数据挖掘项目

————说明文档

1452822 洪嘉勇

1454093 夏陈

2017 年 6 月 24 日

目录

1 问题重述

1.1 第一问

1.1.1 a

问题一的 a 问要求从所给 2g 和 4g 数据中挖掘出所有的主基站，将所有的 MR 记录按照主基站位置分组。组内坐标均为主基站的相对坐标。对每个分组建立对应的以主基站作为相对位置标签的随机森林回归模型。

1.1.2 b

问题一的 b 问要求制作训练集训练模型，通过随机森林模型预测测试数据的相对位置，然后计算还原为原始数据。计算定位误差并排序，重复 10 次并绘制图表。

1.1.3 c

问题一的 c 问要求对每个分组对应的随机森林模型预测分组对应的 20% 测试数据 MRi 的原始位置。同时，尝试利用其他 k-1 个分组对应的随机森林模型对分组的测试数据 MRi 进行定位。对每个分组进行上述步骤，比较 k-1 个 RFi 预测结果中和 RFi 预测结果最为接近的分组序号。

1.2 第二问

1.2.1 第一层 HMM

构建一个单层 HMM 模型，直接对测试集的经纬度位置进行预测。

1.2.2 第二层 HMM

将地图进行栅格化，使用 a 中的单层 HMM 预测 MR 数据对应的栅格，第二层 HMM 用于从栅格直接预测经纬度位置。

2 基本假设

1. 对于给定路段，有可能在下一个时刻，汽车仍然是在那个路段
2. 一辆车只能从最后出发的一个路段到下一个的开始（这个确保输出路段连续）

3. 一辆汽车不能在某一路段上以不合理的速度行驶

3 模型建立和分析

3.1 第一问

3.1.1 数据预处理

该问中的数据预处理基本上就是从数据集中提取出问题中要用到的数据，并洗掉重复的数据。从共参数据中我们一共得到了 38789 个基站，和 MR 记录数据合并去重之后我们一共获得了 75 个不同的基站。

3.1.2 特征选取

鉴于 UE_Rx_Tx_x,EcNo_x,RSCP_x 中很多数据都是-999，毫无意义，我们最后选取比较具有特征，数据又相对完整的 EcNo_1，RSCP_1，RTT_1，UE_Rx_Tx_1，RNCID_2，CellID_2 EcNo_2，RSCP_2，RTT_2，UE_Rx_Tx_2，RNCID_3，CellID_3，EcNo_3，RSCP_3 作为特征训练随机森林。

