

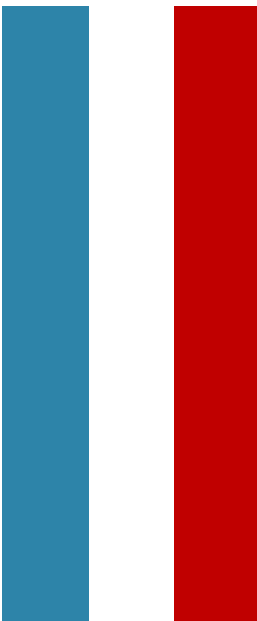


Graph2Route: A Dynamic Spatial-Temporal Graph Neural Network for Pick-up and Delivery Route Prediction

ACM SIGKDD Conference on Knowledge Discovery and Data Mining

汇报人：白云迪

2023/1/28



主要内容

Main Contents

1

背景

2

内容

3

方法

4

实验

5

总结

1 背景

PDRP(Pick-up and Delivery Route Prediction task): 快递员/外卖员的揽件派送问题

现实中，工人的实际路线与给出的预测路线不一致，准确率较差



■ 概括

提出一种基于动态时空图神经网络的P&D路径预测模型Graph2Route，首次将P&D路线预测问题建模为图上的预测问题

2 内容

□ 现有模型：DeepRoute

1/基于序列(Sequence-based)

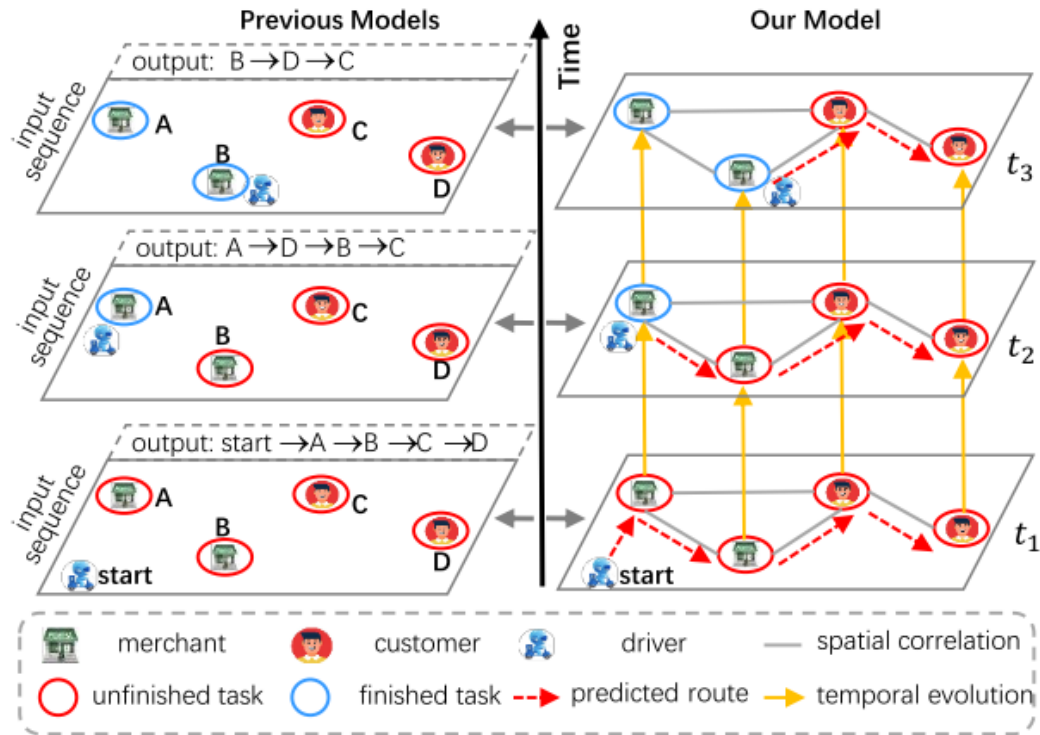
将未完成的任务视为一个序列，使用的是基于序列的编码器（LSTM）**缺乏编码任务之间的时空相关性**

2/输出不合理

基于序列的编码无法避免解码极其不合理的路线

3/缺乏与其他实例之间的联系

随时间的推移，无法对新节点或任务需求变化的节点进行建模



■ 改进：Graph2Route

■ 基于图的动态时空模型

■ 对图结构进行编码

■ 捕获决策上下文进行预测

2 内容

■ 问题定义

□ Input-Graph:

