第一组论文研究进度报告

当前进度

1. 站点聚类(3.1)
    - [done]K-means聚类
    - [done]根据站点聚类之间的流量信息进行聚类
2. 利用一个multi-similarity-based inference model 来预测check-out proportion(3.3)
    - [done]分析时段特征值（Time Similarity/P5）
    - [doing]分析天气特征值（Weather similarity/P6）
    - 气温与风速特征（Temperature & wind speed similarity）
3. 采用同样的multi-similarity-basedinference model 预测inter-cluter transition.(3.4)
4. 利用最大似然函数去学习trip duration(3.5)。
5. 以及第4节的check-out inference和check-ininference。

T矩阵聚类的类与接口已经完成，有待调试。另，分析天气特征值已经由张明华在负责。进度未知。

数据集分析

已经实现了数据预处理的部分内容。当前的数据集分析接口可以支持K-means的使用，但是尚未完成对于交通信息聚类的接口内容。

K-means聚类

已经完成所有K-means聚类的聚类算法以及接口工作。

代码已经提交到GitHub上：https://github.com/AllLiving/DataMining

T矩阵聚类

T矩阵聚类即根据站点之间的车辆流量分布对K-means已经实现的聚类中心进行进一步的聚类，下一步的迭代操作是使用K-means在T矩阵聚类的聚类结果内继续划分。

笔者在原文的分类方式上进行了适当的简化。

原文的思路如下：

1. 使用T矩阵描述当前的路况信息
    - T矩阵有7行，每行对应着不同的时间段
    - 矩阵中的每一个元素表示到达下一处聚类点的可能性
2. 根据T矩阵的信息对K-means的聚类中心聚类

    具体的聚类操作原文并未提及。" After obtaining a set of t-matrices, {Ai}i=1 n cluster the stations into K2 clusters, ..., here K2 < K1 "。

T矩阵聚类的疑问

那么问题就出现了，虽然T矩阵中使用了时段这种因素去衡量各个聚类点之间的关联性。但是其实本质上使用的衡量标准依然是流量，流量为王，流量至上。此时使用时段信息，是希望对流量造成怎样的影响呢。

笔者的意思是无论造成怎样的影响，可解释性都很差，因为站点之间的流量总量是非常可以说明问题的，这并不是某一个时段的问题。此时考虑最极端的情况，节假日的车流量被考虑进当前的流量数据集中——数据集高达470000行，时间跨度非常广，即便是节日信息确实会影响流量信息，也无法影响整体的流量趋势。何况之后还有针对时间上得特征似然函数对流量信息进行分析。

修改后的T聚类

不再依赖于T矩阵，转而直接使用总体流量值对各个站点直接进行聚类。

1. 统计每一个站点对于其他站点的总车流量
2. 获取最大的车流量，并记录该流量的目的地所在的聚类中心A
3. 将具有同样聚类中心的聚类点聚类，获得新的聚类点B
4. 使用K-means对各个聚类进行重新聚类，聚类后的中心数量依然是K_1
5. 重复迭代

    修改之后的交通聚类算法的复杂度降低，同时可解释性更强。

当前还没有实现重复迭代的过程，一方面是PC机的运行效率有限，另一方面是PC机还要做其他的Project呜呜呜

分析时段特征值

GitHub地址：https://github.com/Aroya/Database_Project_Predict_Model

原文中针对时段特征值有如下考虑：

- 前提一： 考虑站点的基本单位是聚类
- 前提二：每天同一时间的车流量类似
- 前提三：随着时间的推移，越久远的数据参考性越差

因此在根据过去数据进行预测的时候需要遵循以下规则：

$$\lambda(t_1,t_2)=1(t_1,t_2)\times\rho^{\Delta h(t_1,t_2)}\times\rho^{\Delta d(t_1,t_2)}$$

其中t_1与t_2分别表示两个时间点（其实是需要知道日期信息以及时段信息的，原文中并没有考虑日期信息，但是不考虑日期信息是无法获知当前是否是工作日的）。

1. 使用一系列函数实现工作日与双休日的划分；
2. 实现对最近时间的流量信息施加更多的参考权值，即距离现在越靠近的数据越值得参考；

就是实现这两点。相关的代码已经编写完成，其中包含文件流以及对日期时间信息的转换

根据天气信息构造似然函数

使用天气因素进行预测需要考虑以下前提：

- 前提一：天气因素有四种
- 前提二：不同时段的天气是有关系的
- 前提三：不同天气出现的概率可以由频率预测

所以在考虑天气因素的时候，既然是以聚类为单位，那么就很难获得一个精确值。因此，任何一种模拟天气的算法可解释性都非常有限。原文中使用天气矩阵对已有的天气信息进行衡量：

	雪	雨	雾	晴
雪	1	a_1	a_2	a_3
雨		1	a_4	a_5
雾			1	a_6
晴				1

原文中表示通过一些未知的方法，得知矩阵中的元素满足关系：

$$
\begin{aligned}
a_1 > a_2 > a_3 \\
a_4 > a5 \\
a_6 > a_5 > a_3 \\
a_4 > a_2
\end{aligned}
$$

这里笔者猜测原文中可能使用了某些似然函数对实验数据进行了拟合，但是文中的具体操作笔者并不了解，这里笔者建议团队内成员使用频率代替概率实现天气信息的预测。

一点心声

师姐我最近有一点窒息emmm，我感觉自己并不能完成这个论文的实现了。平心而论，现在就算是没有这个project也没有空余时间了——何况实现的难度确实略大了一点。

嗯师姐，我之前说过有一些改进的意见，这里就涉及到一些，其实我感觉最恶心的部分就是文中总是尝试使用上每一个考虑到的因素。但是事实上，每一个步骤都应该有非常纯粹的目的，添加过多的考虑反而会使得可解释性非常差，表现出来就是你把论文写出来了，人家看到了，知道你用了哪些因素有衡量标准，但是还是不知道你是怎么实现的。

因为有一些信息的考虑就显得非常冗余，当然全面一些是会很好的，但是有那么一些信息你都不知道要怎么添加到当前的数学模型中来，这就给人一种云里雾里的感觉。我在读论文的时候，有几处这样的地方，就让我感觉到这帮人就是在故弄玄虚。所以在具体的实现中就有很多处我没有使用更多的更复杂的算法，反而是对一些可解释性很差的部分进行删减。师姐有什么想法及时跟我说啊，我要是还活着说不定就向你道声晚安~