最终报告

组员：张燕妮，韩梦乔，刘畅，罗佩

一、简介

目前基于 O2O 平台的驾驶服务平台越来越被人们使用，当人们在某些情况下不能驾驶，比如说聚会喝酒后，人们就可以通过手机上的 APP 发送一条代价需求，离得比较近的司机就会响应需求，并提供代驾服务。一般来说。用户请求地点覆盖整个城市地图，因此，高效的预测某个区域某一小段时间的订单数量成为了运营公司的一大重要且紧急的任务。通过预测的需求亮提前分配好司机到相应的区域，可以减少用户的等待时间，也能提高代驾服务的质量。

该项目主要想解决的问题是：高效且有效的预测每个区域的未来几个时间段的代驾需求数。为此，我们对数据进行了预处理，对北京城市进行 了区域划分，以 15 分钟为一个时间间隔，在每个区域上建立 一个隐马尔科夫模型，进行代驾需求预测。

二、问题陈述

2.1 问题场景

如图表 1 所示，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？

2.2 数据集

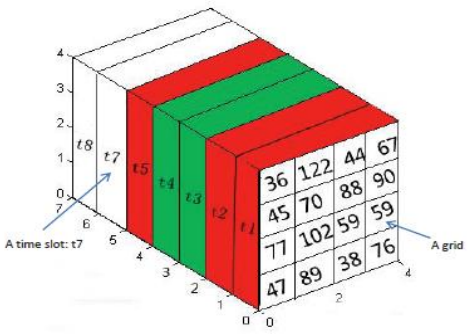
原始数据集每一条记录包括以下字段：id 号，下单时间，下单所在经度，下单所在纬度，订单成功与否标志字段。

我们把数据集按照一定的规则分解成区域-时间段的需求量矩阵，如图表 2 的子图（b）所示，整个数据集分解后就得到图表 2 的子图（a）所示的数据组织效果，任务就是预测每个 区域的未来时间段的需求数。

2.3 预期结果

预测结果输出形式的即为各区域的未来一个时间段的代驾需求量。

图表 1 应用场景



(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

图表 2 问题定义

2.4 评价指标

我们采用 MAE 和 MRSE 两个指标来衡量预测结果。计算公式如下：

其中y_i 是真实需求量，$\hat{y}_i$是预测值。

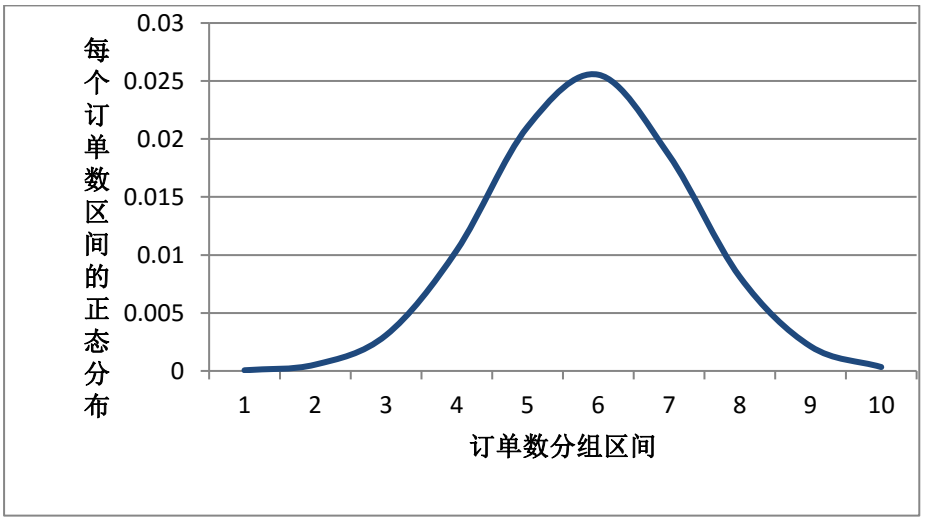
$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}, RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}$$