$\mathcal{G}_t^w = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t, \mathbf{E}_t)$: 工人 w 在 t 时刻的时空图

$\mathcal{V}_t = \{v_1, \dots, v_n\}$: 工人的任务节点

$\mathcal{E}_t = \{(i, j) | v_i, v_j \in \mathcal{V}_t\}$: 工人的边节点

$\mathbf{X}_t \in \mathbb{R}^{n \times d_v}$: d_v 节点特征维度

$\mathbf{E}_t \in \mathbb{R}^{n \times n \times d_e}$: d_e 边特征维度

$\mathcal{V}_t^F = \{v | v \in \mathcal{V}_t, x^{FT} \leq t\}$: 已完成的节点, x^{FT} 表示任务的完成时间

$\mathcal{V}_t^U = \{v | v \in \mathcal{V}_t, x^{FT} = -1\}$: 未完成的节点

□ 规则集C: 考虑到服务中存在的路线约束:

1) 先取后送约束、2) 容量约束

$$\mathcal{F}_C(\mathcal{G}_t^w) = \pi_1, \pi_2 \cdots \pi_{|\mathcal{V}_t^U|} \cdots \cdots \cdots (1)$$

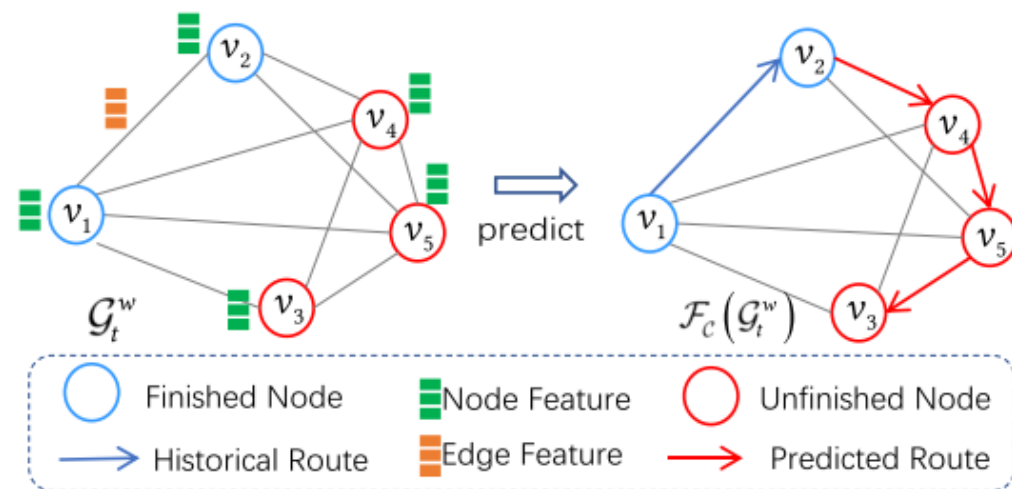


Figure 2: Problem Illustration. In this case, $\mathcal{V}_t^F = \{v_1, v_2\}$ and $\mathcal{V}_t^U = \{v_3, v_4, v_5\}$, the output of the model $\hat{\pi} = [\pi_1, \pi_2, \pi_3]$ is $[4, 5, 3]$.

2 内容

■ Input ST-Graph

在给定t时刻的已完成任务集和未完成任务集 $\mathcal{V}_t = F_t^w \cup \bar{U}_t^w$ 构造一个ST图 $\mathcal{G}_t^w = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t, \mathbf{E}_t)$

在传统GCN只考虑节点特征的基础上，本文同时对边特征和节点特征进行建模，并同时更新

■ 节点特征

■ 节点i的特征向量

包含了时空信息

$$\mathbf{x}_i = (x_i^{Co}, x_i^{AT}, x_i^{PT}, x_i^{FT}, t - x_i^{AT}, x_i^{PT} - t, x_i^{Dis}), \quad (3)$$

地理坐标 | 接收时间 | 承诺到达时间 | 完成时间 | 距工人w的距离

■ 边特征

$$\mathbf{e}_{ij} = (d_{ij}, a_{ij}).$$

两节点的距离 | 相似度

$$a_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are } k\text{-nearest spatial neighbors} \\ 1, & \text{if } i \text{ and } j \text{ are } k\text{-nearest temporal neighbors} \\ -1, & \text{if } i = j \\ 0, & \text{others.} \end{cases} \quad (2)$$