3.1.3 结果呈现

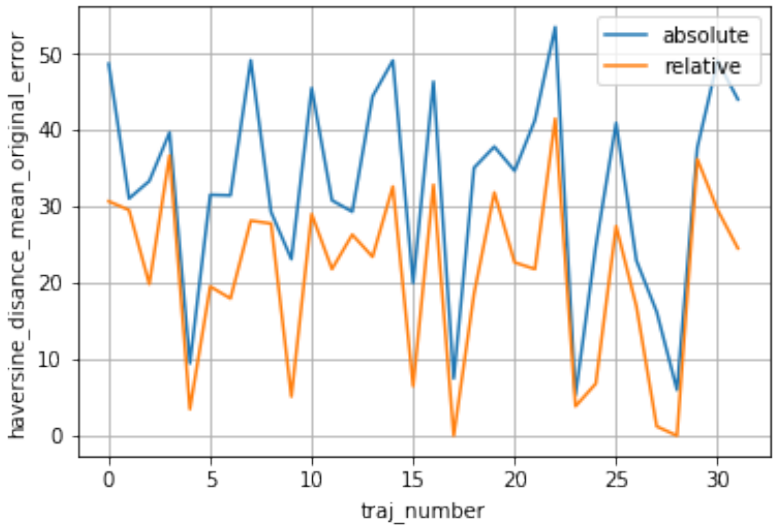

图 1: 平均误差和相对误差

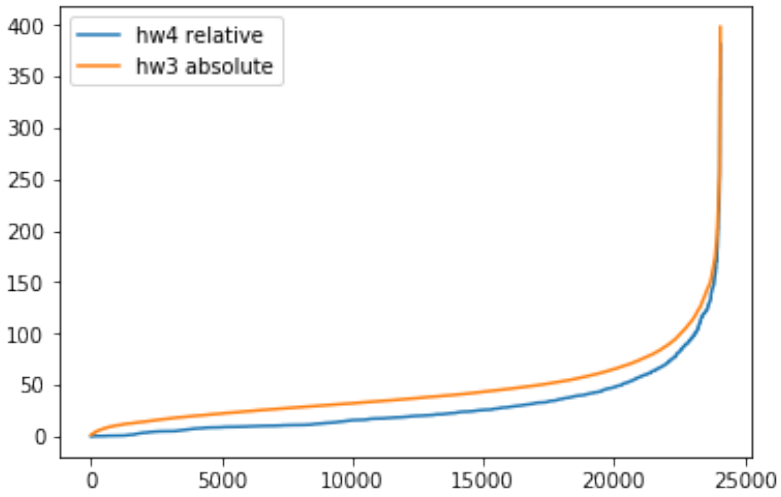

图 2: 作业 3 和 4 对比图

相对位置比绝对位置预测的距离误差在 haveresine 下稍微好一点
相对位置平均距离误差为 28.961245352168337m
绝对位置平均距离误差为 44.601022705329427

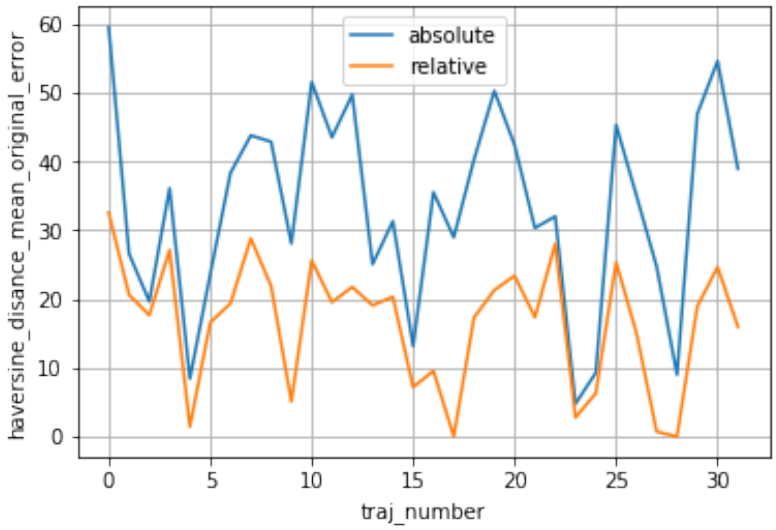

图 3: 中位误差

```
 [2, 5, 12, 16, 19, 21, 28, 32, 36, 39, 40, 41, 45, 46, 4
8, 54, 64],
 [12, 47],
 [2, 12, 16, 19, 21, 28, 32, 36, 39, 40, 41, 45, 46, 47,
 48, 54, 58, 64],
 [49, 53],
 [4, 22, 50, 51, 52, 56],
 [4, 13, 14, 22, 50, 51, 56, 59, 63],
 [1, 4, 10, 13, 14, 15, 17, 22, 34, 35, 37, 51, 52, 55, 5
6, 59, 63],
 [5, 18, 24, 27, 38, 53],
 [2, 5, 12, 16, 19, 21, 28, 32, 36, 39, 41, 45, 46, 48, 5
4, 64],
 [1, 10, 13, 14, 15, 17, 19, 20, 24, 32, 34, 35, 40, 45,
 52, 55, 59, 62, 63],
 [3, 6, 13, 14, 15, 22, 37, 50, 51, 52, 56, 59, 63],
 [7, 42, 57],
 [1, 2, 9, 12, 16, 17, 19, 28, 34, 37, 40, 45, 58, 60],
 [1, 3, 4, 6, 9, 13, 14, 15, 17, 19, 22, 34, 37, 51, 56,
```

图 4: 模型迁移之后的预测部分截图

我们通过模型迁移，通过其他训练好的随机森林模型来预测，得到预测结果，其中，1，13，14，17 号等基站对其他大部分基站的预测误差范围在 150m 内数量较多。

3.2 第二问

3.2.1 第一层 HMM

预处理

Test **数据轨迹拆分/去除** NaN 首先我们需要对所有的 Test 数据进行数据预处理，这边的预处理我们主要涉及以下两种:

1. 将数据中的轨迹拆分出来
2. 去除 NaN 的数据

窗口化 以每 1s 为窗口大小做窗口化，将原先要处理的 60000 条数据集减少到了 6000

观察链 每一条从 Test 数据中找出的项目

隐藏集合 Train 数据的栅格 ID

Emission **矩阵的计算**

1. 计算 RSSI 向量（6 维）
2. 将 Train 数据作为数据库并且以栅格 ID 分组（计算临时 0-1 矩阵提高匹配速度）
3. 对每一条 Test 数据进行以下公式计算（每一条 Test 数据运行速度为 0.3-0.5s)：

$$E_p(F_1, F_2) = M\lambda_{match} + (d_R^{max} - d_R(F_1, F_2))$$

M: 匹配的公参个数,d_R:RSSI 欧几里得距离

4. 将 Emission 矩阵按列做 normalize

意义

1）信号强度越强的越可能进入某一观测值的集合

Grid_ID	2.0	5.0	7.0	13.0	14.0	20.0	22.0	26.0	31.0	33.0	...	532.0	533.0	535.0	5
obeservable_63356	0.457831	0.465909	0.504870	0.468409	0.313953	0.551426	0.358039	0.487307	0.000000	0.0	...	0.508876	0.435294	0.497255	0
obeservable_63381	0.457831	0.487937	0.515034	0.488986	0.403963	0.577918	0.358039	0.490874	0.000000	0.0	...	0.508876	0.435294	0.481331	0
obeservable_63382	0.457831	0.482392	0.517442	0.456274	0.313953	0.542159	0.358039	0.487307	0.000000	0.0	...	0.508876	0.435294	0.486028	0
obeservable_63383	0.457831	0.482392	0.517442	0.456274	0.313953	0.542159	0.358039	0.487307	0.000000	0.0	...	0.508876	0.435294	0.486028	0
obeservable_63408	0.433735	0.474590	0.505814	0.451866	0.313953	0.528750	0.358039	0.475705	0.000000	0.0	...	0.482780	0.435294	0.472627	0
obeservable_63409	0.433735	0.474590	0.505814	0.451866	0.313953	0.528750	0.358039	0.475705	0.000000	0.0	...	0.482780	0.435294	0.472627	0
obeservable_63410	0.409639	0.429316	0.482558	0.481245	0.397141	0.564732	0.358039	0.481506	0.000000	0.0	...	0.456684	0.435294	0.488372	0
obeservable_63436	0.000000	0.893630	0.960924	0.964772	0.960924	0.973655	0.000000	0.982034	0.000000	0.0	...	1.000000	0.000000	0.972552	0
obeservable_63437	0.000000	0.893630	0.960924	0.964772	0.960924	0.973655	0.000000	0.982034	0.000000	0.0	...	1.000000	0.000000	0.972552	0
obeservable_63438	0.000000	0.871181	0.984464	0.962067	0.972837	0.972475	0.000000	0.982318	0.000000	0.0	...	0.974223	0.000000	0.972837	0

图 5: Emission 矩阵部分截图

Transition **矩阵的计算**

1. 取出每个基站的中心经纬度（Grid_center_x, Grid_center_y）
2. 每个基站的中心经纬度计算曼哈顿距离（以球面距离作为绝对值距离并且相加）
3. 将对角线置 1
4. 将矩阵每个元素取倒数

