs 和时间间隔 Δt 计算时空间隔，以它们的最小/最大值 $\Delta s/t_{min/max} \in RN$ 作为上/下界：

$$\begin{aligned}
& * d = d_t + d_s \\
& * \begin{cases} d_s = \frac{[vsl \cdot (\Delta s_{max} - \Delta s + vsu \cdot (\Delta s - \Delta s_{min}))] \cdot inv_s}{(\Delta s_{max} - \Delta s_{min}) + \theta} \\ d_t = \frac{[vtl \cdot (\Delta t_{max} - \Delta t + vtu \cdot (\Delta t - \Delta t_{min}))] \cdot inv_t}{(\Delta t_{max} - \Delta t_{min}) + \theta} \end{cases} \quad (0.3)
\end{aligned}$$

- 通过应用 self-attention 将获得的时空间隔注入到用户/位置表示中，将时间嵌入作为注意力输入结合到表示中，以强调顺序影响

$$* U_{Att} = W_d \cdot (E_U + E_L + E_T) + b$$

$$* L_{Att} = W^Q[W_d \cdot (E_L + E_T)]^T + b$$

- 访问概率由用户和位置表示的叉积计算：

$$* P = (L_{Att}^T \times U_{Att})$$

• Neural Network Training

- 位置 l_j 在 P 中的概率分布可以表示为 $p(l_j|u_i, r_m, t_k)$ ，表示用户 u_i 在时间 t_k 到访问 r_m 旁边的位置 l_j 的访问概率
- 训练使用交叉熵作为损失函数，并在每个训练步骤随机抽取 s 个负样本：

$$* \mathcal{L} = - \sum_{j \in L} (\log \sigma(p_j) + \sum_{j' \in L}^s \log \sigma(1 - p_{j'}^N)) - \lambda \Theta$$

• 图 18

(5) 实验：

• 数据集：

- NYC、TKY 和 Gowalla
- 图 19

Algorithm 1: STGCAN

Input: User history visit sequence Y ; GPS; TKG \mathcal{G}
Data: negative sample size s ; neighbour entity size K ; receptive field depth H ;
Output: Prediction function: $\mathcal{P} = p(l_j|u_t, r_m, t_k)$
while *Network not converge* **do**
 for (u, l, t) **in** Y **do**
 $N_t^u \leftarrow \text{Get-User-Neighbour-Entity}(u, \mathcal{G}, K)$
 for h **in** H **do**
 $v^u[h+1] = \text{Agg-User-Entity}(N_t^u, v^u[h])$
 $L_t^u \leftarrow \text{Get-Loc-Neighbour-Entity}(u, l, \mathcal{G}, K)$
 for h **in** H **do**
 $e^u[h+1] = \text{Agg-Loc-Entity}(L_t^u, e^u[h])$
 $v_t^u, e_t^u \leftarrow \text{ST-Aware-Attention}(v_t^u[H], e_t^u[H])$
 Calculate predicted probability:
 $\hat{y}_t^u = p(v^u, e^u)$
 Update parameters by gradient descent
 return \mathcal{P}

图 18:

	NYC	TKY	Gowalla
# users	1,083	2,293	18,737
# locations	5,136	7,708	32,510
# check-ins	147,938	443,477	1,496,597

Table 1: Details of Datasets

图 19:

- 评估指标:
 - Recall@k
 - $Imp@k = \frac{\hat{R}_{STGCAN@k} - R@k}{R@k} - \sum_{i < k} Imp@i$
- Results:
 - 图 20

Model	NYC			TKY			Gowalla		
	Recall@1	Recall@5	Recall@10	Recall@1	Recall@5	Recall@10	Recall@1	Recall@5	Recall@10
BPR-MF	0.074	0.148	0.224	0.079	0.179	0.260	0.046	0.128	0.155
TransRec	0.113	0.314	0.408	0.064	0.161	0.268	0.062	0.163	0.243
Tiasrec	0.119	0.252	0.393	0.098	0.222	0.389	0.073	0.201	0.306
Deepmove	0.134	0.300	0.374	0.124	0.288	0.341	0.098	0.200	0.275
PRME	0.157	0.385	0.476	0.117	0.318	0.416	0.089	0.209	0.267
GeoSAN	0.167	0.401	0.544	0.125	0.316	0.371	0.106	0.277	0.364
STAN	0.181	0.412	0.593	0.133	0.326	0.397	0.109	0.302	0.400
KGCN	0.024	0.076	0.117	0.023	0.055	0.072	-	-	-
CyGNet	0.207	0.413	0.586	0.109	0.294	0.406	0.112	0.311	0.385
STGCAN	0.257	0.544	0.629	0.171	0.357	0.457	0.129	0.343	0.414
-TKG	0.171	0.471	0.587	0.129	0.328	0.428	0.096	0.268	0.349
-ST	0.209	0.513	0.619	0.161	0.384	0.449	0.107	0.228	0.314