■ 空间：根据两节点空间距离

■ 时间：根据两节点承诺到达时间相近程度

3 方法

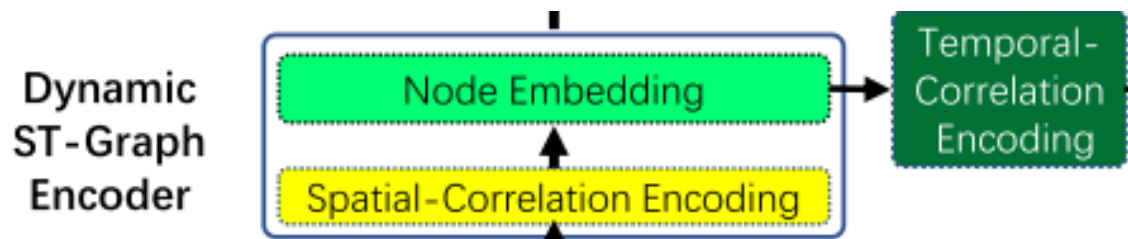
- 动态时空图编码器（Dynamic ST-Graph Encoder）
- 空间编码（Spatial-Correlation Encoding）
使用GCN捕获不同节点之间的时空相关性
与标准GCN不同，引入了边特征

$$\mathbf{h}_i^{l+1} = f(\mathbf{h}_i^l, \text{Agg}\{\mathbf{h}_j^l, \mathbf{z}_{ij}^l : j \in \mathcal{N}_i\})$$
$$\mathbf{z}_{ij}^{l+1} = g(\mathbf{z}_{ij}^l, \text{Agg}\{\mathbf{h}_i^l, \mathbf{h}_j^l\}),$$



$$\mathbf{h}_i^{l+1} = \mathbf{h}_i^l + \sigma(\text{BN}(\mathbf{W}_1^l \mathbf{h}_i^l + \sum_{j \in \mathcal{N}_i} \eta_{ij}^l \odot \mathbf{W}_2^l \mathbf{h}_j^l)) \quad (6)$$

$$\mathbf{z}_{ij}^{l+1} = \mathbf{z}_{ij}^l + \sigma(\text{BN}(\mathbf{W}_3^l \mathbf{z}_{ij}^l + \mathbf{W}_4^l \mathbf{h}_i^l + \mathbf{W}_5^l \mathbf{h}_j^l)), \quad (7)$$



- 时间编码（Temporal-Correlation Encoding）

$\mathbf{H}_t = \text{GRU}(\mathcal{G}_t^w, \mathbf{H}_{t-1})$. 使用标准GPU门控单元对输入图和当前嵌入更新节点嵌入

3 方法

■ 基于图的个性化路线解码器（Graph-based Personalized Route Decoder）

▣ 掩码机制 (Mask Mechanism)

在t时刻

- 1) 屏蔽已经完成的节点 \mathcal{V}_t^F
- 2) 屏蔽通过解码器已经输出的 \mathcal{R}_j
- 3) 屏蔽未访问取货节点的交付节点 $\mathcal{V}_j^d = \{v_d^o \mid v_p^o \notin \mathcal{V}_t^F \text{ and } v_p^o \notin \mathcal{R}_j\}$
- 4) 屏蔽不是输出节点邻居的节点 $\{\mathcal{V} - \mathcal{N}_{v_{\pi_{i-1}}}\}$