意义

1. 曼哈顿距离越远的栅格状态转换的概率越低
2. 栅格自身的转换概率为 1

效果呈现 以下为 HMM 第一层的效果呈现

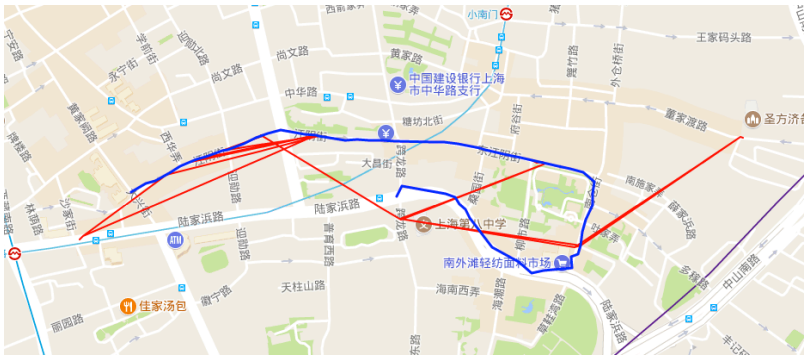

图 6: 第一层效果展现

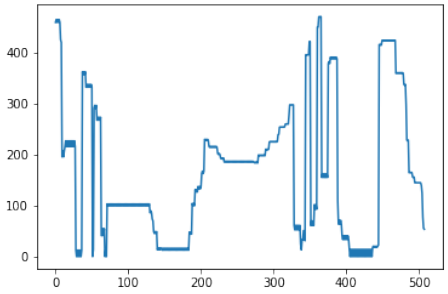

图 7: 距离误差图，平均误差为 168.72m，误差距离范围为 0-476m(haversine 距离)

3.2.2 **第二层** HMM

预处理 在第二层 HMM 的数据预处理中，我们主要包含以下两种操作:

平滑 我们把一层 HMM 的栅格作为输入，已 window 大小作为可调参数，计算每 window 窗口的中心点，其实就是一个“压缩”过程。

注：实验证明，没有做过滤波结果稍好，所以做了这部分工作，但是结果不好

滤波 使用简单的均值滤波，做一个“扩大”的过程。

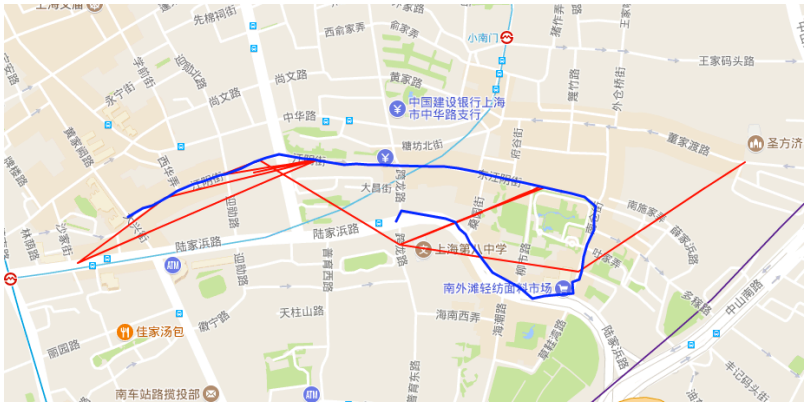

图 8: 平滑和滤波之后轨迹的投影

观察链 把预处理之后的栅格状态和静止状态形成二元组 {grid id, Hm}, Hm 为 0 或 1, 0 为静止, 1 为运动。

隐藏状态 这边的隐藏状态为 {road segment, Hm} 二元组, 该二元组中, road segment 为找到轨迹之后它经纬度的范围。之后去Open Street Map找对应的 osm 文件, 用 python 读入之后进行序列化, 得到路段信息和点信息。

图 9: Open Street Map 路段信息截图, 导出为 osm 文件

node

	lat	lon	timestamp	uid
id				
65563955	31.203207	121.514661	1390073985	527986
75354580	31.212207	121.498477	1328381422	23785
83320642	31.203301	121.514691	1390073986	527986
103736843	31.214264	121.501118	1403650228	13203
103736983	31.227474	121.495424	1346254163	66160
103737003	31.224116	121.498812	1194600773	15740
103737114	31.221157	121.500510	1194599312	15740
103737164	31.217092	121.502167	1325498238	360397
266072851	31.261622	121.488936	1325607263	527986

图 10: 所有路段点信息

	nodes	timestamp	uid	traj_long_la	HM
id					
11620182	[603837322, 603837471, 603837476, 603837483, 6...	1473917183	3836466	[[[31.2033062, 121.4852734], [31.2033859, 121....	1
11622557	[103736983, 1888382862, 677731611, 1888382802,...	1424184842	1190212	[[[31.2274745, 121.4954241], [31.2267359, 121....	0
39752563	[476562102, 856909369, 1328999596, 476562277, ...	1328381411	23785	[[[31.2134061, 121.4938402], [31.2128704, 121....	1
39752628	[618178152, 618178154, 618178156, 618178159, 6...	1403650229	13203	[[[31.2103796, 121.4964692], [31.2105421, 121....	0
47229047	[483103400, 477445166, 1576353686, 483103082, ...	1408447487	1487220	[[[31.2131932, 121.4774692], [31.213251, 121.4...	1
48686765	[476562329, 476562330, 476562331, 476562332, 4...	1441940845	1018672	[[[31.2134714, 121.4939716], [31.2133928, 121....	0
48686923	[618170145, 1618038112, 1618038108, 618171215,...	1488727796	5432507	[[[31.2111438, 121.4970333], [31.2111068, 121....	0
48687099	[476561733, 618178104, 1618038120, 618178107, ...	1403650819	13203	[[[31.212538, 121.4985733], [31.2120471, 121.4...	0
48687101	[83320642, 618178169, 4373253466, 4055294449, ...	1488727795	5432507	[[[31.2033014, 121.5146905], [31.2043965, 121....	0
48687102	[618178152, 618178212, 618178215, 618178218, 6...	1327049906	304705	[[[31.2103796, 121.4964692], [31.2105805, 121....	0
49073765	[622622486, 1404618558, 622622491, 622413798, ...	1465054217	527986	[[[31.2108616, 121.4870898], [31.2109488, 121....	0

图 11: 所有路段点信息

Emission **矩阵的计算**

1. 先计算每个点到 road segment 的距离 (D)
   1) 计算该点到 road segment[i] 每个信息点的距离
   2) 求距离的标准差 a
   3) 以 $\mu = 0, \sigma = a$ 为参数计算高斯距离向量
2. 将 Emission 矩阵按列做 normalize

高斯函数

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Transition **矩阵的计算**

1. 从路网信息 (osm 文件) 中找出恰好首尾相连的道路
2. 如果 road segment[i] 和 road segment[j] 按顺序首尾相连，我们定义 $P = 1 * \lambda_M$
3. 其他不相连的置为 0

3.2.3 分析讨论

实验中我们一共提供了 2 种比较方式，一种是平均距离误差，另一种是道路覆盖。

从平均距离误差来看，第一层 hmm 的平均距离误差约为 168.72，第二层 hmm 约为 192.3，比较起来第一层 hmm 的效果稍好。目前实验下来的原因是因为第一层保留了大部分信息，并

且信息大多都比较确定，比如栅格隐含状态的转换，发射矩阵的集合确定，所以在信息较多的情况下可以做到比较好的结果。但是第一层的投影结果不好，只有大致的轮廓比较符合真实轨迹。

从道路覆盖的角度来看，第二层覆盖的道路较多，但仍然有很多多余的道路或者过长的道路没有匹配上，比对第一层的投影，覆盖面积第二层 hmm 较多，所以第二层 hmm 在这个角度上来看较好。

综上所述，转换矩阵和发射矩阵的确定和参数选择至关重要，初始概率也是非常重要的因素，所以对该模型优化需要对调节参数进一步调节

4 Hmm2 效果呈现

4.1 未插值

以未插值的经纬度作为输入

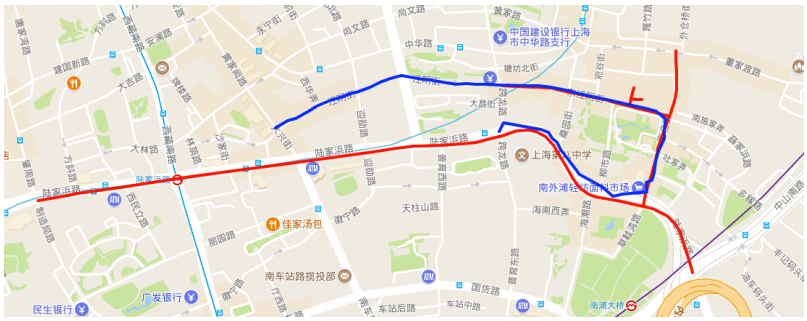

图 12: 1. 初始概率矩阵 $p_0 = [1, 1, 1..., 1]$ 2. 概率矩阵选择 $\varepsilon = 1/(\text{道路出度} + 1)$(来自 VTrack)

图 13: 1. 初始概率矩阵 $p_0 = average$ 2. 概率矩阵选择 $\varepsilon = 1/(\text{道路出度}+1)$(来自 VTrack)

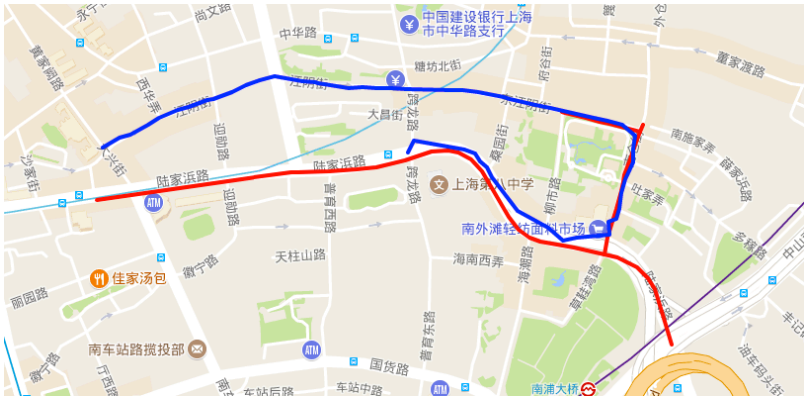

图 14: 1. 初始概率矩阵 $p_0 = dirichlet$函数 2. 概率矩阵选择 $\varepsilon = 1/(\text{道路出度}+1)$(来自 VTrack)

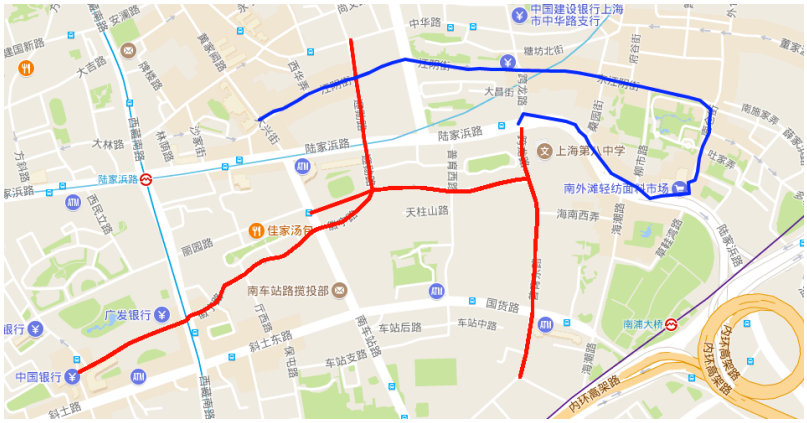

图 15: 1. 初始概率矩阵 $p_0 = average$ 2. 概率矩阵选择：只要道路有连接就置为 1

图 16: 汇总图

4.2 插值以后

以已插值的经纬度作为输入

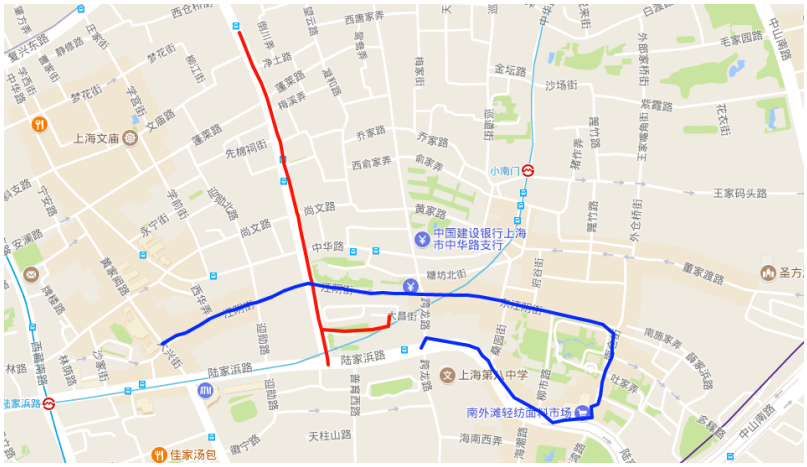

图 17: 1. 初始概率矩阵 $p_0 = [1, 1, 1..., 1]$ 2. 概率矩阵选择 $\varepsilon = 1/(\text{道路出度} + 1)$(来自 VTrack)

图 18: 1. 初始概率矩阵 $p_0 = average$ 2. 概率矩阵选择 $\varepsilon = 1/(\text{道路出度} + 1)$(来自 VTrack)

图 19: 1. 初始概率矩阵 $p_0 = average$ 2. 概率矩阵选择：只要道路有连接就置为 1

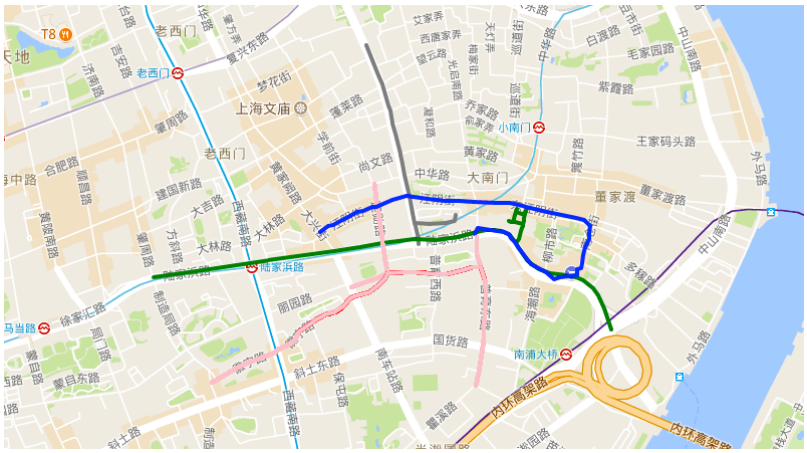

图 20: 汇总

5 总结和心得

经过这次作业的锤炼，我们对数据挖掘有了更深的认识。首先，数据挖掘并不是简单地套用论文里的公式进行计算，是必须有自己的思考在里面的。但它又不能说是一门数学，因为这里面又有很多的编程和经验性的东西。有些操作是目的性特别强的，比如数据集的清洗，但是有些操作完全是经验上的东西，并不知道为什么要这样做，但是这样做了效果就是会很好。数据挖掘需要不断地锤炼，不断地充实自己的方法库，才能让自己更好地掌握这门学科。