-TKG: STGCAN without Neighbour Aggregation Layer on TKG.

-ST: STGCAN without Spatial-Temporal Interval Aware Attention Layer.

Table 2: Recommendation performance comparison with baselines.

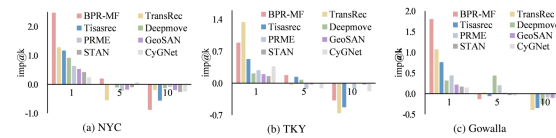


Figure 6: Improvement recall ratios of STGCAN compared with baselines.

图 20:

- Impact of embedding dimension
 - 图 21
- Impact of depth of receptive field

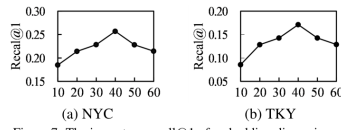


Figure 7: The impact on recall@1 of embedding dimension

图 21:

– 图 22

depth	1	2	3
NYC	0.214	0.257	0.199
TKY	0.157	0.157	0.128
Gowalla	0.128	0.129	0.096

Table 3: The impact on recall@1 of depth of receptive field

图 22:

- Impact of neighbour entity size

– 图 23

neighbour	1	2	4	6
NYC	0.145	0.228	0.257	0.199
TKY	0.114	0.157	0.171	0.129
Gowalla	0.071	0.129	0.126	0.096

Table 4: The impact on recall@1 of neighbour entity size

图 23:

3.Next Point-of-Interest Recommendation with Auto-Correlation Enhanced Multi-Modal Transformer Network.

(1) 研究内容：下一个兴趣点 (POI) 推荐

(2) 创新点及要点：

- 已有的基于递归神经网络的下一个 POI 推荐算法的问题：
 - 大多数模型只考虑连续访问的依赖关系，而忽略了 POI 在痕迹中错综复杂的依赖性
 - POI 序列中分层的性质和子序列的匹配在现有方法中很难建模
 - 大多数的解决方案都忽略了 POI 与密度类别这两个模型之间的相互作用
- 为 next-POI 推荐提出了一种自相关的增强型多模态 Transformer 网络 (AutoMTN)。特别是，AutoMTN 使用 Transformer 网络显式地利用跟踪中所有 POI 的连接。
- 发现子序列级别的依赖关系并关注 POI 和类别序列之间的跨模态交互，将 Transformer 中的 self-attention 替换为自相关机制，并设计了一个多模态网络。
 - POI 的类别和位置之间的交互很重要，因为下一个位置受类别影响
 - 由于该模型无法学习类别的信息。为了预测下一个用户类别以支持下一个用户 POI 的预测，设计了一个双通道 Transformer 网络来分别预测 POI 和类别。

- 将 Transformer 网络中的自注意机制替换为基于序列周期性的自相关机制，以增强双通道 Transformer 网络，该网络可以发现子序列级别的依赖关系和聚合表示
- 模型的核心是定向跨模态自相关，关注 POI 和类别序列之间跨不同时间步长的交互，并潜在地将子序列的信息从一种模态调整到另一种模态
- 基于 RNN 架构的方法推荐 POI 存在的问题：
 - 在短期连续访问方面受到限制，即它们几乎无法模拟时间线上很远的访问之间的隐式连接
 - 忽略了子序列的重要影响，这超出了 POI 级别的序列模式
 - 无法捕捉到 POI 和类别之间的跨模态知识，而 POI 的类别和位置之间的交互很重要，因为下一个位置受类别影响
- 本文的主要贡献：
 - 第一个利用多模态 Transformer 网络获得 POI 和类别序列之间的交叉模态信息
 - 利用多模态 Transformer 网络中的自相关机制来学习子序列模式
- 自相关层：
 - 由于以前的方法基于循环结构，总是忽略不连续访问的信息，故无法在子序列级别提取相关性
 - 利用自相关机制通过子序列关系来扩展信息利用，即通过计算子序列的自相关来发现子序列的依赖关系，并通过时间延迟聚合来聚合相似的子序列。