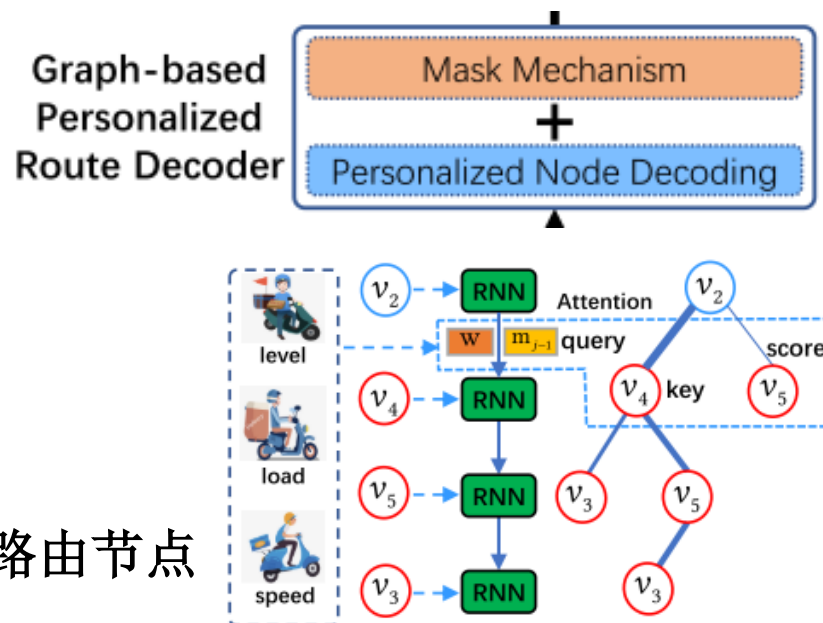
$$\mathcal{V}_{mask}^j = \mathcal{V}_t^F \cup \mathcal{R}_j \cup \mathcal{V}_j^d \cup \{\mathcal{V} - \mathcal{N}_{v_{\pi_{j-1}}}\}.$$

■ 个性化解码（Personalized Node Decoding）

在每个时间步 t ，解码器选择候选节点中概率最大的节点作为路由节点

□ 解码过程

在每个解码步骤中，将选中的节点反馈到RNN中记录当前解码状态，帮助进行下一步选择



$\mathbf{w} \in \mathbb{R}^{d_w}$ 工人的个性化属性: 平均速度、最大装载量

$$u_i^j = \begin{cases} \mathbf{q}^T \tanh(\mathbf{W}_7 \mathbf{h}_i + \mathbf{W}_8 [\mathbf{m}_{j-1}; \mathbf{w}]) & \text{if } i \notin \mathcal{V}_{mask}^j \\ -\infty & \text{otherwise,} \end{cases} \quad (11)$$

$$p_i^j = p(\pi_j = i \mid s, \pi_{1:j-1}; \theta) = \text{softmax}(u_i^j)$$

3 方法

Algorithm 1 Graph2Route.

Input: Input Graph $\mathcal{G}_t^w = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t, \mathbf{E}_t)$ at time t of worker w .

Output: Output predicted service route $\hat{\pi}_t$.

- 1: // Dynamic ST-Graph Encoder
- 2: **for** $l = 1, \dots, L$ **do** // Spatial-Correlation Encoding
- 3: Update node embedding by Eq. (6);
- 4: Update edge embedding by Eq. (7);
- 5: **end for**
- 6: // Carry on the decision context though GRU
- 7: Update \mathbf{H}_t according to Eq. (8);
- 8: // Graph-based Personalized Route Decoder
- 9: $\hat{\pi}_t \leftarrow []$;
- 10: **for** $j = 1, \dots, |\mathcal{V}_t^U|$ **do**
- 11: Mask infeasible nodes in \mathcal{V}_{mask}^j ;
- 12: Calculate the output probability p_i^j by Eq. (12);
- 13: Select output node $\pi_j = \arg \max_k p_k^j$;
- 14: Append π_j into $\hat{\pi}_t$;
- 15: **end for**
- 16: **return** $\hat{\pi}_t$;

