Forecasting Fine Grained Drive Requests

--罗佩，张燕妮，韩梦乔，刘畅

一、问题描述

1、问题背景分析

如何根据历史和现在的订单数据，预测下一个时间段的订单数据，使得司机能选择更加合适的区域和时间段去接受用户请求，减少每一个订单的等待时间，提高订单数量和订单成功数？

场景举例：如右图

1. 已知：该区域晚上21:00-21:15的历史订单数平均20单/天，今天的订单数为25。

2. 预测今天21:15-21:30坂田区域会产生多少订单？

2. 问题描述：

1.数据准备：采集订单数据，将区域（如北京）分成适当的n x n(如4x4)的数量级的小区域集，如下图，然后在每个小区域集上又根据时间统计各个时间段的订单数据,最后可以得到根据时间段和区域统计的数据统计，完成初步的数据准备

2.模型建立，参数训练,得到订单数据规律，估算选定区域的下一个时间段的订单数据

已知各区域在各个时间段上的订单分布：

第1时间(t1)：

36	122	222	444
45	678	567	345
222	33	445	78
457	678	123	344

第2时间(t2)：

34	234	212	321
23	342	36	23
123	234	234	121
111	234	12	123

第3时间段(t3):

100	122	22	44
41	600	527	321
231	221	432	69
234	333	44	44

预计下一个时间段各区域上的订单数量：

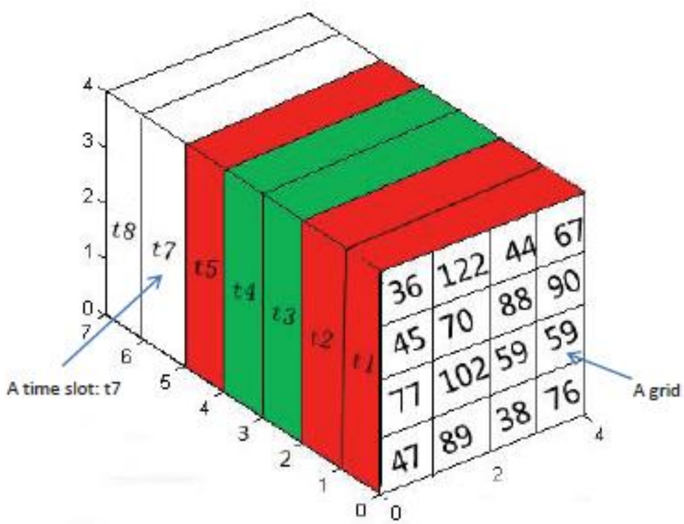
预计第4时间段(t4)：

?	?	?	?
?	?	?	?
?	?	?	?
?	?	?	?

表中的“?”是即是需
要我们估算的数据

各时间段上各区域的订单分布图：

其中标有颜色的代表已知的订单数据分布，未标颜色的是我们需要预计的未来时间段的订单分布



(a) Whole View

	t_{i-2}	t_{i-1}	t_i	t_{i+1}
g1	23	12	11	?
g2	14	13	12	?
g3	17	17	15	?
…	…	…	…	…
g16	12	18	15	?

(b) Time-Gird Matrix

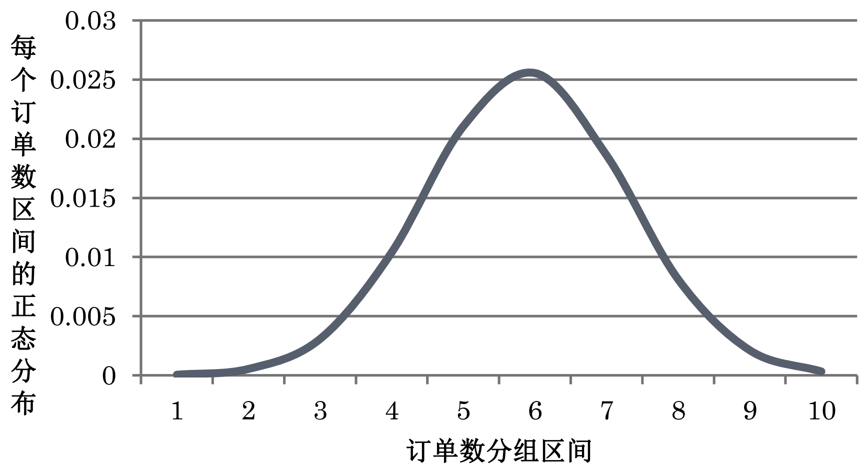
较多

较少

3. 解决问题的依据：

1.历史记录：

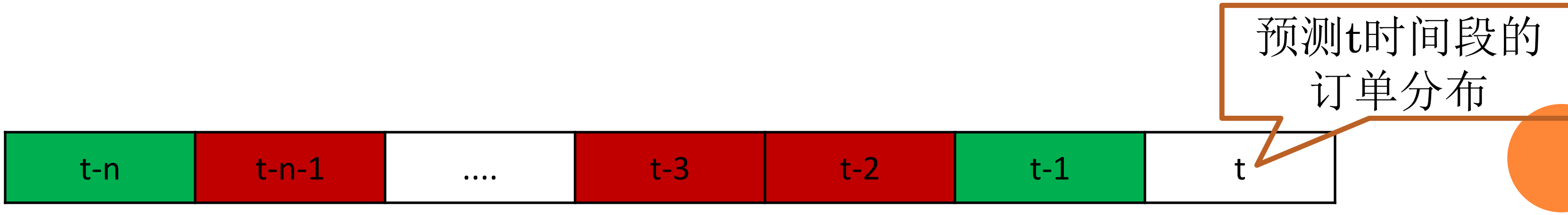
譬如：根据历史的订单数据可以得到订单数据在某个时间段上的正态分布



2.实时规律：时域的马尔可夫性

地域的马尔可夫性

时域的马尔可夫性：根据t-n,t-n-1,...t-3,t-2,t-1 各时间段预测t时间段的订单分布



区域的马尔科夫性：根据周边区域的历史订单数据，预测该区域未来几个时间段的订单分布

367	345	222
234	?	450
360	380	400

预测该区域在下一个时间段的订单分布

3.综合其他平台数据源

譬如，当Uber在某区域某时间段的订单数据量太少，不足以预测未来客户需求的时候，可以借助其他020平台的数据源，如，滴滴专车，神州专车等数据源，通过结合其他平台的数据，得出潜在的客户需求量。

Uber

45	0	55
66	?	5
32	10	55

+

滴滴专车

30	15	65
50	?	5
45	33	40

结合多平台数据预测该区域在下一个时 间段潜在的订单分布

	15	
116	?	10
	43	

二、数据预处理

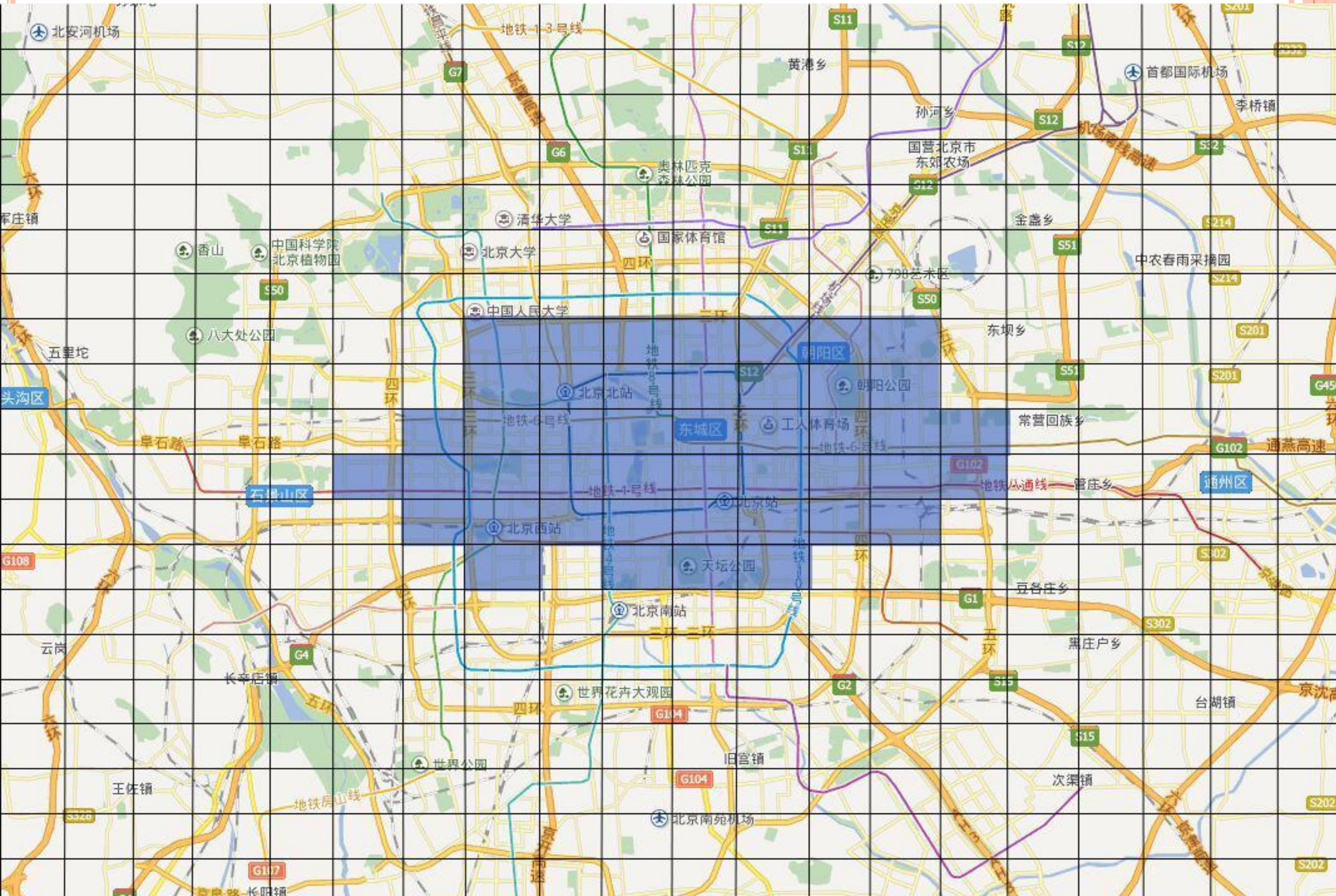
- 以15分钟为单位，把21:00-22:00 分成四个时间间隔，统计订单数据，得到每个block前这四个间隔的订单数
- 2. 由于很多block数据为0，所以刷选平均每个时间间隔得订单数>=5的block进行预测活动，刷选出来的block，如下页地图所示

avg_four @t_order (localhost_3306) - 表

id	time_lag_1	time_lag_2	time_lag_3	time_lag_4
231	19.5109	19.5761	19.2826	18.6087
232	28.1739	28.4239	28.7174	28.7391
233	11.3261	11.8043	12.6087	11.4348
234	6.1957	6.5217	6.7283	7.2283
235	0.9783	1.0870	0.9674	1.0109
236	3.2174	3.2500	3.9457	3.0217
237	1.2609	1.2174	1.3913	1.4022
238	0.6630	0.7174	0.7391	0.6087
239	0.7391	0.8152	0.7717	0.7065
240	0.1304	0.0978	0.1630	0.1739
241	0.1304	0.1522	0.1087	0.0435
242	0.0761	0.1739	0.1196	0.0543
243	0.0109	0.0109	0.0109	0.0000
244	0.6739	0.4348	0.4891	0.3478
245	2.3261	2.1739	2.0000	1.9348
246	2.6087	2.4130	2.2174	2.1957
247	17.2935	15.8370	14.6522	13.4891
248	16.3478	15.4783	14.9674	14.0326

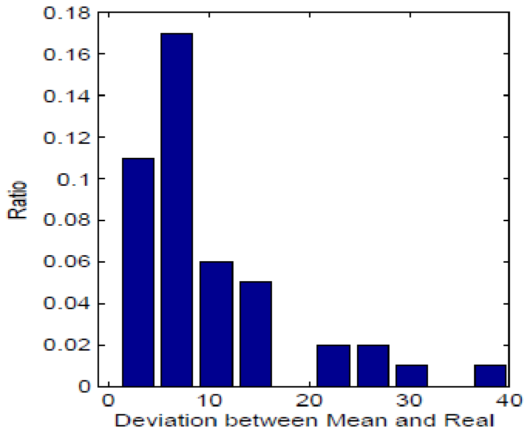
block_number_forecast @t_order (loc

block_number
167
169
170
171
186
187
188
189
190
191
192
193
205
206
207

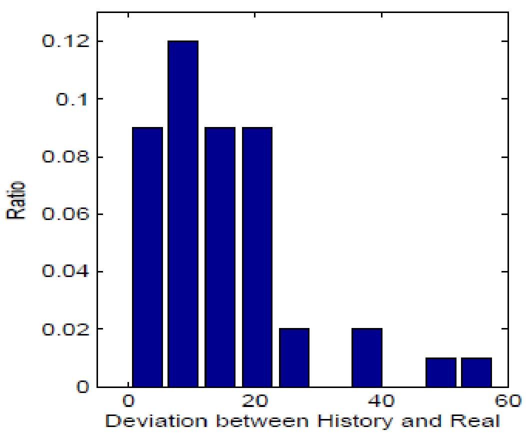


北安河机场
地铁13号线
黄港乡
首都国际机场
李桥镇
孙河乡
国营北京市
东郊农场
奥林匹克
森林公园
清华大学
国家体育馆
金盏乡
香山
中国科学院
北京植物园
北京大学
四环
中农春雨采摘园
798艺术区
中国人民大学
东坝乡
八大处公园
五里坨
朝阳区
北京北站
朝阳公园
头沟区
地铁6号线
东城区
工人体育场
常营回族乡
阜石路
通燕高速
地铁1号线
地铁八通线
管庄乡
石景山区
通州区
北京站
北京西站
天坛公园
豆各庄乡
北京南站
云岗
黑庄户乡
长辛店镇
世界花卉大观园
台湖镇
旧宫镇
世界公园
次渠镇
王佐镇
地铁房山线
北京南苑机场
长阳镇

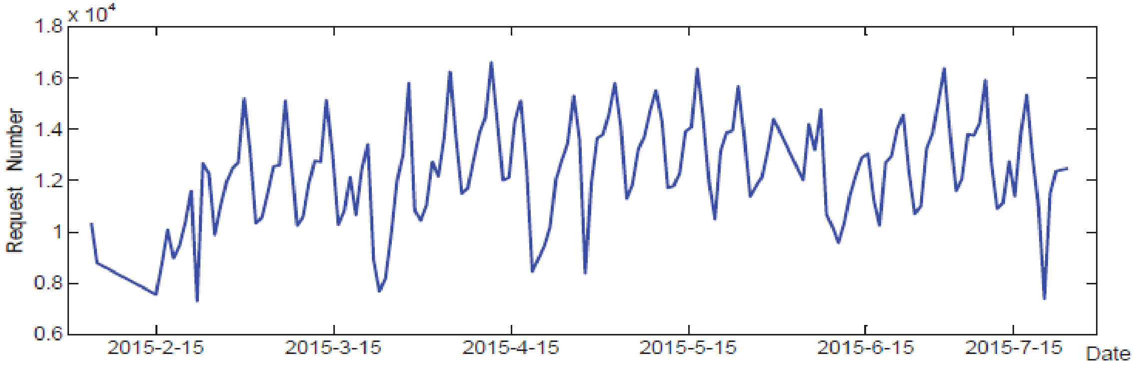
部分订单分析统计图



(a) Distribution of Mean Error



(b) Distribution of History Error



(c) Line Chart of Requests Data

三、问题解决方案（baselines）

方案一
- 利用前一时间段订单预测

方案二
- 利用历史数据的平均值预测

方案三
- 利用时域上的马尔可夫性预测

方案四
- 结合时域与地域上的马尔可夫性预测

预测结果评定指标及公式

- 1.单个样本的误差：

$$\mathrm{Error(average)} = |\text{预测值} - \text{实际值}|$$

- 2.所有样本的整体平均误差：

$$\mathrm{Error(average)} = \frac{\sum|\text{预测值}-\text{实际值}|}{\text{样本总数}}$$

- 3.单个样本的相对误差：

$$\mathrm{Error}(compare) = \frac{|\text{预测值}-\text{实际值}|}{\text{实际值}}$$

- 4.所有样本的整体平均相对误差：

$$\mathrm{Error}(compare) = \frac{\sum\frac{|\text{预测值}-\text{实际值}|}{\text{实际值}}}{\text{样本总数}}$$

方案一

- 基本思路：

利用当前时间段的实际订单数，作为未来一个时间段的订单预测数

- 预测误差：

1.所有样本的整体平均误差：5.283(个)

2.所有样本的整体平均相对误差：53.4%

方案二

- 基本思路：

  计算该时间段历史三个月上的订单平均数，作为预测订单数

- 预测误差：

  1.所有样本的整体平均误差：4.55(个)

  2.所有样本的整体平均相对误差：57.6%

方案三

- 基本思路

利用时域上面的马可夫性，创建马尔可夫模型

- 设计方案

1、时域上HMM的模型参数定义为λ（π, A, B）

π:　初始状态概率{“红”，“绿”}

A：状态转移矩阵：$\begin{bmatrix} \text{"红红"} & \text{“红绿”} \\ \text{“绿红”} & \text{“绿绿”} \end{bmatrix}$

B：观测概率矩阵：$\begin{bmatrix} \text{“红状态下的平均值”} & \text{“红状态下标准差”} \\ \text{“绿状态下的平均值”} & \text{“绿状态的标准差”} \end{bmatrix}$

2、HMM模型各算法

（1）前向算法

初始：α_1（i）=$\pi_i b_i(o_1)$

迭代：α_{t+1}（i）=[$\sum_k \alpha_t$（k）a(k,i)] $b_j(0_{t+1})$

概率计算公式：$P(O|\lambda)=\sum_i \alpha_T(i)$

（2）后向算法

初始：β_T（i）=1

迭代：β_t（i）=$\sum_k$ a(k,i)bk($0_{t+1)}$ β_{t+1}（k）

概率计算公式：$P(O|\lambda) = \sum_i \pi_i b_i(o_1)$ β_1（i）

(3)预测算法(viterbi)

初始：$\delta(1, \mathrm{i}) = \pi \mathrm{ibi}(\mathrm{o}_1)$

$\psi(1, i) = 0$

迭代：$\delta(\mathrm{t}, \mathrm{i}) = \max_k[\delta(\mathrm{t}-1, \mathrm{k})\mathrm{a}(\mathrm{k}, \mathrm{i})] * \mathrm{b}_i(0\mathrm{t}+1)$

$\psi(t, i,) = \arg max_k[\delta(\mathrm{t}-1, \mathrm{k})\mathrm{a}(\mathrm{k}, \mathrm{i})]$

$P^* = \max(i)[\delta(\mathrm{t}, \mathrm{i})]$

$q^*_T = \mathrm{argmax}\ [\delta(\mathrm{t}, \mathrm{i})]$

$q^*_{t-1} = \psi(t, \mathrm{q} *_{\mathrm{t}}), t = T-1, T-2, \ldots \ldots 1$

（4）模型学习算法

用EM算法迭代求状态转移矩阵A

E-step: $P(O,\ \ i_t = qi,\ \ i_{t+1} = qj\ |\lambda)= \alpha_t(i,j)a(i,j)b_j(ot_{\ +\ 1)}\beta_{t+1}(i,j)$

$P(O,\ \ i_t = qi|\lambda)=\alpha_t(i,j)\beta_t(i,j)$

M-step $\mathrm{a(i,j)}=\frac{\sum_{t=1}^{T-1} P(O, i_t=qi, i_{t+1}=qj\ |\lambda)}{\sum_{t=1}^{T-1} P(O, i_t=qi|\lambda)}$

- 预测误差：

  1.训练样本的整体平均误差：3.42(个)

  2.训练样本的整体平均相对误差：41.2%

  3.测试样本的整体平均误差：3.21(个)

  4.测试样本的整体平均相对误差：52.7%

方案四

- 基本思路：

  结合时域和地域上面的马尔可夫性做预测

- 设计方案

  I、HMM的模型参数定义为λ（π，C, A, B）

  π: 初始状态概率{“红”，“绿”}

  C：单状态转移矩阵：$\begin{bmatrix} \text{"红红"} & \text{“红绿”} \\ \text{“绿红”} & \text{“绿绿”} \end{bmatrix}$

  A: 组合状态转移矩阵：$\begin{bmatrix} \text{“红红红”} & \text{“红红绿”} \\ \text{“红绿红”} & \text{“红绿绿”} \\ \text{“绿红红”} & \text{“绿红绿”} \\ \text{“绿绿红”} & \text{“绿绿绿”} \end{bmatrix}$

  B：观测概率矩阵：$\begin{bmatrix} \text{“红状态下的平均值”} & \text{“红状态下标准差”} \\ \text{“绿状态下的平均值”} & \text{“绿状态的标准差”} \end{bmatrix}$

2、HMM模型各算法

（1）前向算法

初始：α_1（i,j）=$\pi_i b_i(o_1)c(i,j)b_j(0_2)$

迭代：α_{t-1}（i,j）=[$\sum_k \alpha_{t-2}$（i,j）]$a(k,i,j)b_j(0_t)$

概率计算公式：$P(O|\lambda)=\sum_i \sum_j \alpha_{T-1}$（$i,j$）

(2)后向算法

初始：β_{T-1}（i,j）=1

迭代：β_{t-1}（i,j）=$\sum_k a(i,j,k)bk(0_{t+1)}\beta_t$（$j,k$）

概率计算公式：$P(O|\lambda) = \sum_i \sum_j \pi_i b_i(o_1)c(i,j)b_j(0_2)\ \beta_1$（$i,j$）

（3)预测算法（viterbi）

初始：$\delta(1,i,j)=\pi_i b_i(o_1)c(i,j)b_j(0_2)$

$\psi(1,i,j)=0$

迭代：$\delta(t,i,j)=\max_k[\delta(t-1,k,i)a(k,i,j)]*b_j(0_{t+1})$

$\psi(t,i,j)=\arg max_k[\delta(t-1,k,i)a(k,i,j)]$

$P^*=\max_{(i,j)[}\delta(t-1,i,j)]$

q^*_{T-1}， q^*_T=argmax $[\delta(t-1,i,j)]$

$q^*_{t-1}=\psi(t,q*_t, q*_{t+1}), t=T-1, T-2, \ldots\ldots 1$

（4）模型学习算法

1.将以该小block为中心的9个小的block组成一个大block，学习大block上面的HMM模型，并标记每一个大样本的状态

（注：大样本即为组成大block的每个小block的样本之和）

2.对该小block的每个样本初始化一个状态

3.根据大小block的状态标记，统计得到A

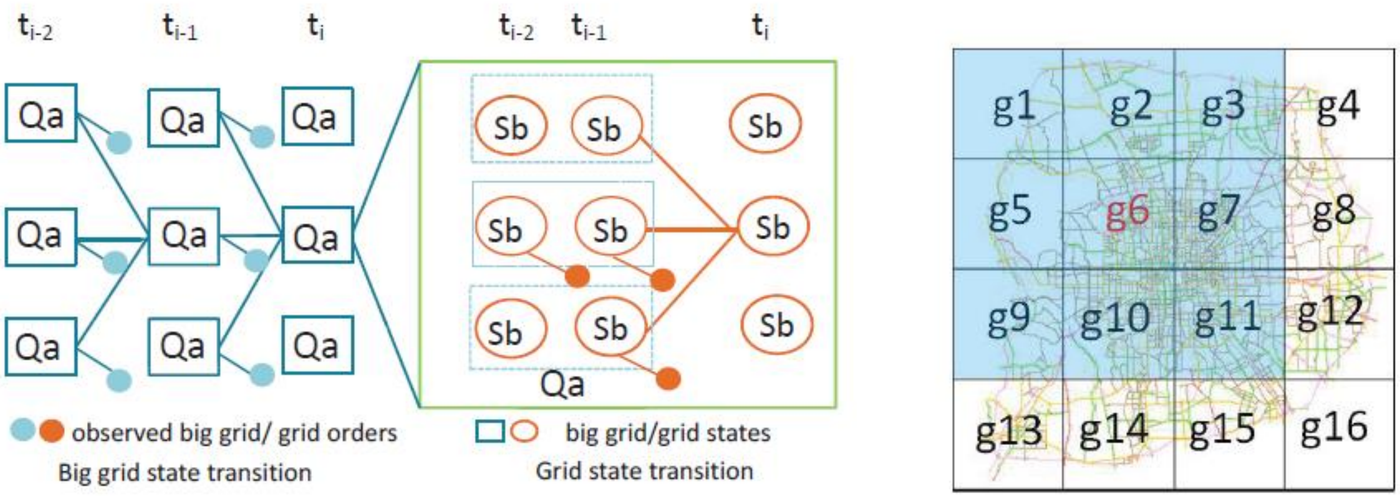
A:　组合状态转移矩阵：

$$\begin{bmatrix} \text{“红红红”} & \text{“红红绿”} \\ \text{“红绿红”} & \text{“红绿绿”} \\ \text{“绿红红”} & \text{“绿红绿”} \\ \text{“绿绿红”} & \text{“绿绿绿”} \end{bmatrix}$$

大block的t-1时刻的状态

小block的t时刻的状态

小block的t-1时刻的状态



(a) The Hierarchical Hidden Markov Model (b) The Big grid definition

- 预测误差：

  1.训练样本的整体平均误差：2.68(个)

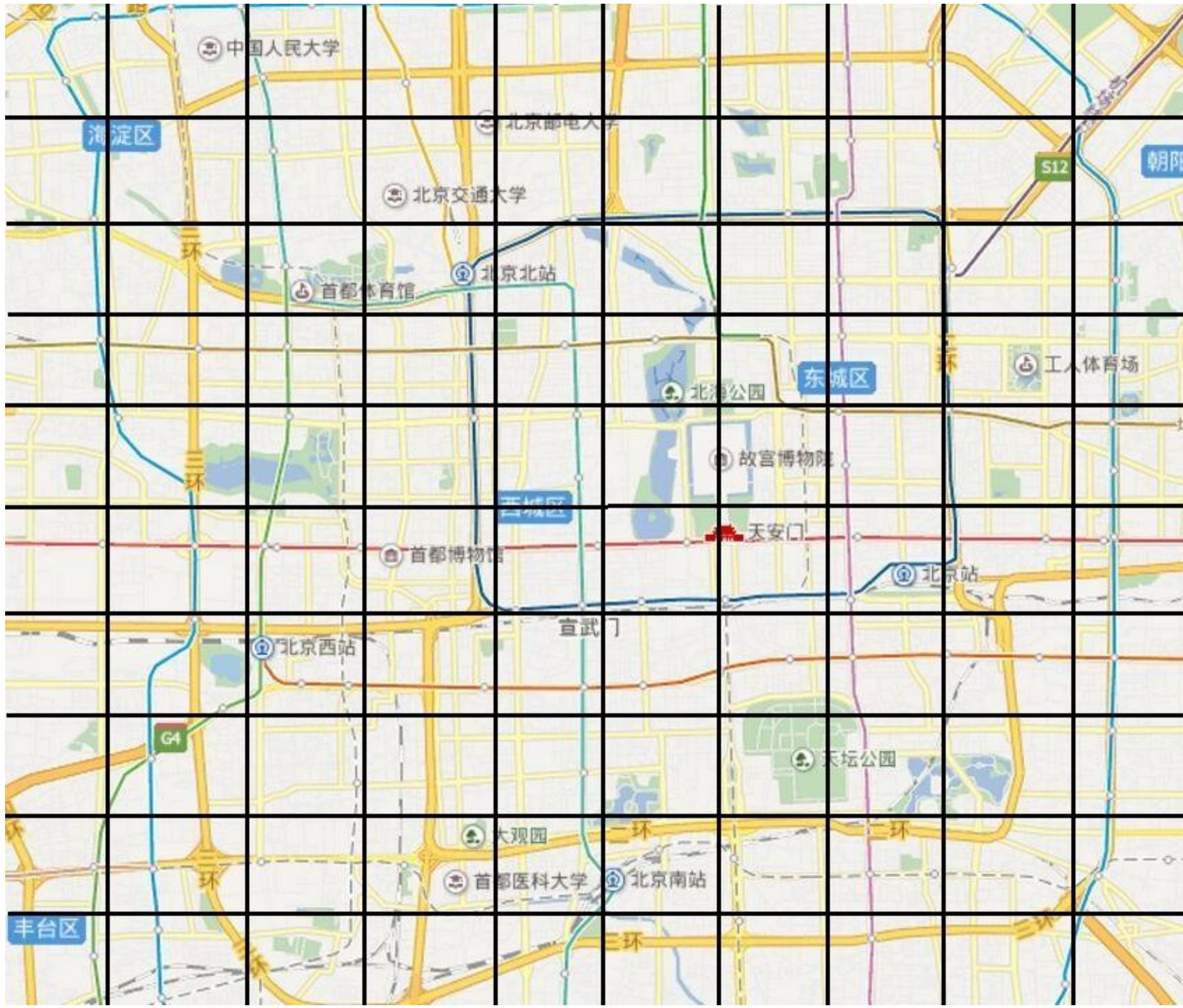
  2.训练样本的整体平均相对误差：32.5%

  3.测试样本的整体平均误差：2.8(个)

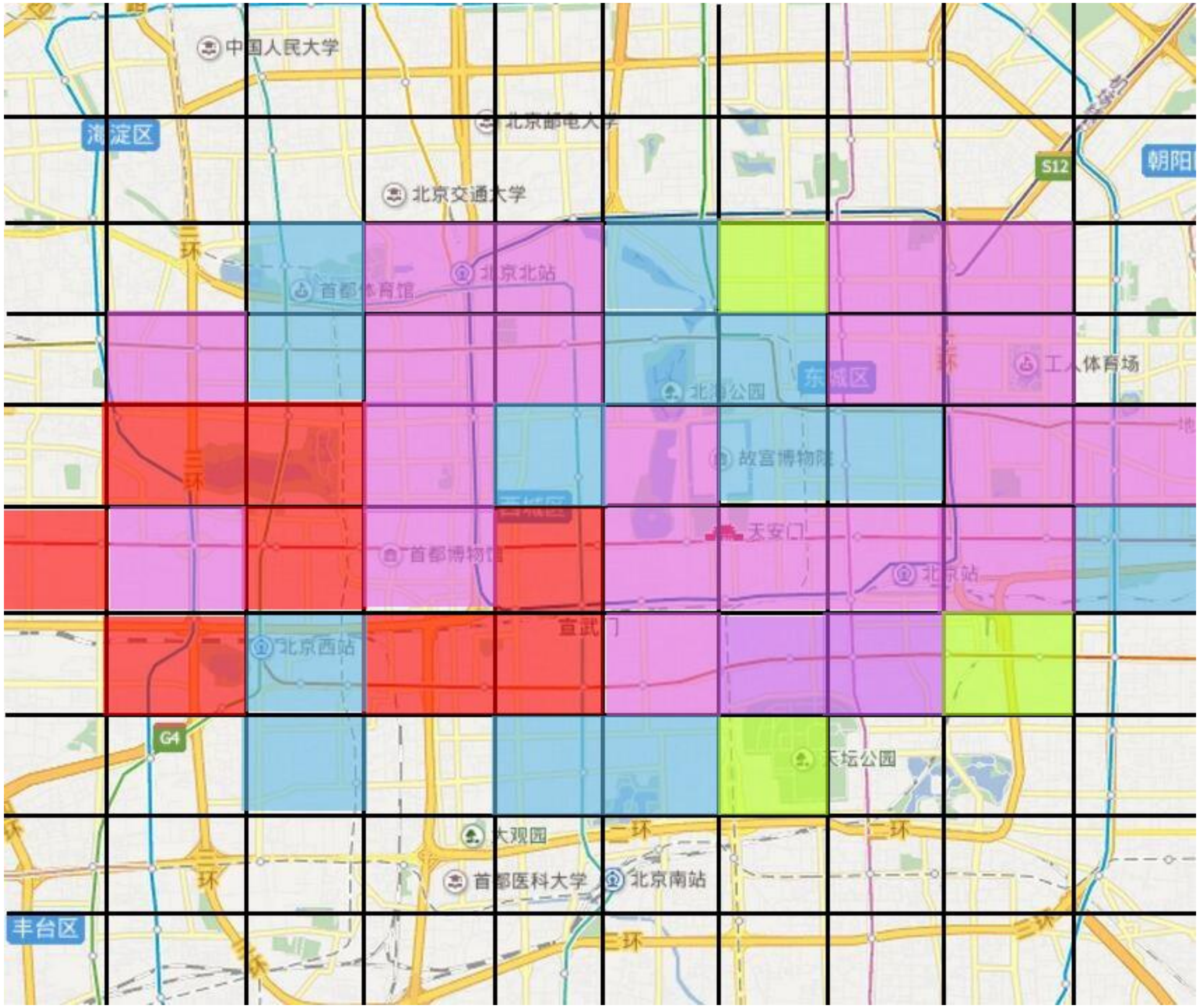
  4.测试样本的整体平均相对误差：33.8%

四、分块情况图

平均订单>=5的block全部集中在四环以内，这些block的分块情况大致如右边：



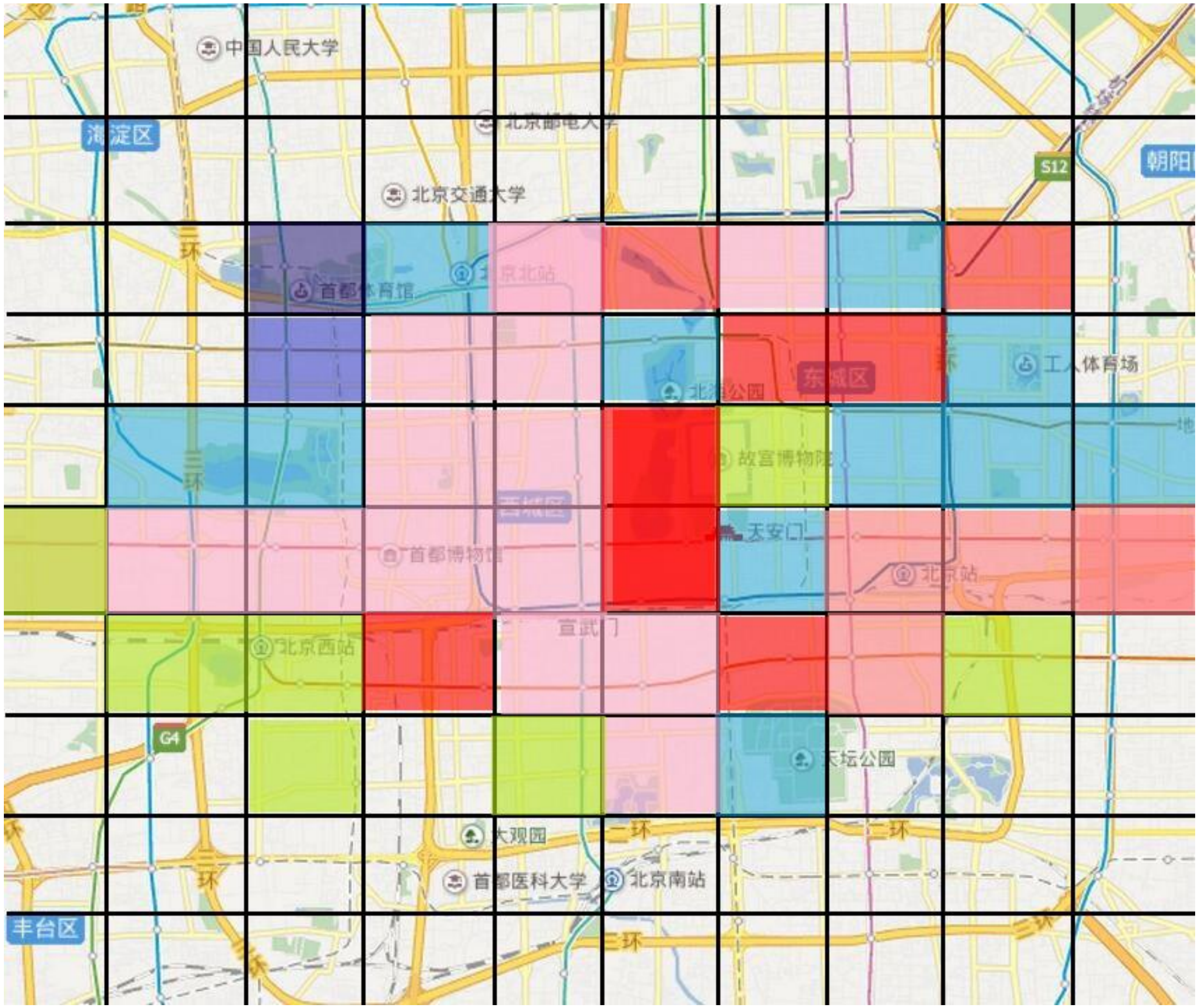
五、方案四上各分块的绝对误差分布图



绝对误差图例：

- <2
- 2~3
- 3~4
- >4

五、方案四上各分块的相对误差分布图



相对误差图例：

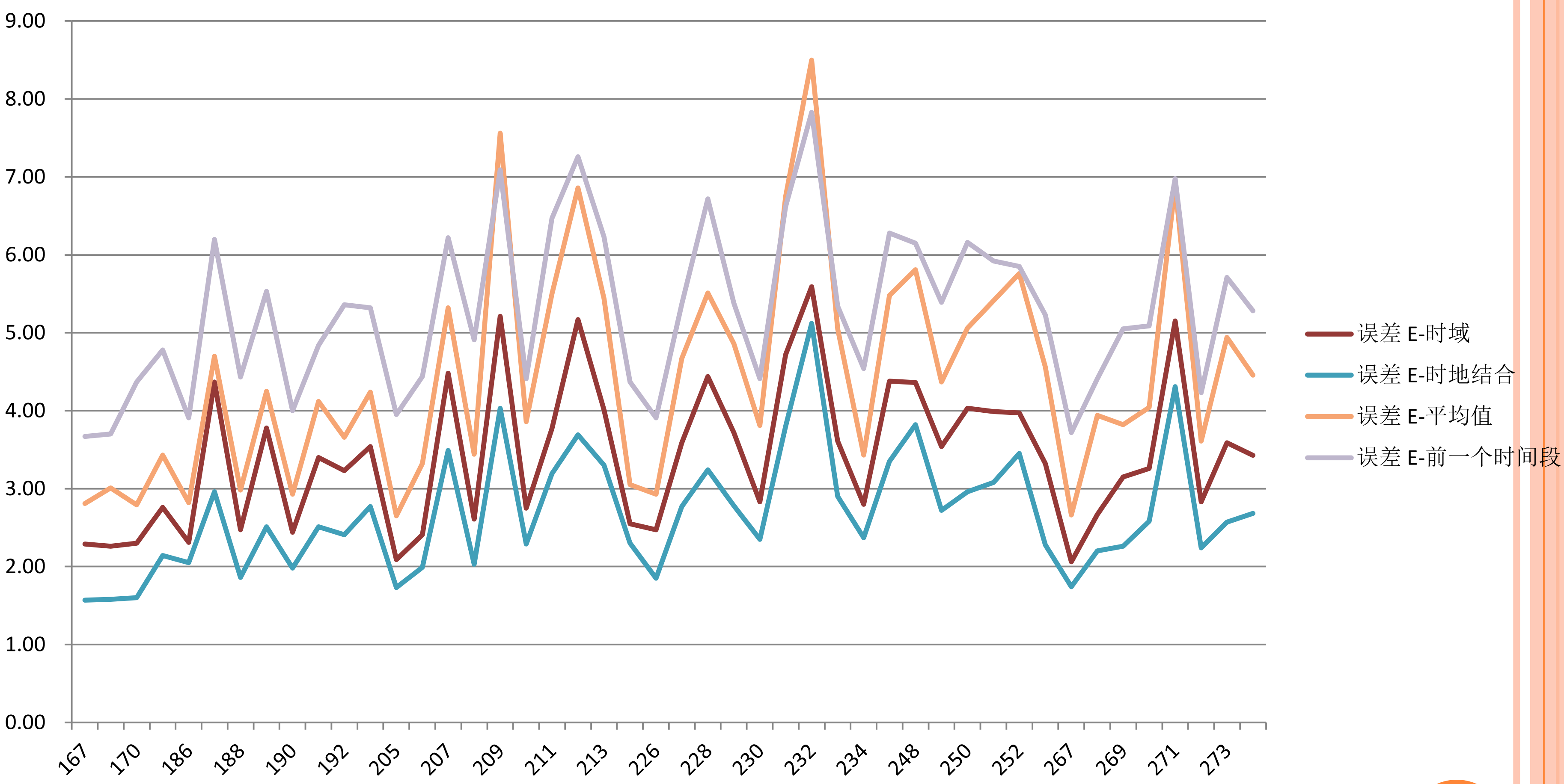
- <0.2
- 0.2~0.3
- 0.3~0.4
- 0.4~0.5
- >0.5

六、四个方案的误差对比分析表

以下是四个方案的整体样本上的绝对误差与相对误差统计结果，以及各个方案的比较

	历史值（方案一）	平均值（方案二）	时域（方案三）	时地结合（方案四）	方案二VS方案一	方案三VS方案二	方案四VS方案三
整体样本的绝对误差（个）	5.283	4.55	3.43	2.68	0.733	1.12	0.75
整体样本的相对误差（%）	53.40%	57.60%	42.30%	32.60%	-4.20%	15.30%	9.70%

七、每个block的四种方案的 预测误差折线图



八、每个block的四种方案的 预测相对误差折线图