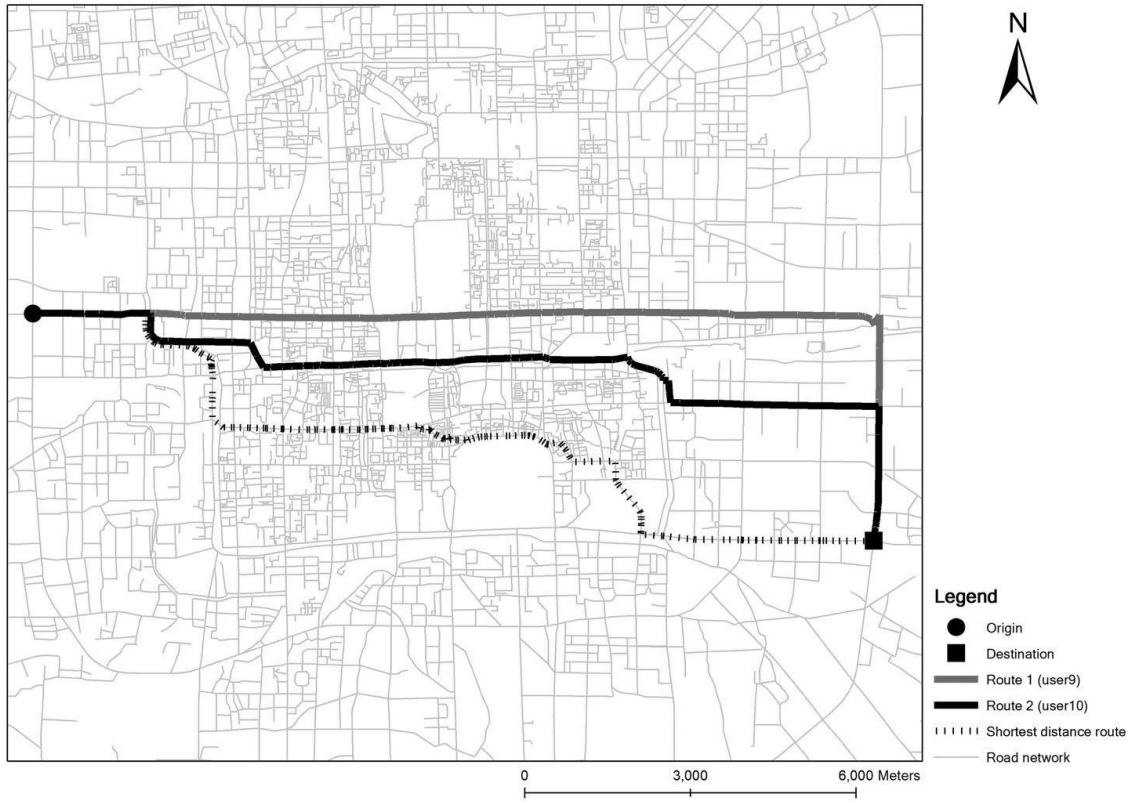
轨迹补全方法调研与介绍

邹增禹 **12.30**

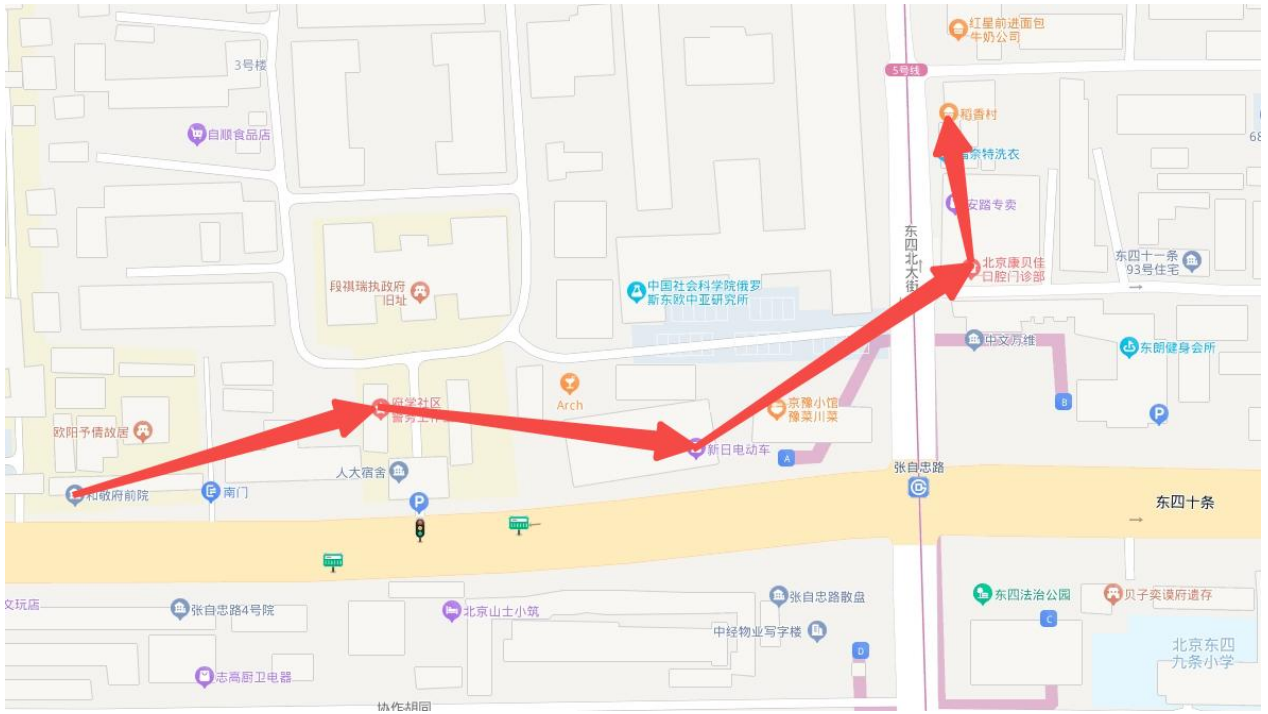
目录

路径补全

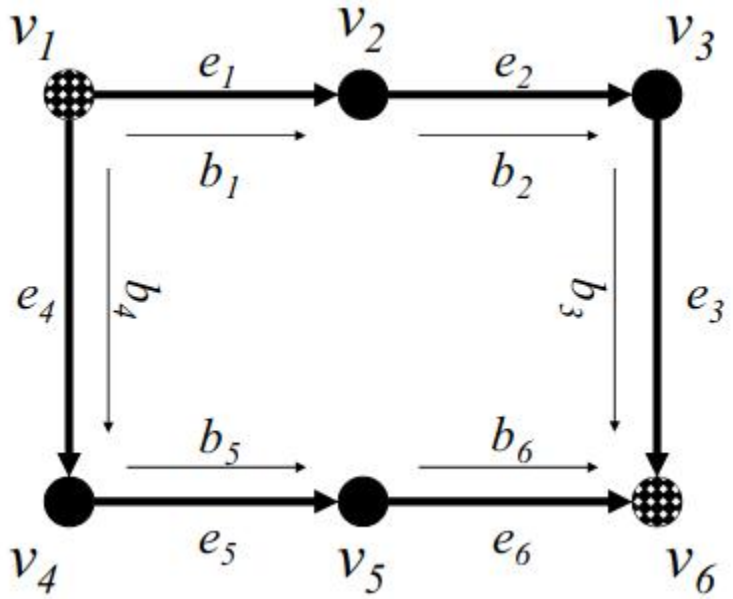
- 依靠路网



- 不依靠路网

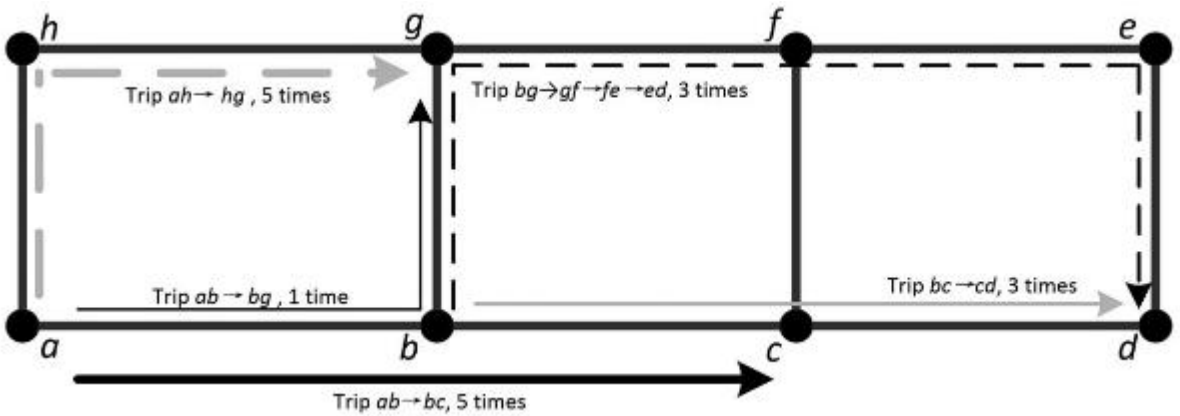


方法1：计算边权



边	权（偏好）
v1->v2	0.2
v2->v3	0.5
v1->v4	0.4
v4->v5	0.6
v3->v6	0.5
v5->v6	0.4

方法1：计算边权——协同过滤



根据用户之前的喜好以及其他兴趣相近的用户的选择来给用户推荐物品