三、技术方案

3.1 解决问题的依据：

1.1 历史记录分布：

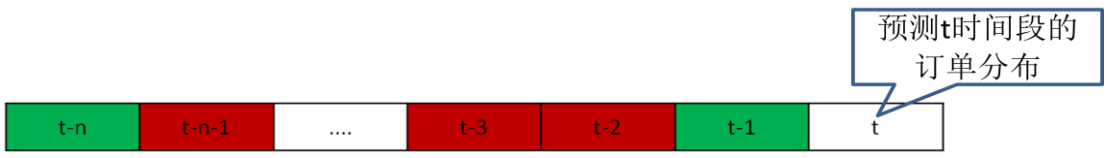
譬如：根据历史的订单数据可以得到订单数据在某个时间段上的正态分布



图表 3 某区域 订单正太分布示意图

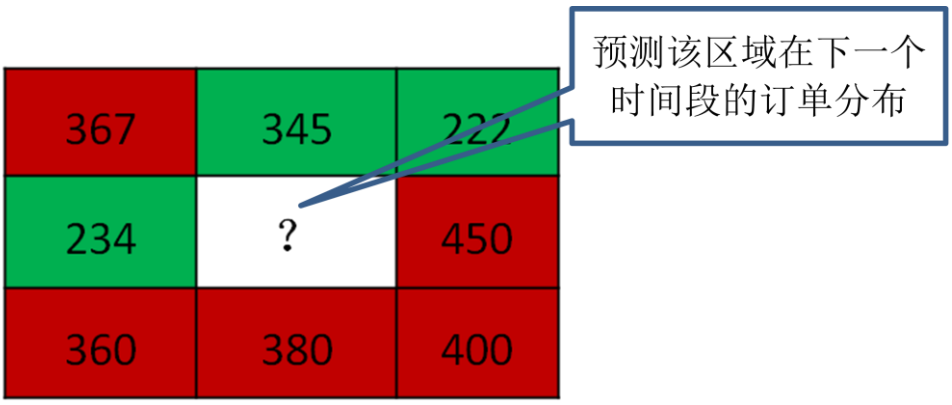
1.2 实时规律

时域的马尔可夫性 ： 假设下一个时间段的订单数与前面两个时间段的订单数量有关



图表 4 时域的马尔科夫性

地域的马尔可夫性 ：假设某个区域某个时间的订单数和其周边的区域订单数有关



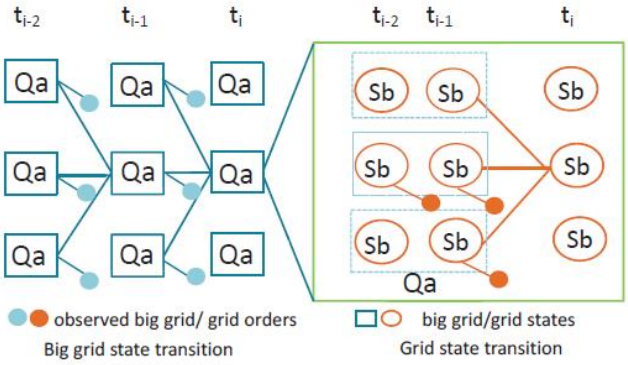
图表 5 地域的马尔科夫型

3.2 分层隐马尔科夫模型

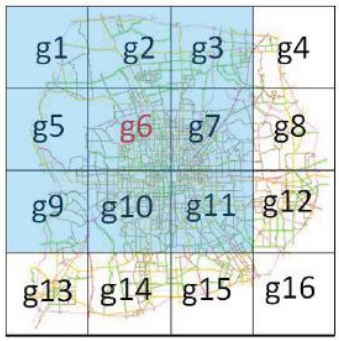
根据上面的解决问题依据，我们提出 了一个分层次隐马尔科夫模型。

将订单状态分成{“红” ，“绿”}两种，假设某个 区域下一个时间段的订单状态与该区域前两个时间段的订单状态以及它所在的大区域当前订单状态有关。

下图中子图（a）是整个分层隐马尔科夫模型的示意图。另外，我们约定一个某个区域所在的大区域如下子图（b）所示。



(a) The Hierarchical Hidden Markov Model



(b) The Big grid definition

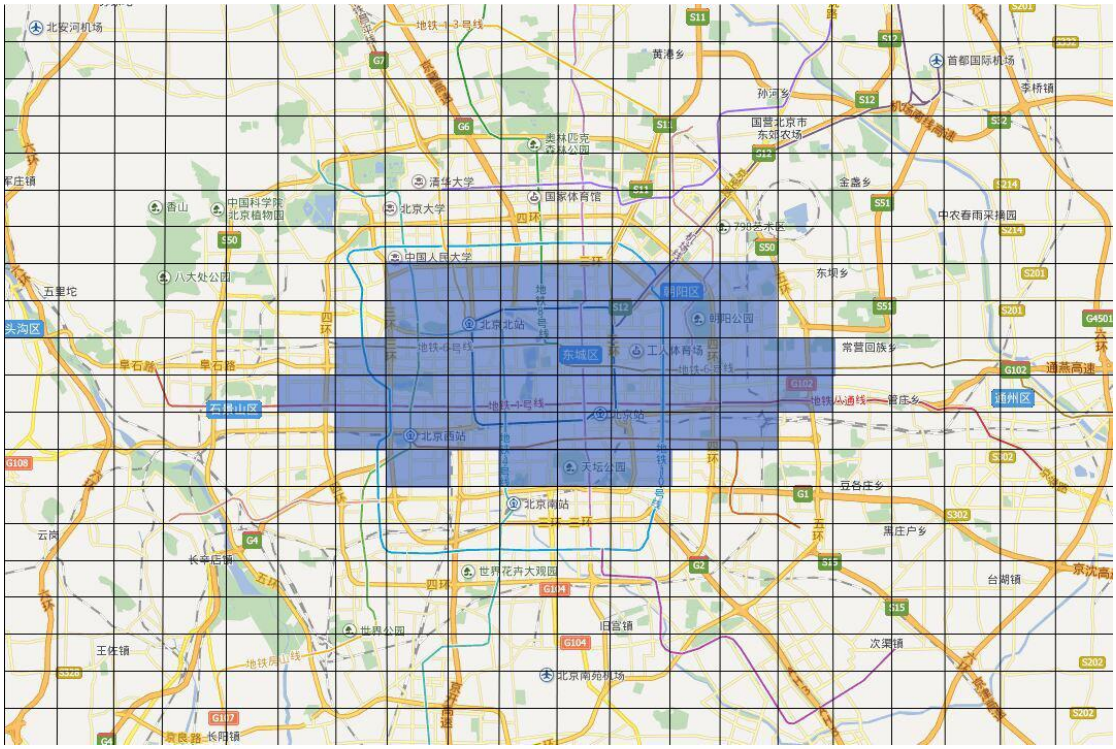
四、实现和实验结果

4.1 数据预处理

1.数据清洗：导入数据，并去除无用数据

— 北京六环以内的地理位置范围：116.09°E-116.73°E，39.60°N--40.19°N

— 分块情况为：20X20，其中蓝色涂层含义下文解释



图表 6 北京区域划分（20x20）

— 时间间隔：15min 为一个间隔，统计 21:00-23:00 三个小时，12 个时间间隔的数据

— （1）创建 t_order_201408-t_order_201507 ，共 12 个表，将各月份的 txt 数据导入相应的表中

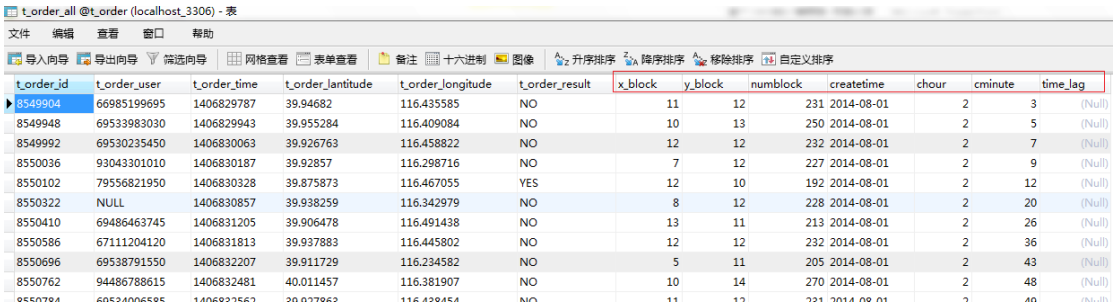
— （2）删除经纬度信息为 NULL 的数据

— （3）删除重复记录（数据中有很多各列都完全一样的数据）

— （4）删除经纬度不在 116.09°E-116.73°E，39.60°N--40.19°N 之间的数据

— **2.数据整合：得到每个 block 在每个时间间隔 t 上的订单数 的分布情况**

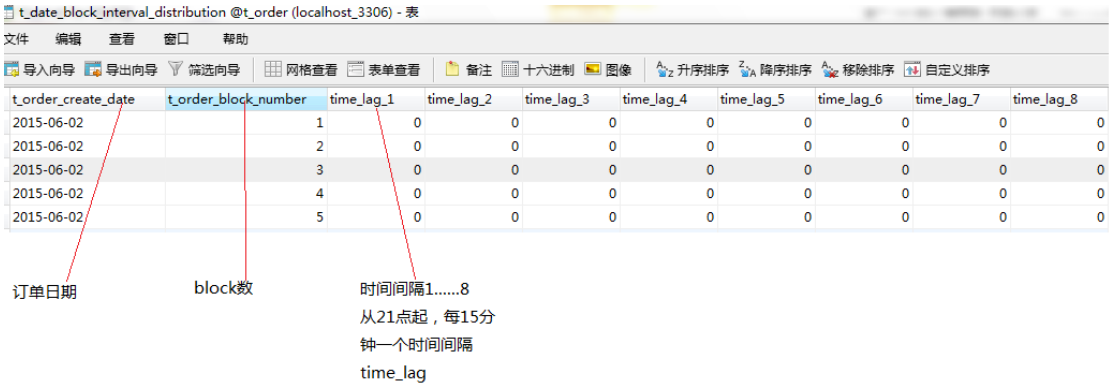
— （1）给 t_order_201408-t_order_201507 这 12 个表加上必要的统计列：block（订单所在的区域块）、createtime（订单创建的日期）、t(订单创建所属的时间间隔数)



t_order_id	t_order_user	t_order_time	t_order_lantitude	t_order_longitude	t_order_result	x_block	y_block	numblock	createtime	chour	cminute	time_lag
8549904	66985199695	1406829787	39.94682	116.435585	NO	11	12	231	2014-08-01	2	3	(Null)
8549948	69533983030	1406829943	39.955284	116.409084	NO	10	13	250	2014-08-01	2	5	(Null)
8549992	69530235450	1406830063	39.926763	116.458822	NO	12	12	232	2014-08-01	2	7	(Null)
8550036	93043301010	1406830187	39.92857	116.298716	NO	7	12	227	2014-08-01	2	9	(Null)
8550102	79556821950	1406830328	39.875873	116.467055	YES	12	10	192	2014-08-01	2	12	(Null)
8550322	NULL	1406830857	39.938259	116.342979	NO	8	12	228	2014-08-01	2	20	(Null)
8550410	69486463745	1406831205	39.906478	116.491438	NO	13	11	213	2014-08-01	2	26	(Null)
8550586	67111204120	1406831813	39.937883	116.445802	NO	12	12	232	2014-08-01	2	36	(Null)
8550696	69538791550	1406832207	39.911729	116.234582	NO	5	11	205	2014-08-01	2	43	(Null)
8550762	94486788615	1406832481	40.011457	116.381907	NO	10	14	270	2014-08-01	2	48	(Null)

—

— （2）统计每天的订单分布情况到表 t_date_block_interval_distribution 中



t_order_create_date	t_order_block_number	time_lag_1	time_lag_2	time_lag_3	time_lag_4	time_lag_5	time_lag_6	time_lag_7	time_lag_8
2015-06-02	1	0	0	0	0	0	0	0	0
2015-06-02	2	0	0	0	0	0	0	0	0
2015-06-02	3	0	0	0	0	0	0	0	0
2015-06-02	4	0	0	0	0	0	0	0	0
2015-06-02	5	0	0	0	0	0	0	0	0

订单日期

block数

时间间隔1.....8
从21点起，每15分
钟一个时间间隔
time_lag

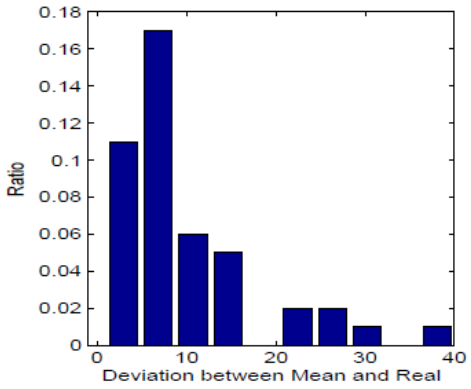
—

（4）创建表 avg_four，得到每个 block 在 2015 年 5 月后的前四个时间间隔，每天每个时间间隔的平均订单数

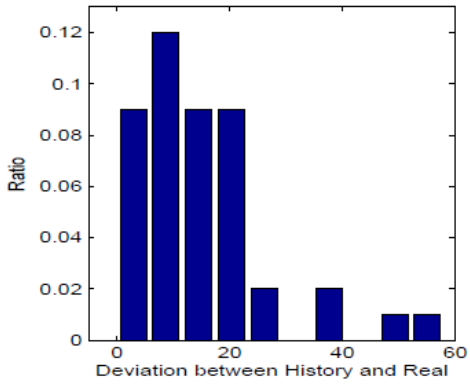
（5）创建 block_number_forecast，得到每个时间间隔订单数>=5 的 block number

结果发现，平均订单数大于等于 5 的 block 基本上集中在三环内，这些 block 即为图表 6 中的蓝色涂层区域。

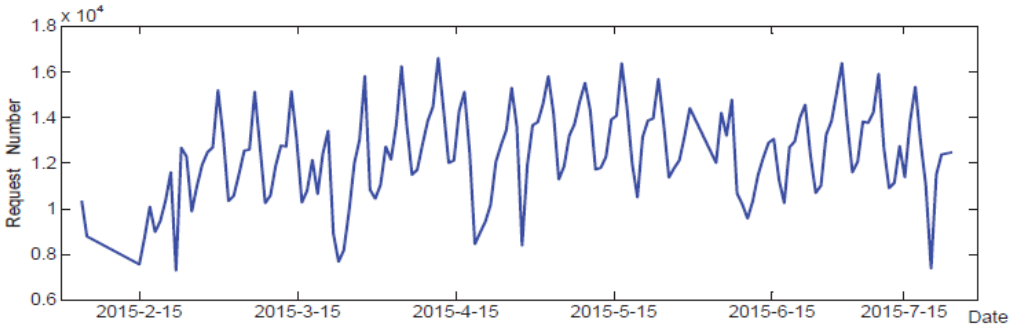
3.数据可视化，查看规律



(a) Distribution of Mean Error



(b) Distribution of History Error



(c) Line Chart of Requests Data

图表 7 原始数据可视化分析

图表 2 中， 子图（a）是真实订单数和平均订单数的偏差分布图，子图（b）是真实订单数和上一个时间间隔上的订单数数的偏差分布图。从这两个偏差图我们看到，偏差大多在 0-20 个订单数，这个误差是比较大的，所以直接使用平均值或者历史订单值作为预测值的话，并不能给实际应用带来成效。子图（c）是以天为单位统计订单数得到的订单分布图，可以看到，订单数并不稳定，所以用简单的平均订单数和历史订单数直接做预测会有较大的误差。

把平均订单数的数量分成三个等级，统计结果如下：

表格 1 区域订单数等级划分与比例

	Level 1	Level 2	Level 3
Range of Requests(N)	5 ≤N < 15	15 ≤ N < 25	N≥25
Percentages	60%	31%	9%

4.2 Baselines

为了验证我们提出的模型的实验结果，我们选择了以下几个 Baselines：

（1） 直接用该区域该时间段的平均订单数作为预测结果

（2） 用该区域当前时间段的订单数直接作为下一个时间段的预测数

（3） 用 KNN 模型来预测

（4） 用 NN（神经网络）来预测回归

（5） 简单的时域上的一阶隐马尔科夫模型预测

（6） 时域上的二阶隐马尔科夫模型预测

4.3 实验结果分析

1. 整体结果分析

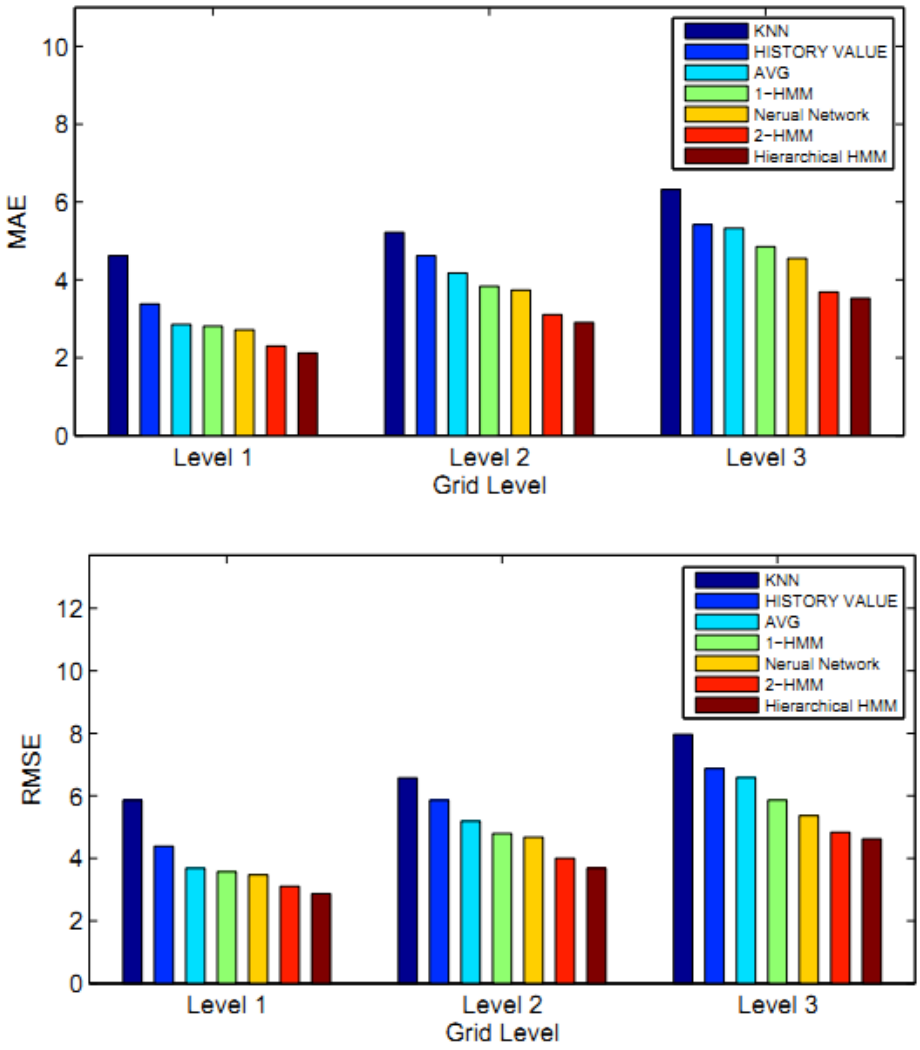
发现我们提出的模型效果最好：

表格 2 整体性能测试结果

Methods	MEA	RMSE
KNN	5.15(45.37%)	7.13(46.49%)
HISTORY VALUE	4.28(34.23%)	5.564(31.42%)
AVG	3.784(25.61%)	5.069(24.72%)
1-HMM	3.642(22.71%)	4.642(17.79%)
Neural Network	3.45(18.45%)	4.55(16.17%)
2-HMM	2.988(5.78%)	4.106(7.06%)
Hierarchical HMM	2.815	3.816

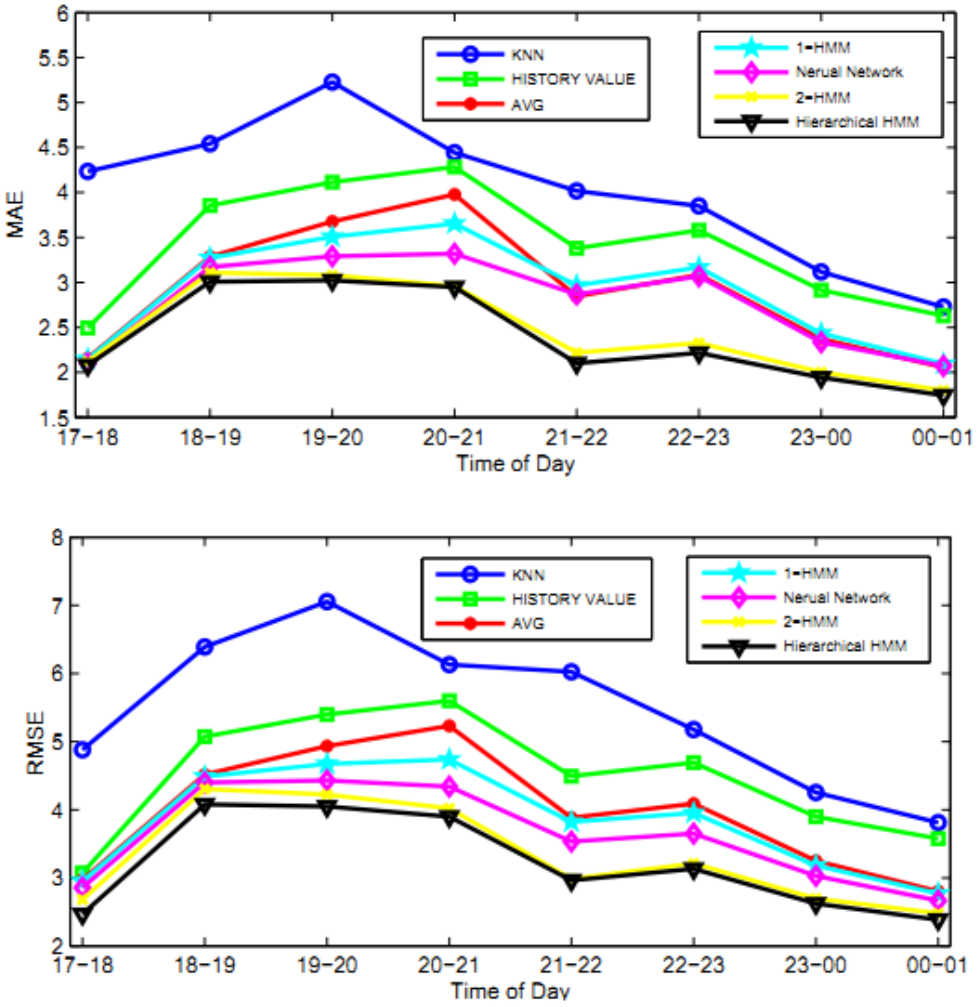
2.其他分析结果：

（1）每一个 level 上七种方案的误差结果，在每一个 level 上，我们的模型也体现出了很好的优势：



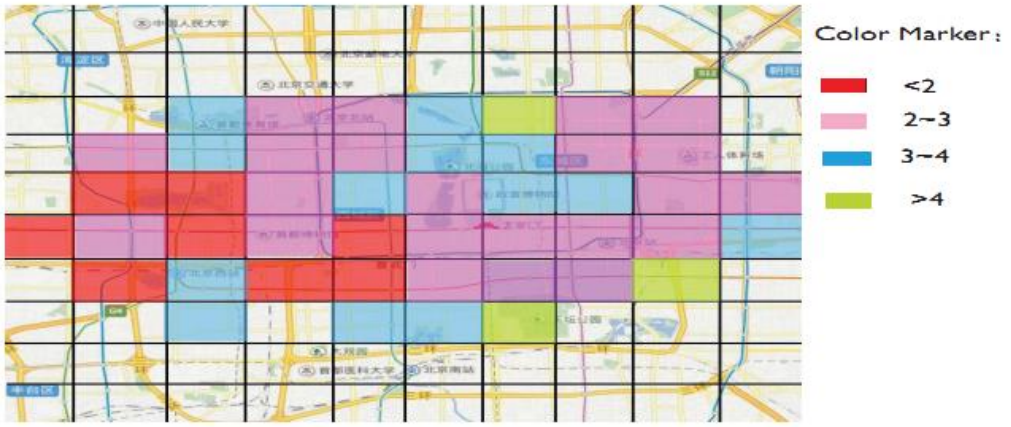
图表 8 不同 level 的 性能结果

（2）把每一天按照 15 分钟一个间隔分成 8 组，在时域上的表现性能，发现下班时间的预测误差最小，这也符合我们的需求，即尽可能的保证大部分关键时间点（如上下班）的预测准确性：



图表 9 同一天中不同时间段的性能结果

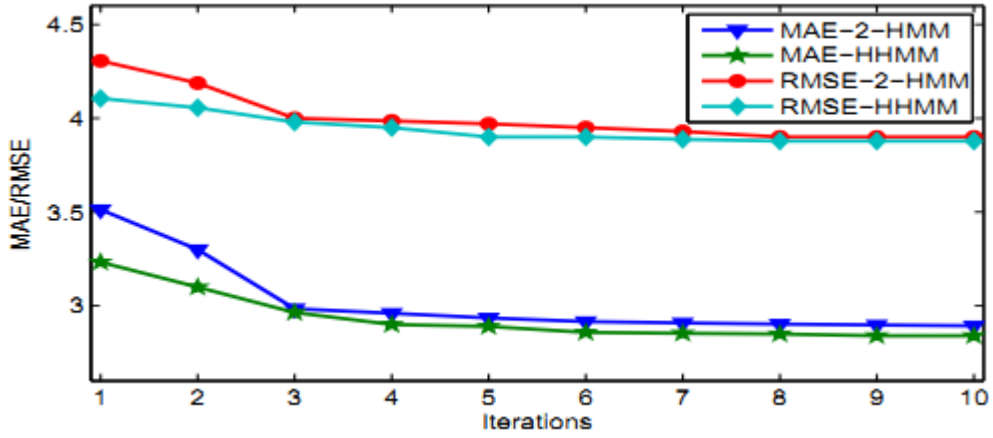
（3） 每个区域的整体误差结果分布，发现越中心的地区，即平时需求数越多的地区，预测数和实际数的偏差最小。



图表 10 预测数与真实数之间的偏差分布

4.4 模型收敛性分析

从下图可以发现平均迭代 8 次左右模型就已经收敛，说明该模型是可行的。



图表 11 模型收敛性

五、结论

从实验结果看，我们的模型比简单的用平均数或者历史订单数直接作为预测值的准确率提高了 30%左右。所以我们的模型是有效的，又根据其收敛性分析，证明了该模型的可行性。

但是依然存在很多的不足：

比如：该模型如何在线更新，是否可以利用其他驾驶服务平台的数据作为补充以提高预测准确度？如何衡量那些平均每个时间段的订单数小于 5 的区域的未来订单趋势？

这些都将在我们未来的工作中去考虑。