3 问题定义:

- 令 $U = \{u_1, u_2, \dots, u_{|U|}\}$, $L = \{l_1, l_2, \dots, l_{|L|}\}$, $C = \{c_1, c_2, \dots, c_{|C|}\}$ 分别为用户，POI 和类别
- 对用户 u ，将其历史 POI 序列表示为 $L_u = \{(u, l_t, \tau_t) | t = 1, 2, \dots, N\}$ ，其中 t 代表第 t 次访问
- 对用户 u ，将其历史类别序列表示为 $C_u = \{(u, c_t, \tau_t) | t = 1, 2, \dots, N\}$
- 在历史序列之后为用户 u 推荐 top-K 优选 POI，即为每个用户学习个性化排名函数 f_u ，将用户的历史 POI 和类别序列映射到概率分布目标 POI 集：
 - $f_u = Pr(y \in L | L_u, C_u)$

(4) AutoMTN

- 整体架构：
 - 图 24
- auto-correlation mechanism:
 - 图 25
- Embedding Layer
 - 嵌入层用于将用户、类别、位置和时间信息编码为潜在表示

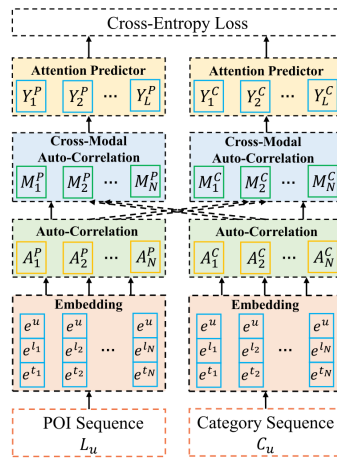


Figure 2: The overall architecture of AutoMTN.

图 24:

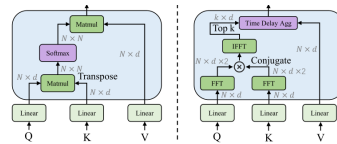


Figure 3: Self-attention (left) and auto-correlation (right).

图 25:

- 在其他层中使用嵌入将输入转换为高维向量，以增强模型的代表能力
- Auto-Correlation Layer
 - 利用自相关机制通过子序列关系来扩展信息利用。即自相关通过计算子序列的自相关来发现子序列的依赖关系，并通过时间延迟聚合来聚合相似的子序列
 - Dependencies of Sub-sequences
 - * 自相关可以反映 POI 序列与其滞后序列之间的时延模式。即根据随机过程理论，真实离散时间过程 χ_t 的自相关 $R_{\chi\chi}(\epsilon)$ 可以表示为

$$R_{\chi\chi}(\epsilon) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \chi_t \chi_{t-\epsilon}$$
 - Time Delay Aggregation
 - * 从计算的所有滞后长度的自相关中选择具有最高自相关的滞后长度，然后对齐这些选定的子序列
 - * 汇总来自选定子序列的信息，聚合可以形式化为：

$$\epsilon_1, \dots, \epsilon_k = \underset{\epsilon \in \{1, \dots, N\}}{\text{argTopk}}(R_{Q,K}(\epsilon))$$

$$\tilde{R}_{Q,K}(\epsilon_1), \dots, \tilde{R}_{Q,K}(\epsilon_k) = SM(R_{Q,K}(\epsilon_1), \dots, R_{Q,K}(\epsilon_k))$$

$$AC(Q, K, V) = \sum_{i=1}^k Roll(V, \epsilon_i) \tilde{R}_{Q,K}(\epsilon_i)$$
 - * 对于 POI 和类别序列的嵌入，自相关机制可以表述为：

$$\begin{aligned} \cdot A^P &= AC(E(L_u)W^{Q^{P_1}}, E(L_u)W^{K^{P_1}}, E(L_u)W^{V^{P_1}}) \\ \cdot A^C &= AC(E(C_u)W^{Q^{C_1}}, E(C_u)W^{K^{C_1}}, E(C_u)W^{V^{C_1}}) \end{aligned}$$