循环解码机制，将前一步的输出作为下一步的输入

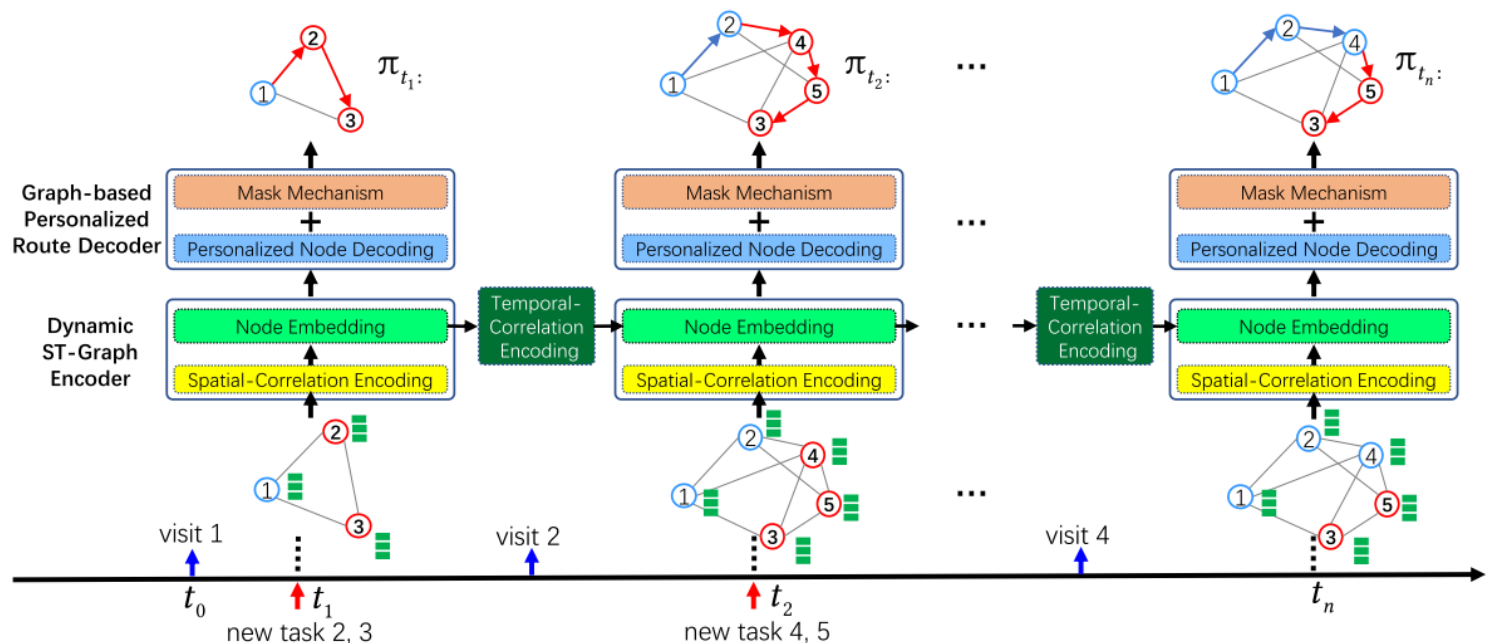


Figure 3: Architecture of Graph2Route.

3 方法：模型训练和预测

■ 样本构建

选择新加入节点的时刻进行图建模

■ 损失函数

在解码过程中会选择概率最大的节点作为输出，将其看作多分类问题，使用交叉熵损失函数

$$\mathcal{L} = - \sum_{w \in \mathcal{W}} \sum_{t \in \mathcal{T}} \sum_{i \in \pi_{t:t'}} y_i \log(p(y_i | \theta)),$$

所有工人 | 所有新加入节点的时刻 | 所有节点

■ 预测过程

每次输入为某一时刻的时空图，不包含新加入节点的过程

4 实验

- 数据集：分别从饿了么与菜鸟获取的两种数据

表示未完成任务的平均数量

Type	Time Range	City	ANUT	#Workers	#Samples
Food-PD	02/02/2020 - 02/29/2020	Dalian	4	916	166,026
Logistics-P	03/29/2021 - 05/27/2021	Shanghai	9	2,344	208,202

- Baseline:

DeepRoute: 包含Transformer的编码器和注意力机制的解码器，采用Pointer架构对整个路线进行循环解码

DeepRoute+: 额外添加一个编码模块模拟工人的决策偏好，同时考虑到最近访问的路线

4 实验

■ 验证方法:

- **KRC(Kendall Rank Correlation):**衡量两个序列之间有序关联的统计标准

任意任务对 (i, j) , 如果 $O_{\hat{\pi}}(i) > O_{\hat{\pi}}(j)$ 并且 $O_{\pi}(i) > O_{\pi}(j)$ 或者 $O_{\hat{\pi}}(i) < O_{\hat{\pi}}(j)$ 并且 $O_{\pi}(i) < O_{\pi}(j)$, 表示任务对一致

$$\text{KRC} = \frac{N_c - N_d}{N_c + N_d}, \quad N_c/N_d \text{表示一致/不一致的数量}$$

- **ED(Edit Distance):** 将一个序列(预测路线)转换为另一个序列(实际路线)所需的最小操作次数
- **LSD(位置平方偏差)和LMD(位置平均偏差):** 衡量预测路线偏离实际路线的程度:

$$\text{LSD} = \frac{1}{m} \sum_{i=1}^m (O_{\pi}(\pi_i) - O_{\hat{\pi}}(\pi_i))^2 \quad \text{LMD} = \frac{1}{m} \sum_{i=1}^m |(O_{\pi}(\pi_i) - O_{\hat{\pi}}(\pi_i))|.$$

- **HR@k:** 前k项命中率, 量化两个序列前k项之间的相似性 $\text{HR@k} = \frac{|\hat{\pi}_{[1:k]} \cap \pi_{[1:k]}|}{k}.$

- **ACC@k:** 计算预测到达时间与真实时间的时间差小于k分钟的比率 $\text{ACC@k} = \prod_{i=0}^k \mathbb{I}(\hat{\pi}_i, \pi_i),$

4 实验

□ 模型在不同的任务数量下评估模型（0-11）与（0-25）

Method	Logistics-P												Food-PD											
	$n \in (0 - 11]$						$n \in (0 - 25]$						$n \in (0 - 11]$						$n \in (0 - 25]$					
	HR@1	ACC@3	KRC	LMD	LSD	ED	HR@1	ACC@3	KRC	LMD	LSD	ED	HR@1	ACC@3	KRC	LMD	LSD	ED	HR@1	ACC@3	KRC	LMD	LSD	ED
TimeRank	26.37%	13.62%	37.76%	2.30	11.54	2.41	25.19%	12.47%	35.44%	2.45	12.84	3.19	45.02%	27.59%	60.78%	0.57	0.78	1.74	44.99%	27.57%	60.77%	0.57	0.79	1.74
DisGreedy	45.98%	26.09%	51.29%	1.72	8.45	2.01	45.43%	24.52%	49.72%	1.84	9.27	2.66	53.16%	39.92%	68.09%	0.47	0.67	1.41	53.12%	39.89%	68.09%	0.47	0.68	1.42
Or-Tools	48.59%	28.04%	54.30%	1.54	6.87	1.95	47.81%	26.26%	52.60%	1.67	7.73	2.61	54.98%	42.32%	70.37%	0.44	0.62	1.35	54.96%	42.29%	70.36%	0.44	0.62	1.35
OSquare	47.03%	24.24%	55.20%	1.52	6.01	2.05	46.32%	22.55%	53.58%	1.64	6.88	2.74	63.43%	44.79%	72.62%	0.45	0.67	1.37	63.39%	44.75%	72.59%	0.45	0.68	1.38
FDNET	49.50%	27.73%	55.75%	1.60	7.59	1.96	48.81%	25.91%	54.08%	1.72	8.38	2.62	64.69%	49.64%	75.46%	0.41	0.63	1.27	64.65%	49.60%	75.42%	0.41	0.64	1.28
DeepRoute	51.87%	28.35%	59.07%	1.42	5.98	1.96	50.88%	26.46%	57.31%	1.55	6.81	2.62	66.94%	48.67%	75.10%	0.41	0.61	1.27	66.91%	48.63%	75.08%	0.41	0.62	1.28
DeepRoute+	<u>52.03%</u>	<u>28.75%</u>	<u>59.80%</u>	<u>1.39</u>	<u>5.73</u>	<u>1.94</u>	<u>51.14%</u>	<u>26.87%</u>	<u>58.09%</u>	<u>1.52</u>	<u>6.54</u>	<u>2.60</u>	<u>67.19%</u>	<u>49.77%</u>	<u>75.76%</u>	<u>0.40</u>	<u>0.60</u>	<u>1.24</u>	<u>67.16%</u>	<u>49.73%</u>	<u>75.75%</u>	<u>0.40</u>	<u>0.61</u>	<u>1.24</u>
Graph2Route	52.53%	29.25%	61.22%	1.34	5.21	1.92	51.56%	27.28%	59.45%	1.46	6.02	2.58	67.97%	53.72%	78.68%	0.35	0.51	1.13	67.92%	53.68%	78.67%	0.36	0.52	1.13
Improvement	1.0%	1.7%	2.4%	3.6%	9.1%	1.0%	0.8%	1.5%	2.3%	3.9%	8.0%	0.8%	1.2%	7.9%	3.9%	12.5%	15.0%	8.9%	1.1%	7.9%	3.9%	10.0%	14.8%	8.9%

□ 各方法在Food-PD中的性能优于LogisticsP，在送餐中，未完成任务的平均长度更短

4 实验

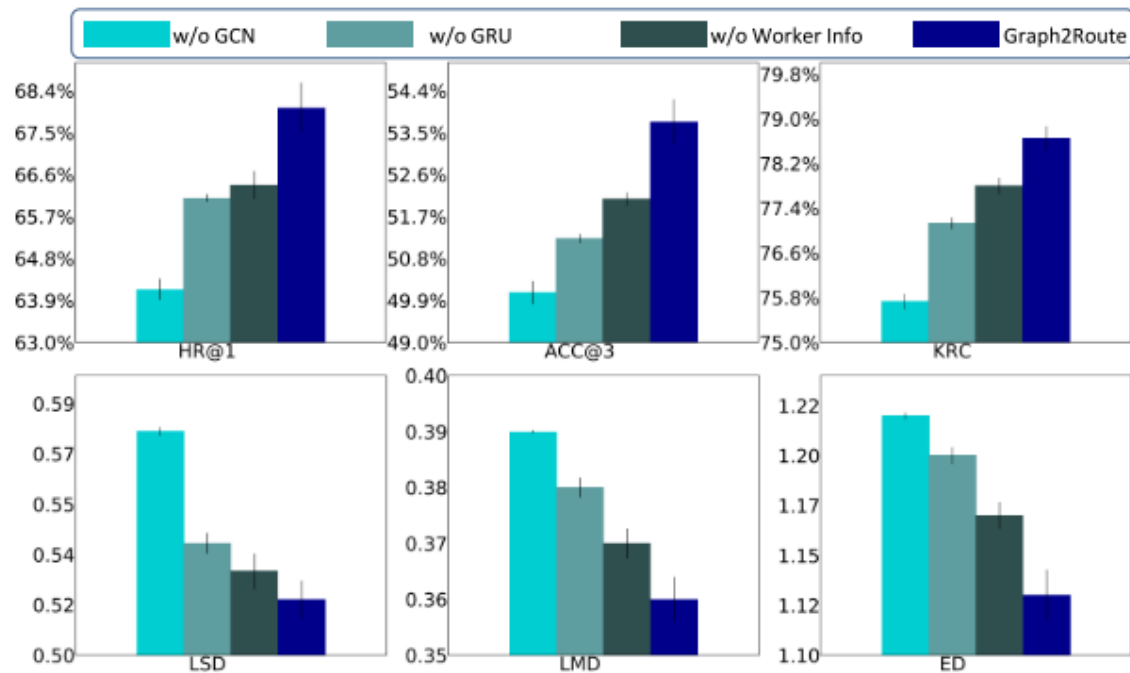


Figure 5: Component Analysis.

消融实验:

w/o GCN: 空间编码中不使用GCN

w/o GRU: 时间编码中不使用GRU

w/o Worker Info: 解码器中不加入工人个性化信息

实验表明时空编码、融合工人个性化信息带来的改进

4 实验

□ 超参数:

两个数据集中的最佳参数是

$$k = n - 1, d_h = 8, L = 2, d_w = 10, B = 64, lr = 1e - 3.$$

□ 实验表明时空编码、融合工人个性化信息带来的改进

Table 4: The hyper parameter setup of Graph2Route.

Hyper parameters	Description	Search space
k	k -nearest spatial-temporal neighbors	$[n-1, n-2]$
d_h	embedding dimension of nodes/edges	$[8, 16, 32]$
L	the number of GCN layers	$[1, 2, 3]$
d_w	embedding dimension of a worker's ID	$[10, 20]$
B	batch size	$[64, 128]$
lr	learning rate	$[1e-3, 5e-4]$

5 总结

□ 创新

- 1) 将原本基于序列的预测模型转换为包含时空信息的图模型
- 2) 在解码过程中充分利用图的结构和工人的个性化信息，输出更精确的服务路线；

□ 局限

循环解码机制：

循环体系结构可能会遇到效率问题。特别是在现实世界的情况下，例如物流场景，一个工作人员在同一时间可能承担着50个甚至更多的任务

缺乏对道路网络的建模：

只考虑到空间距离，忽视了作为空间信息的路网信息

□ 未来研究方向

更高效的解码机制：

从模型结构的角度，研究一种能够同时生成多个输出的非自回归解码器。

从模型压缩的角度，探索一种更轻量级的模型，以加快推理速度泛化到其他领域：社交网络轨迹、移动应用数据等。

道路网络的建模：在模型设计中对路网进行深度建模

参考综述：《A Survey on Service Route and Time Prediction in Instant Delivery: Taxonomy, Progress, and Prospects》

谢谢！