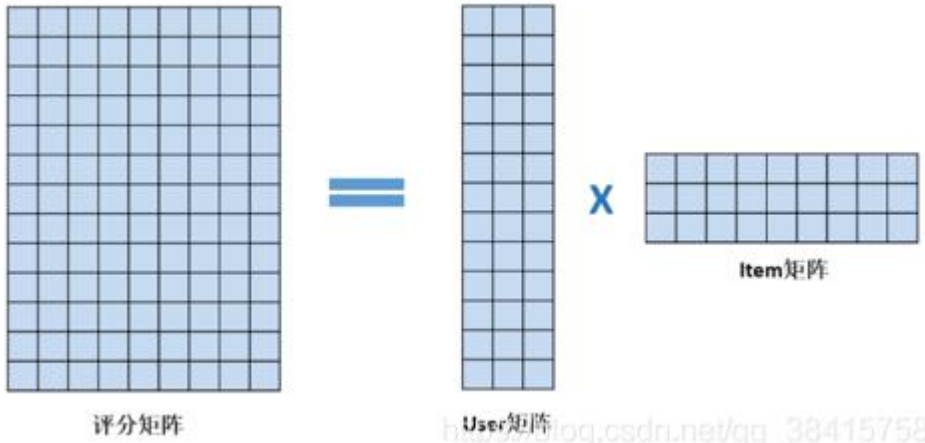
User	Behaviour（Item）	Segment	Time interval	Frequency
A	b_1	*ab*	7 am–8 am	6
	b_2	*bc*	7 am–8 am	5
	b_3	*bg*	7 am–8 am	1
	b_4	*bg*	5 pm–6 pm	3
	b_5	*gf*	5 pm–6 pm	3
	b_6	*fe*	5 pm–6 pm	3
	b_7	*ed*	5 pm–6 pm	3
B	b_2	*bc*	7 am–8 am	3
	b_8	*cd*	7 am–8 am	3
	b_9	*ah*	5 pm–6 pm	5
	b_{10}	*hg*	5 pm–6 pm	5

方法1：计算边权——协同过滤——矩阵分解

道路

$$UB_{m\times n}=\begin{matrix} \\ 用户 \end{matrix}\begin{matrix} \begin{matrix} b_1 & b_2 & \cdots & b_n \end{matrix} \\ \begin{bmatrix} frq(u_1,b_1) & frq(u_1,b_2) & \cdots & frq(u_1,b_n) \\ frq(u_2,b_1) & frq(u_2,b_2) & \cdots & frq(u_2,b_n) \\ \vdots & \vdots & \ddots & \vdots \\ frq(u_m,b_1) & frq(u_m,b_2) & \cdots & frq(u_m,b_n) \end{bmatrix} \end{matrix}\begin{matrix} \\ u_1 \\ u_2 \\ \vdots \\ u_m \end{matrix}$$

6 4 7

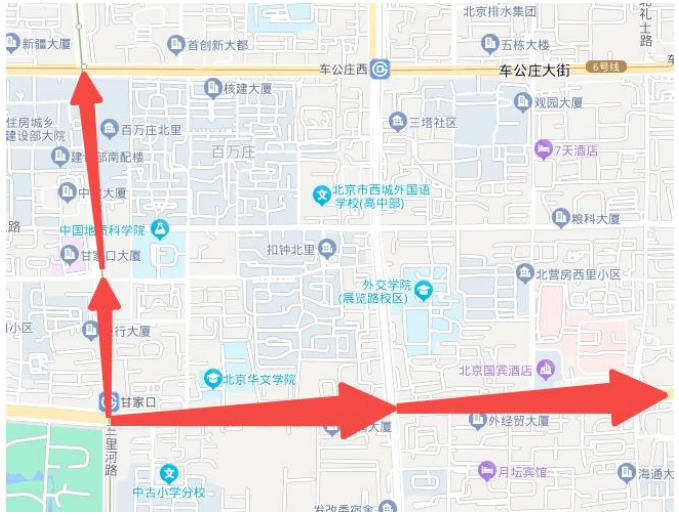


对分解后的user矩阵和item矩阵相乘得到每个用户对每条道路的偏好，值越大，表示用户越喜欢这条道路

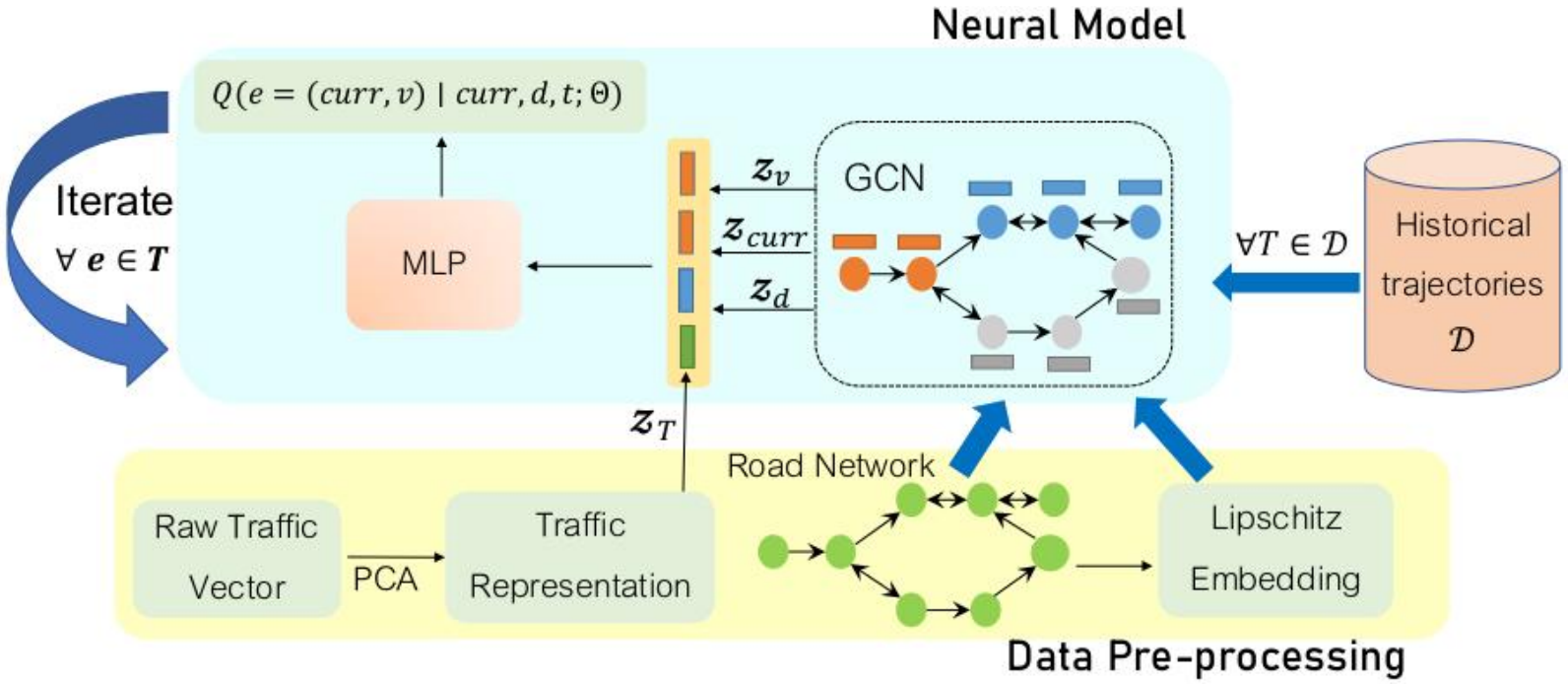
方法1：计算边权——协同过滤

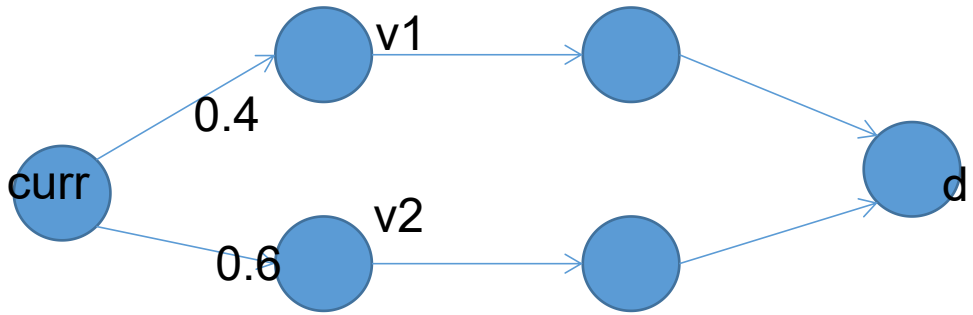
缺点

- 数据稀疏
- 没有起点和终点的信息
- 没有利用地理关系



方法1：计算边权——神经网络

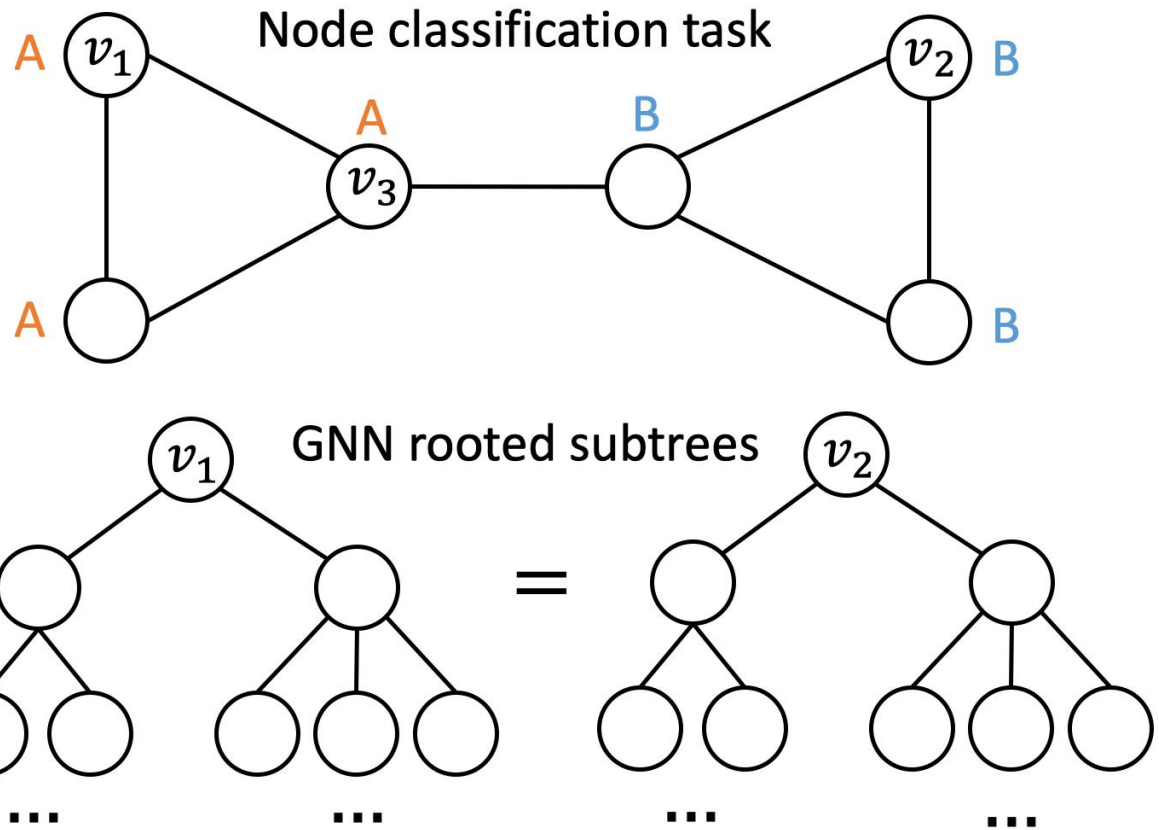


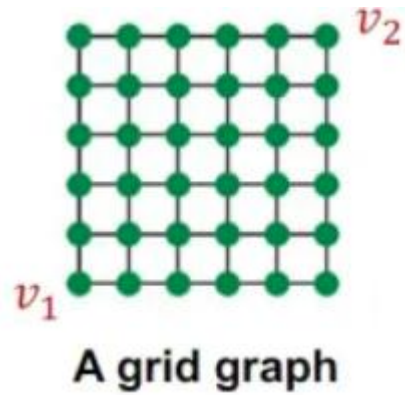


计算下一步走每个相邻点的概率

图神经网络 ->
z_{curr}当前点的 representation
z_v下一个点的 representation
z_d终点的 representation
-> MLP -> softmax

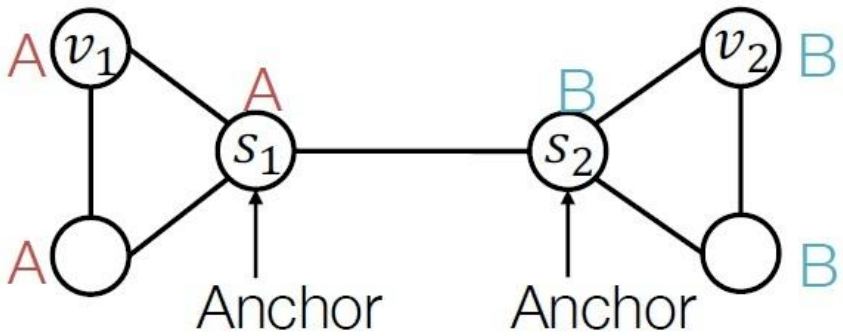
Position-Aware GNN （位置感知GNN）





v1,v2 表征应该相同

Position-Aware GNN （位置感知GNN）

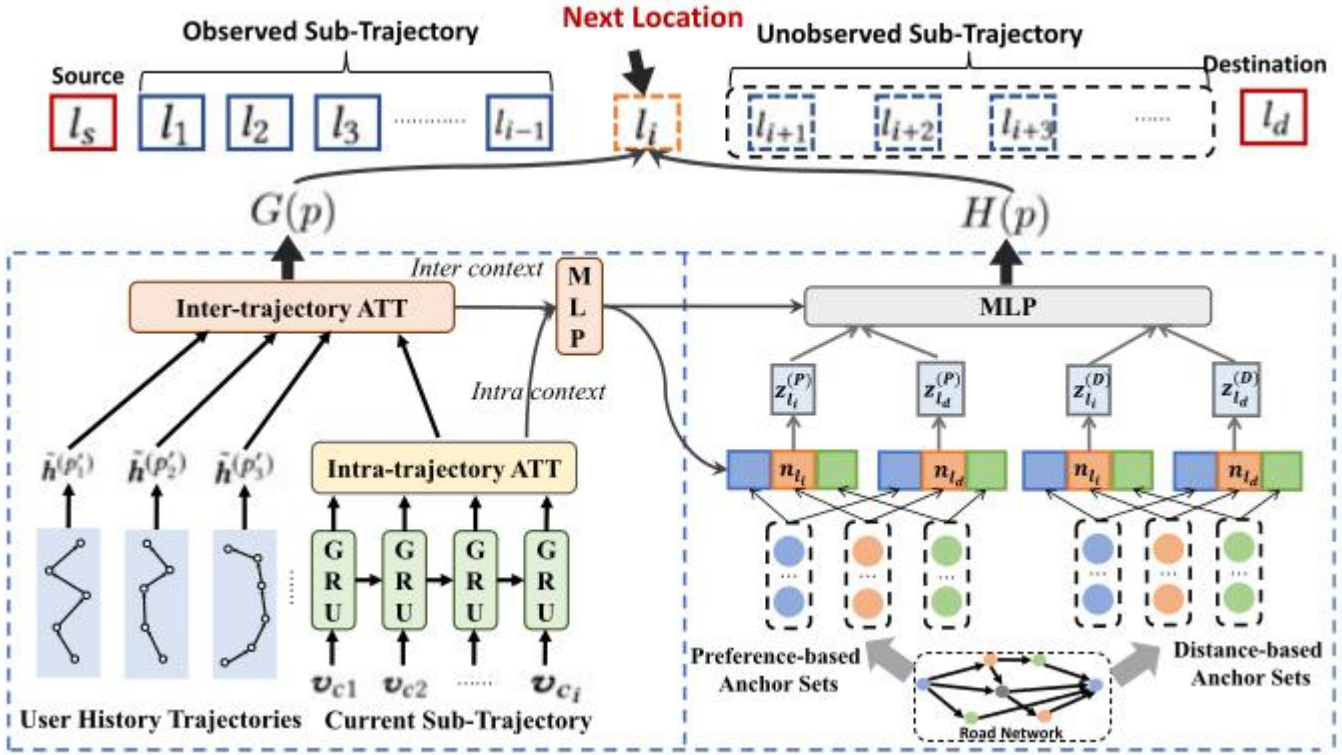


Relative Distances

	s_1	s_2
v_1	1	2
v_2	2	1

锚点：s1, s2
锚点 -> 参照物

方法2：神经网络+A*搜索 F(p) = G(p) + H(p)　　总代价=历史代价+未来代价



$G(l_s \rightarrow l_i)$:
GRU encoder
轨迹内 attention
轨迹间 attention

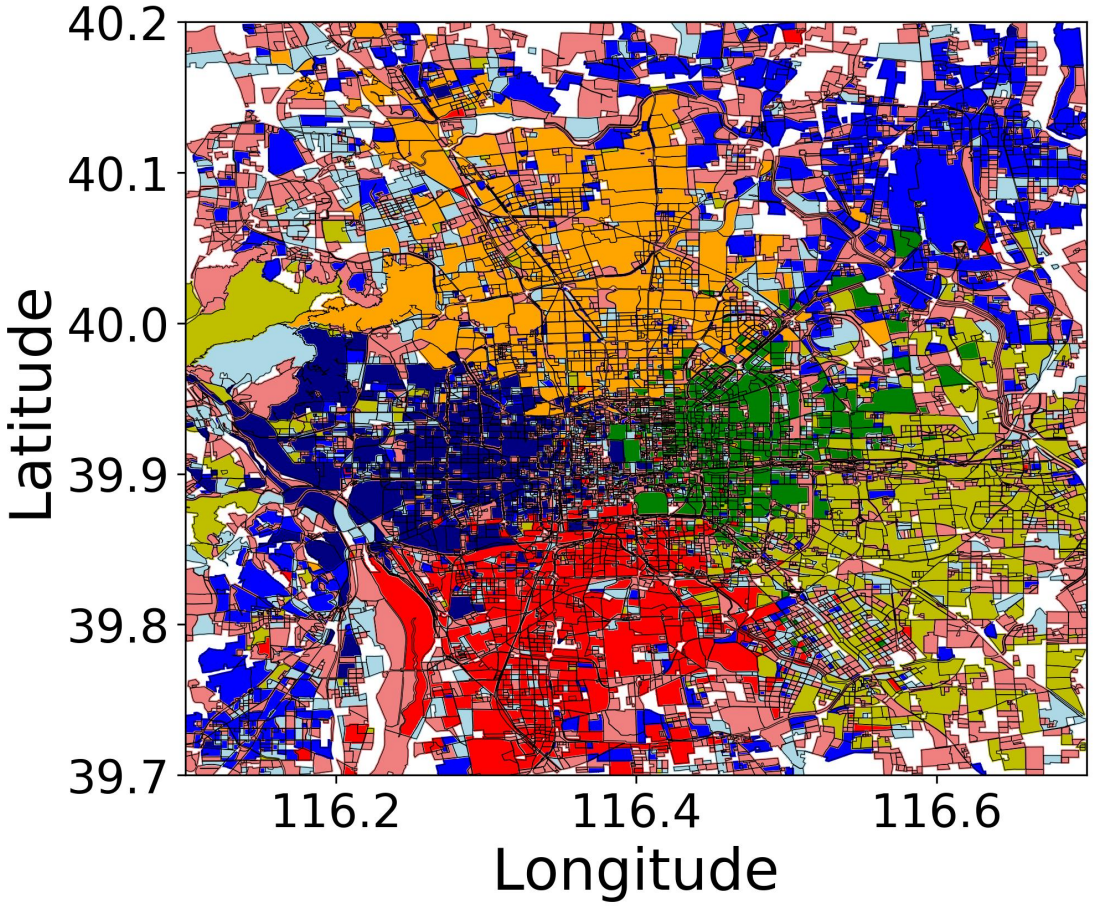
$H(l_i \rightarrow l_d)$:
Position-Aware GNN

$$H(l_i \rightarrow l_d) = \text{MLP}\left(\boldsymbol{h}_i^{(p)}, z_{l_i}^{(D)}, z_{l_d}^{(D)}, z_{l_i}^{(P)}, z_{l_d}^{(P)}\right)$$

需要给出起点和终点

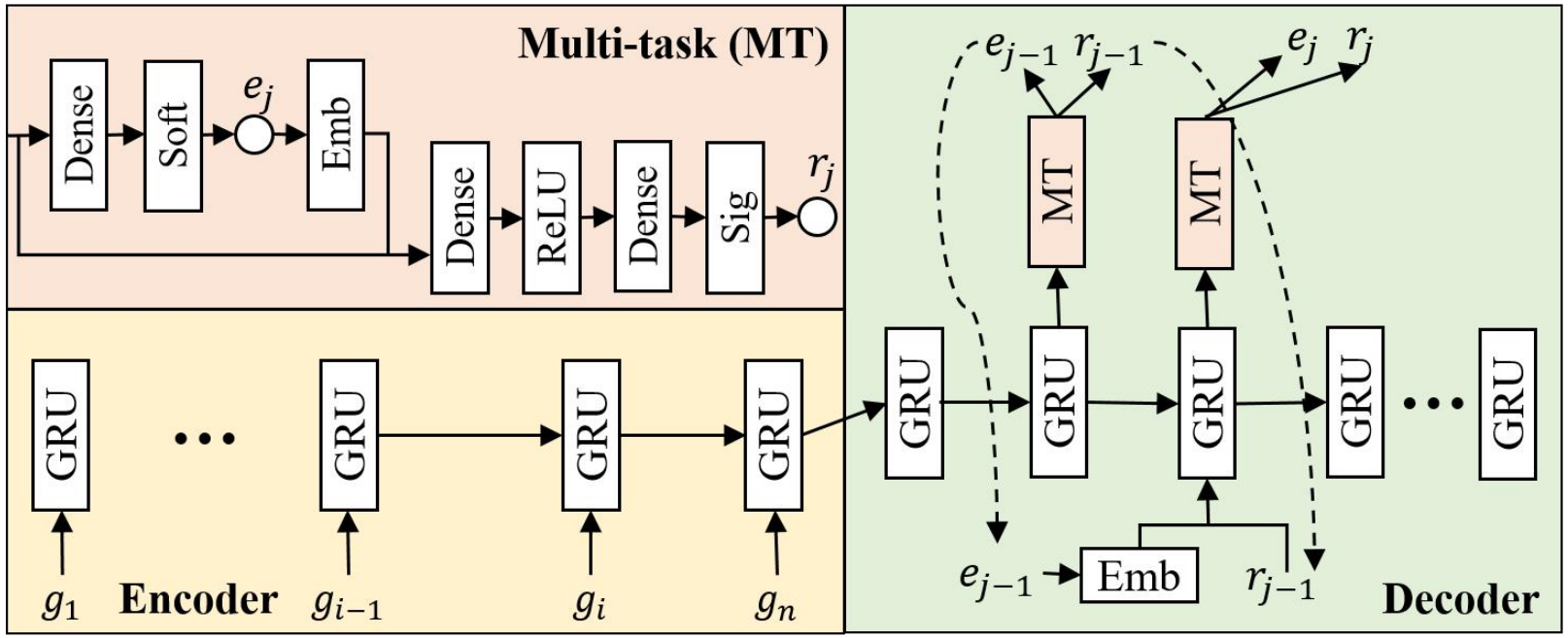
v_{ci}=用户Embedding || 节点Embedding || 日期Embedding || 小时Embedding

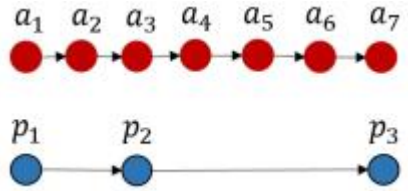
Embedding user, position, time



Embedding 将正整数（ID，下标）转换为具有固定大小的向量，可以被梯度下降优化。表示用户、位置、时间的偏好与特征

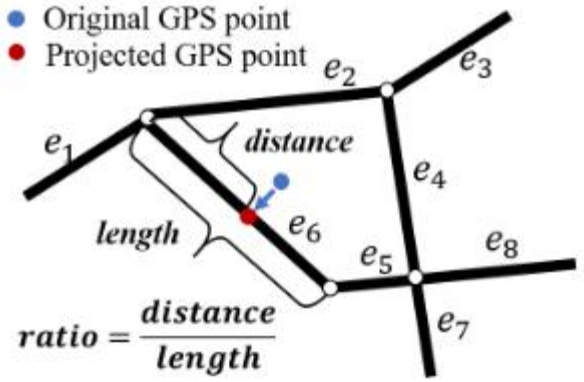
方法3：直接预测下一个点在边和边上的位置





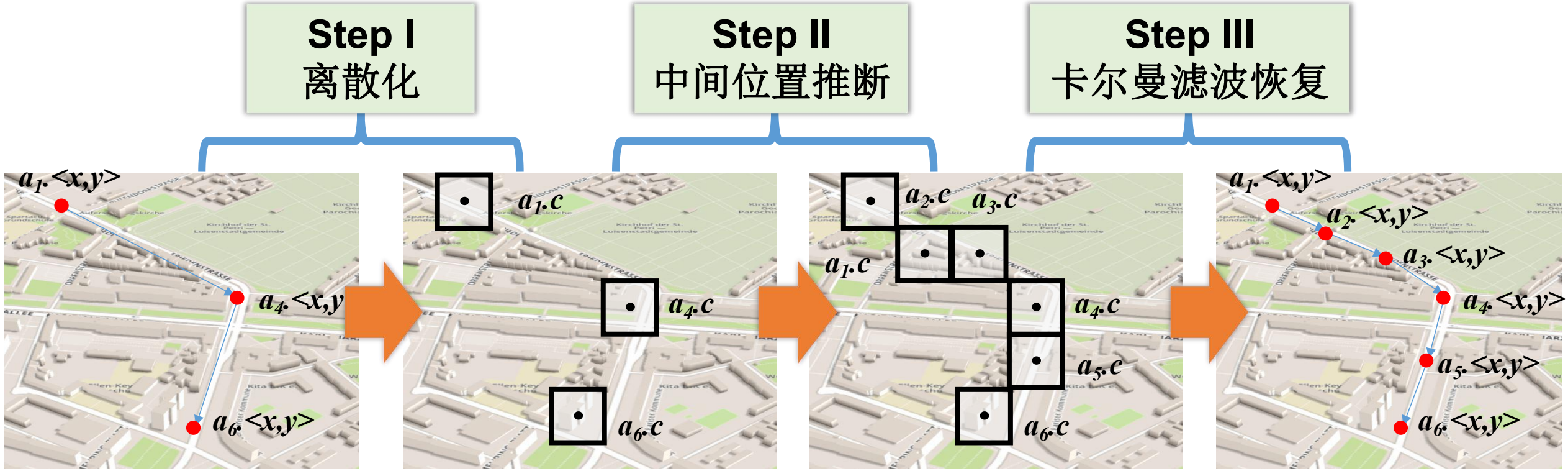
对边进行 Embedding

预测：
边的ID
点在边上的比例

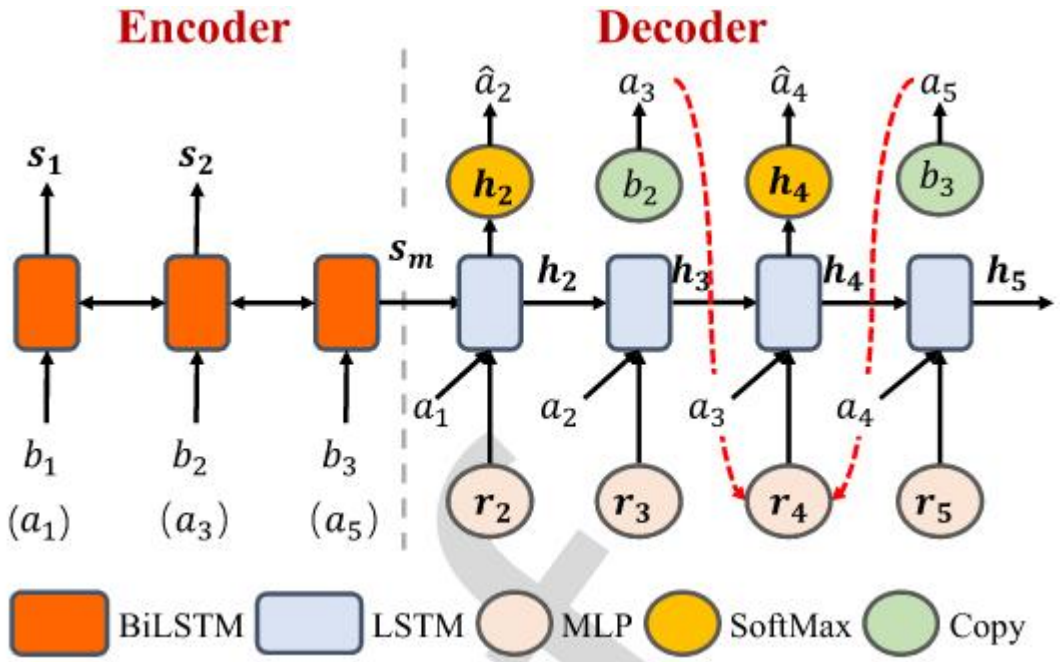


方法4：恢复缺失点+卡尔曼滤波

将平面划分为网格状，对每个网格 Embedding
用网格的中心点替代作为坐标

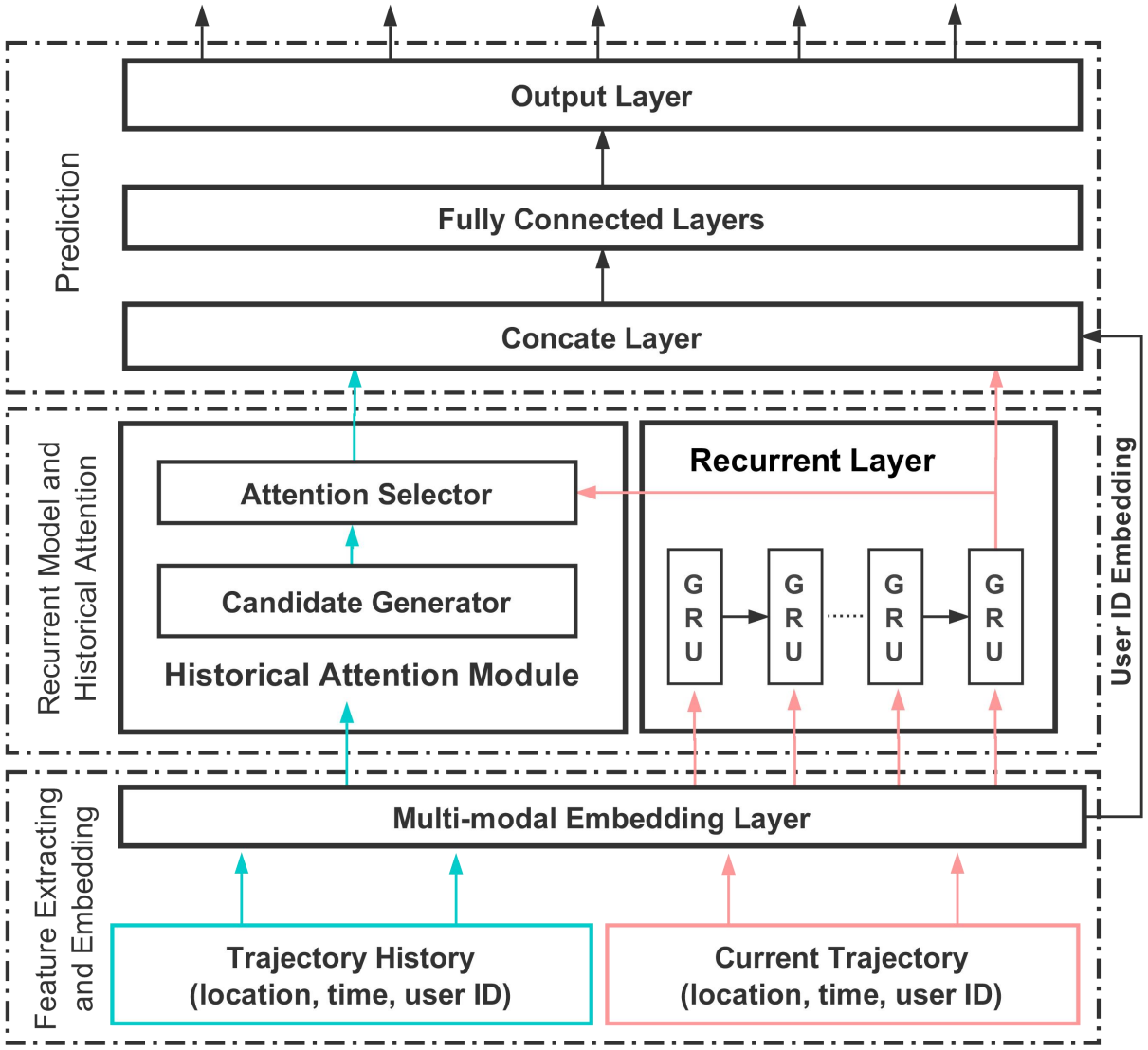


方法4：恢复缺失点+卡尔曼滤波



Encoder-Decoder 架构

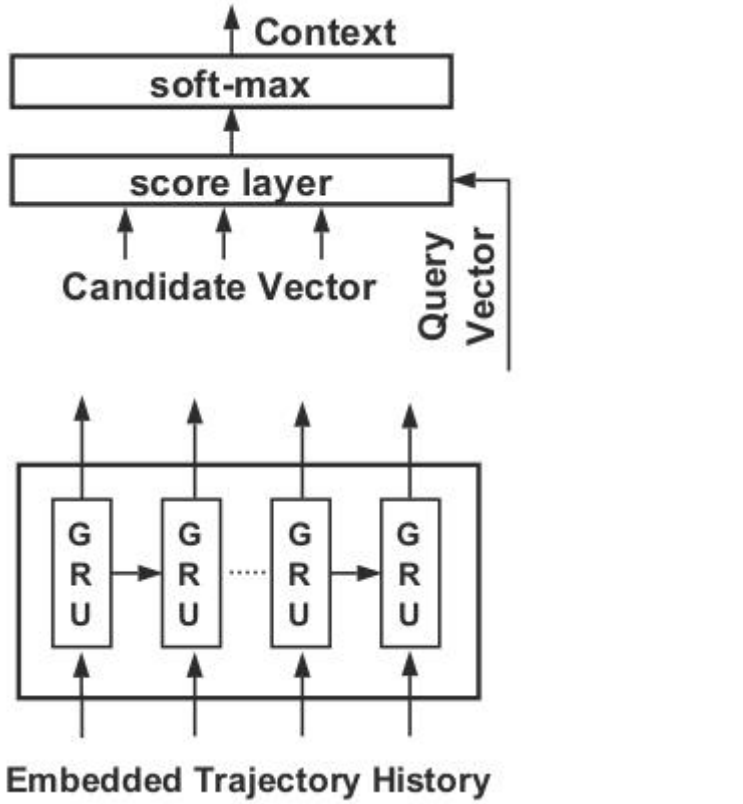
方法5：下一位置预测（预测下一个POI或AOI的位置）



注意力机制：

Q：当前轨迹最后一个输出

K，V：历史轨迹所有的 GRU 输出



总结

- 共性
  - Embedding user, location, time
  - RNN（GRU，LSTM），Encoder-Decoder
  - 注意力机制
  - Position-Aware GNN
  - .....

谢谢收听