- Cross-Modal Auto-Correlation Layer

- 设计了跨模态自相关, 使一种模态能够从另一种模态接收信息, 即 POI 和类别序列相互获取辅助信息
- 即将 POI 序列视为自相关中的查询, 以在子序列级别提取类别的边信息, 反之亦然
- 跨模态自相关可以表示为
 - * $M^P = AC(A^P W^{Q^{P_2}}, A^C W^{K^{C_2}}, A^C W^{V^{C_2}})$
 - * $M^C = AC(A^C W^{Q^{C_2}}, A^P W^{K^{P_2}}, A^P W^{V^{P_2}})$

- Attention Predictor

- 根据 M^P, M^C 的学习表示来计算候选 POI 和类别的概率
- 嵌入的 POI 候选:
 - * $E(L) = \{e_1^l, \dots, e_{|L|}^l\}$
- 嵌入的类别候选:
 - * $E(C) = \{e_1^c, \dots, e_{|C|}^c\}$
- attention 预测器可以表示为:
 - * $\hat{Y}^P = S(SM(\frac{E(L)M^{P^T} + D^T}{\sqrt{d}}))$
 - * $\hat{Y}^C = S(SM(\frac{E(C)M^{C^T}}{\sqrt{d}}))$
- 应用交叉熵损失函数来量化预测值和 ground truth 的差异:
 - * $\mathcal{L} = -\frac{1}{B} \sum_{b=1}^B y_b^{P^T} \log(\hat{y}_b^P) + \eta y_b^{C^T} \log(\hat{y}_b^C)$

(5) 实验:

- 数据集:

- New York City(NYC),Tokyo(TKY);April 2012 to February 2013

Table 2: Statistics of NYC and TKY datasets.

Dataset	#User	#POI	#Category	#Check-in
NYC	866	7725	345	171753
TKY	2034	11650	313	482677

图 26:

—

- 评估指标:

- average Accuracy@N (Acc@K)
- Mean Reciprocal Rank@K (MRR@K)

- Results and Analysis

- Performance Comparison

- Ablation Study
 - * MTN 用普通的 self-attention 取代了模型中的自相关
 - * AutoTN 删除了模型中的类别通道
- Hyperparameter Analysis
 - * 图 27

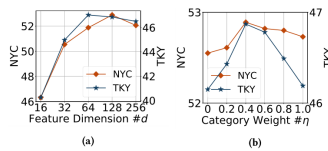


Figure 4: Parameter sensitivity analysis (Acc@10).

图 27:

– 图 28

Table 1: Performance comparison on two datasets. Bold: Best, underline: Second best.

		NYC				TKY			
		Acc@5	Acc@10	MRR@5	MRR@10	Acc@5	Acc@10	MRR@5	MRR@10
Baselines	RNN	0.2814	0.3233	0.1914	0.1970	0.2465	0.2944	0.1714	0.1778
	LSTM	0.3134	0.3651	0.2066	0.2136	0.2634	0.3125	0.1799	0.1865
	STGN	0.3400	<u>0.3879</u>	<u>0.2327</u>	<u>0.2393</u>	0.2786	0.3284	0.1908	0.1975
	LSTPM	0.3326	0.3778	0.2289	0.2326	0.2658	0.3207	0.1855	0.1920
	Flashback	<u>0.3448</u>	0.3877	0.2302	0.2381	<u>0.3353</u>	<u>0.4000</u>	<u>0.2253</u>	<u>0.2340</u>
Ours	MTN	0.4190	0.5066	0.2636	0.2755	0.3697	0.4587	0.2310	0.2491
	AutoTN	0.4063	0.4933	0.2599	0.2689	0.3504	0.4325	0.2297	0.2409
	AutoMTN	0.4370	0.5289	0.2742	0.2867	0.3802	0.4687	0.2434	0.2553
	Improvement	26.74%	36.35%	17.83%	19.81%	13.39%	17.18%	8.03%	9.10%

图 28: