

# Python 数据分析与数据挖掘 ( Python for Data Analysis&Data Mining )

## Chap 13 基于定位数据的分析(Location-based Analysis)

---

### 内容：

- 问题背景
- 数据分析和预处理
- 层次聚类算法
- 基于定位数据的聚类分析实现
- 应用：可以使用移动定位数据，GPS数据，wifi数据进行数据分析，轨迹挖掘，用户出行分析等，可以应用到多个重要领域，例如公共安全、智能交通、城市规划与发展、社交网络、旅游等；应运而生许多新的智能应用，在智能交通领域，进行移动用户的热力分析、停留分析、来源去向分析产生智能拼车、共享单车投放调度、滴滴约车智能调度、精确公交时间计算、郊区型高铁站选址、农村人口运力结构配置、区域交通出行特征、枢纽集散交通设计等。

### 实践：

- 层次聚类算法的实现
- 基于定位数据的智能分析

### 实例：

- 实例1：基于定位数据的分析系统

---

这节课是在前面数据分析和数据挖掘的基础上，针对移动定位数据进行面向应用的分析处理和模型构建。本节课通过对移动定位数据进行分析 and 挖掘来进行商圈特征的聚类分析实践，从而为针对不同商圈进行针对性的商业活动来提高营销的精准性，降低盲目营销的成本。本节课从移动定位数据出发进行商业分析，也可以扩展到其他定位数据（GPS，wifi等），并推广到多个其他领域，例如社交网络，公共安全，智能共享，城市规划等应用。

## 准备工作：导入库，配置环境等

In [1]:

```
from __future__ import division
import os, sys

# 启动绘图
%matplotlib inline
import matplotlib.pyplot as plt

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

# 1. 问题背景

## 1. 定位数据的获取

随着个人手机终端的普及，出行群体中手机拥有率和使用率已达到相当高的比例，手机移动网络也基本实现了城乡空间区域的全覆盖。

根据手机信号在真实地理空间上的覆盖情况，将手机用户时间序列的手机定位数据，映射至现实的地理空间位置，即可完整、客观地还原出手机用户的现实活动轨迹，从而挖掘得到**人口空间分布与活动联系**的特征信息。

移动通信网络的信号覆盖逻辑上被设计成由若干个六边形的基站小区相互邻接而构成的蜂窝网络面状服务区，如下图所示，手机终端总是与其中某一个基站小区保持联系，移动通信网络的控制中心会定期或不定期地主动或被动地记录每个手机终端时间序列的基站小区编号信息。

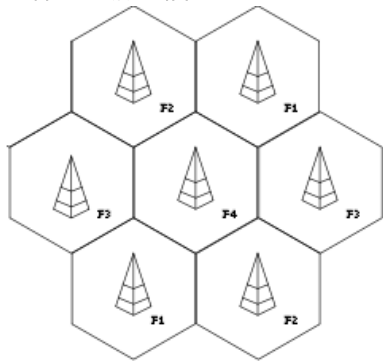


图 1. 移动基站示意图

从某通信运营商提供的特定接口解析得到用户的定位数据，示例如下表：

年	月	日	时	分	秒	毫秒	网络类型	LOC 编号	基站编号	EMASI 号	信令类型
2014	1	1	0	53	46	96	2	962947809921085	36902	55555	333789CA
2014	1	1	0	31	48	38	2	281335167708768	36908	55555	333333CA
2014	1	1	0	17	25	46	3	187655709192839	36911	55558	333477CA
2014	1	1	0	5	40	83	3	232648776184248	36908	55561	333381CA
2014	1	1	0	50	29	4	2	611763545227777	36906	55563	333405CA
2014	1	1	0	1	40	31	2	44710067012246	36909	55563	333717CA
2014	1	1	0	27	32	17	2	975579082112825	36912	55563	333981CA
2014	1	1	0	52	35	83	2	820798260690697	36906	55564	333861CA
2014	1	1	0	11	2	21	3	380420663155326	36910	55564	334149CA
2014	1	1	0	43	38	95	2	897743952380637	36903	55565	334053CA
2014	1	1	0	40	30	87	3	7775693027472	36910	55565	333453CA
2014	1	1	0	1	30	68	3	113404095624425	36911	55565	334125CA
2014	1	1	0	39	20	24	3	393808837659011	36905	55566	334077CA

表 1. 某地某区域的移动基站定位数据示例

这些定位数据各个属性如下表：

序 号	属 性 编 码	属 性 名 称	数 据 类 型	备 注
1	year	年	int	
2	month	月	int	
3	day	日	int	
4	hour	时	int	
5	minute	分	int	
6	second	秒	int	
7	millisecond	毫秒	int	
8	generation	网络类型	int	2 代表 2G, 3 代表 3G, 4 代表 4G
9	loc	LOC 编号	string	15 位字符串
10	cell_id	基站编号	string	基站 ID, 15 位字符串
11	emasi	EMASI 号	string	需要关联用户表取用户号码 (用户号码需要关联用户表得到用户 ID)
12	type	信令类型	string	小于 15 个字符

表 2. 移动基站定位数据属性列表

## 参考文献

[1] 据《21世纪经济报道》报道：工信部部长苗圩在通信展暨ICT中国·2016高层论坛开幕式上致辞时提及，截止今年7月，中国移动电话用户总数达到13.04亿户，其中4G用户总数达到6.46亿户。

[2] 2016年第三季度，美国科技媒体Mashable引述皮尤机构的一份报告称，发展中国家的智能手机普及率已经增加到了37%,发达国家智能手机普及率则为68%,中国的智能手机普及率为58%。

## 2. 基于定位数据应运而生的智能应用

### 1. 商圈划分

商圈是现代市场中企业市场活动的空间，最初是站在商品和服务提供者的产地角度提出的，后来逐渐扩展到商圈，同时也是商品和服务享用者的区域。商圈划分的目的之一是为了研究潜在的顾客的分布以制定适宜的商业对策。

### 2. 网约车供需预测

在出行问题上，中国市场人数多、人口密度大，总体的出行频率远高于其他国家，这种情况在大城市尤为明显。然而，截止目前中国拥有汽车的人口只有不到10%，这也意味着在中国人们的出行更加依赖于出租车、公共交通等市场提供的服务。

另一方面，网约车服务，无论滴滴、Uber、神州专车、首汽约车等，均在不同程度的影响人们的出行和生活方式。网约车面对着巨大的数据量以及与日俱增的数据处理需求。例如，滴滴出行平台每日需处理1400万订单，需要分析的数据量达到70TB，路径规划超过90亿。面对如此庞杂的数据，通过不断升级、完善与创新背后的云计算与大数据技术，保证数据分析及相关应用的稳定，实现高频出行下的运力均衡。

供需预测就是其中的一个关键问题。供需预测的目标是准确预测出给定地理区域在未来某个时间段的出行需求量及需求满足量。调研发现，同一地区不同时间段的订单密度是不一样的，例如大型居住区在早高峰时段的出行需求比较旺盛，而商务区则在晚高峰时段的出行需求比较旺盛。如果能预测到在未来的一段时间内某些地区的出行需求量比较大，就可以提前对营运车辆提供一些引导，指向性地提高部分地区的运力，从而提升乘客的整体出行体验。

2016年滴滴Di-Tech算法大赛竞赛 (<http://research.xiaojukeji.com/>) 提供了订单、交通、天气、位置等数据，预测未来几天中每个时间段的供需差。

### 3. 顺风车拼车智能匹配

拼车是指相同路线的人乘坐同一辆车上下班、上学及放学回家、节假日出游等，车费由乘客分摊。拼车不仅能节省出行费用，也有利于缓解城市交通压力，尤其对于一线城市，因此，拼车和顺风车推荐是轨迹挖掘的研究热点。

现在大部分拼车网站的做法是通过拼车司机在拼车服务网站上发布出发地、目的地、出发时间等信息，再由拼车客户在网站上输入出发地和目的地来搜索符合情况的拼车对象。这种做法很大程度上浪费了拼车用户在望山搜索拼车伙伴的实际，拼车体验较差。

拼车推荐是需要事先对用户的定位数据进行轨迹挖掘，发现用户的轨迹模式集合，再根据两个用户之间移动轨迹模式的相似性，推荐合适的拼车路线。

交通大数据显示，超50%乘客愿意与别人分享，选择拼车出行。以2016年1-10月的武汉为例，有超过2900万人次通过拼车和顺风车出行。据测算，一辆充分使用的分享汽车，如果每次行程能够载2-3组目的地相近的拼车乘客，每天可减少20-40辆私家车上路，大大降低机动车空驶率和上路率。武汉市每天的快车拼车和顺风车出行达10.2万人次，如果按私家车平均每辆车每天出行2次，每次载客1.5人计算，这相当于武汉每天减少3.4万辆小汽车出行。

#### 4. 共享单车投放调度

目前火热的共享单车在城市各区域的投放并不均匀，常常遇到“想骑车却无车可骑”的尴尬。共享单车到底分布在哪里？哪些地方是单车公司忽略的“需求旺地”？爬取共享单车数据，再结合地铁流量数据和房屋地产数据，可以从用户角度为共享单车的投放问题提供一个可能的解决方案。

## 2. 挖掘目标

1. 对用户的历史定位数据，采用数据分析和数据挖掘技术，对基站进行分群。
2. 对不同的商圈分群进行特征分析，比较不同商圈类别的价值，选择合适的区域进行运营商的促销活动。

### 3. 分析方法

#### 1. 定位数据的生成

手机用户在使用短信业务、通话业务、开关机、正常位置更新、周期位置更新和切入呼叫的时候均产生定位数据，定位数据记录手机用户所处基站的编号、时间和唯一标识用户的 EMASI号等。

理论上，一个基站可以覆盖35公里，但在城市的话务量密集区，一般基站相隔0.5公里进行重点覆盖；农村里没有人或人少的地方，覆盖大概1.5-2公里。不同频段的基站覆盖距离是不一样的，2G可以很远，4G可以很近。此外，根据基站的天线高度、下倾角以及设置的参数等，200米到2公里都有可能。

历史定位数据描绘了用户的活动模式，一个基站覆盖的区域可等价于商圈，通过归纳经过基站覆盖范围的人口特征，识别出不同类别的基站范围，即可等同地识别出不同类别的商圈。

衡量区域的人口特征可从人流量和人均停留时间的角度进行分析，所以在归纳基站特征时可针对这两个特点进行提取。

#### 2. 商圈分析的主要步骤

- 1. 从移动通信运营商提供的特定接口上解析、处理、并滤除用户属性后得到用户定位数据。
- 2. 以单个用户为例，进行数据探索分析，研究在不同基站的停留时间，并进一步地进行预处理，包括数据规约、数据变换和数据标准化。
- 3. 利用步骤2 形成的已完成数据预处理的建模数据，基于基站覆盖范围区域的人流特征进行商圈聚类，对各个商圈分群进行特征分析，选择合适的区域进行运营商的促销活动。

基于移动基站定位数据的商圈分析的流程图如下：

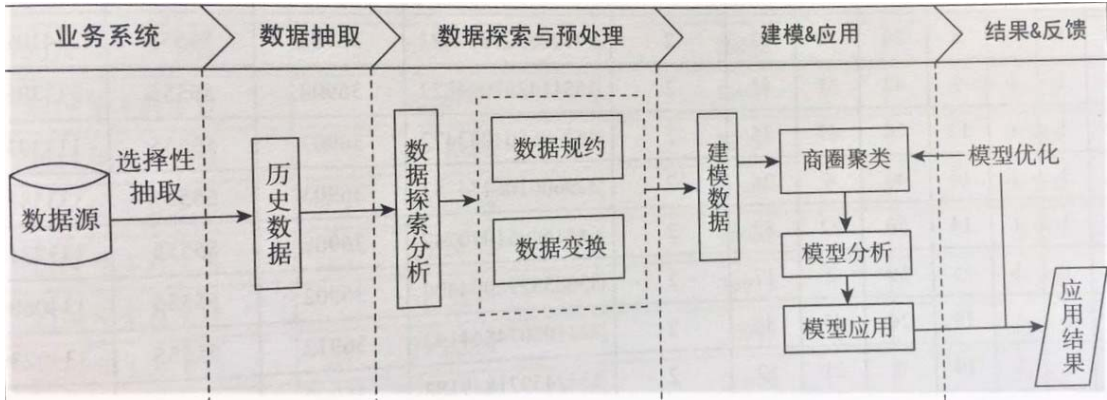


图 2. 基于移动基站定位数据的商圈分析的流程图



# 4. 过程

## 1. 数据抽取

从移动通信运营商提供的特定接口上解析、处理、并滤除用户属性后得到位置数据，以2014年1月1日开始，到2014年6月30日为结束时间，在这个分析观测窗口中，抽取某区域的定位数据来形成建模数据。如前 表1 所示。

年	月	日	时	分	秒	毫秒	网络类型	LOC 编号	基站编号	EMASI 号	信令类型
2014	1	1	0	53	46	96	2	962947809921085	36902	55555	333789CA
2014	1	1	0	31	48	38	2	281335167708768	36908	55555	333333CA
2014	1	1	0	17	25	46	3	187655709192839	36911	55558	333477CA
2014	1	1	0	5	40	83	3	232648776184248	36908	55561	333381CA
2014	1	1	0	50	29	4	2	611763545227777	36906	55563	333405CA
2014	1	1	0	1	40	31	2	44710067012246	36909	55563	333717CA
2014	1	1	0	27	32	17	2	975579082112825	36912	55563	333981CA
2014	1	1	0	52	35	83	2	820798260690697	36906	55564	333861CA
2014	1	1	0	11	2	21	3	380420663155326	36910	55564	334149CA
2014	1	1	0	43	38	95	2	897743952380637	36903	55565	334053CA
2014	1	1	0	40	30	87	3	7775693027472	36910	55565	333453CA
2014	1	1	0	1	30	68	3	113404095624425	36911	55565	334125CA
2014	1	1	0	39	20	24	3	393808837659011	36905	55566	334077CA

表 1. 某地某区域的移动基站定位数据示例

2. 数据探索分析

为了便于观察数据，提取单个用户（如EMASI号为"55555"的用户）在2014年1月1日的定位数据，如下表所示。

年	月	日	时	分	秒	毫秒	网络类型	LOC 编号	基站编号	EMASI 号	信令类型
2014	1	1	0	31	48	38	2	281335167708768	36908	55555	333333CA
2014	1	1	0	53	46	96	2	962947809921085	36902	55555	333789CA
2014	1	1	1	26	11	23	2	262095068434776	36902	55555	333334CA
2014	1	1	2	13	46	28	2	712890120478723	36907	55555	333551CA
2014	1	1	7	57	18	92	2	85044254500058	36902	55555	333796CA
2014	1	1	8	20	32	93	2	995208321887481	36903	55555	334109CA
2014	1	1	9	43	31	45	2	555114267094822	36908	55555	333798CA
2014	1	1	12	20	47	35	2	482996504023472	36907	55555	333393CA
2014	1	1	14	40	4	26	2	329606106134793	36903	55555	333587CA
2014	1	1	14	50	32	82	2	645164951070747	36908	55555	333731CA
2014	1	1	15	19	2	17	2	830855298094409	36902	55555	334068CA
2014	1	1	18	26	43	88	2	323108074844193	36912	55555	334023CA
2014	1	1	19	0	21	82	2	553245971859183	36909	55555	333952CA
2014	1	1	19	50	7	90	2	987606797101505	36906	55555	334096CA
2014	1	1	22	35	0	4	2	756416566337609	36908	55555	333427CA
2014	1	1	23	28	7	98	2	919108833174494	36904	55555	333500CA

表 3. EMASI号为"55555"的用户在2014年1月1日的基站定位数据

可以发现：

1. 用户在2014年1月1日 00:31:48 处于36908基站的范围，然后在2014年1月1日 00:53:46 处于36902基站的范围，表明用户从 00:31:48 到 00:53:46 都是处于36908 基站，共停留了21分58秒，并且在 00:53:46 进入了36902基站的范围。

2. 再下一条记录表明，用户在2014年1月1日 01:26:11 处于36902 基站的范围，这可能是由于用户在进行通话或者其他产生定位数据记录的业务，此时的基站编号未发生改变，用户依旧处于36902 基站的范围，如果要计算用户在36902基站范围停留的实际，则需要继续判断下一条记录，发现用户在2014年1月1日 02:13:46 处于 36907 基站的范围，故用户从 00:53:46 到 02:13:46 都是处于36902 基站，共停留了80分。

该用户的停留示意图如下图所示。

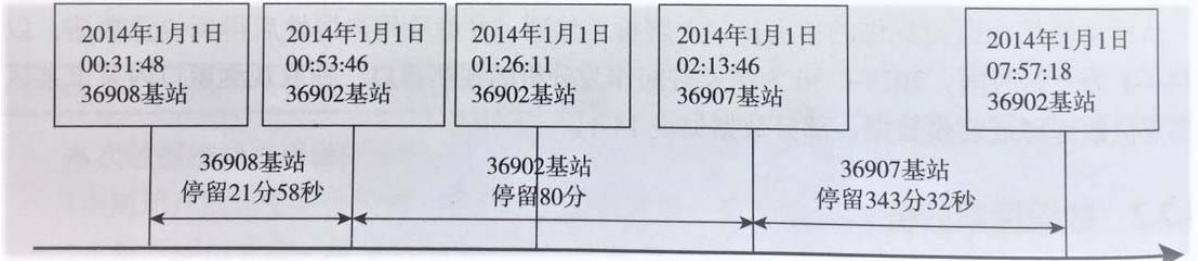


图 3. EMASI号为"55555"的用户在2014年1月1日的停留示意图

### 3. 数据预处理

#### 3.1 数据规约

原始数据的属性较多，但网络类型、LOC编号和 信令类型这3个属性对于挖掘目标没有用处，故剔除这3个冗余的属性。

衡量用户的停留时间，并不需要精确到毫秒级，故也把毫秒这一属性删除。

在计算用户的停留时间时，只计算两条记录的时间差，为了减少数据维度，把年、月和日合并为日期，时、分和秒合并记为时间。

表3 经过规约处理后 得到表4 如下。

日 期	时 间	基 站 编 号	EMASI 号
2014 年 1 月 1 日	00:31:48	36908	55555
2014 年 1 月 1 日	00:53:46	36902	55555
2014 年 1 月 1 日	01:26:11	36902	55555
2014 年 1 月 1 日	02:13:46	36907	55555
2014 年 1 月 1 日	07:57:18	36902	55555
2014 年 1 月 1 日	08:20:32	36903	55555
2014 年 1 月 1 日	09:43:31	36908	55555
2014 年 1 月 1 日	12:20:47	36907	55555
2014 年 1 月 1 日	14:40:04	36903	55555
2014 年 1 月 1 日	14:50:32	36908	55555
2014 年 1 月 1 日	15:19:02	36902	55555
2014 年 1 月 1 日	18:26:43	36912	55555
2014 年 1 月 1 日	19:00:21	36909	55555
2014 年 1 月 1 日	19:50:07	36906	55555
2014 年 1 月 1 日	22:35:00	36908	55555
2014 年 1 月 1 日	23:28:07	36904	55555

表 4. EMASI号为"55555"的用户在2014年1月1日的基站定位数据



### 3.2 特征设计（数据变换）

挖掘的目标是寻找出高价值的商圈，需要根据用户的定位数据提取出衡量基站覆盖范围区域的人流特征，如人均停留时间和人流量等。

高价值的商圈具有人流量大，人均停留时间长的特点；写字楼工作的上班族在白天所处的基站范围基本固定，停留时间也相对较长；住宅区晚上的居民所处的基站范围基本固定，停留时间也相对较长；因此，仅通过 **停留时间** 作为人流特征难以区分高价值商圈和写字楼与住宅区。需要考虑设计提取出的人流特征必须能较为明显的区分这些基站范围。因而，设计一下四个指标作为特征：（1）工作日上班时间人均停留时间、（2）凌晨人均停留时间、（3）周末人均停留时间 和（4）作为基站覆盖范围区域的人流特征。怎么计算这些值？

1. 工作日上班时间人均停留时间：一般上班工作时间是在 9:00 - 18:00，所以工作日上班时间人均停留时间是计算所有用户在工作日 9:00 - 18:00 处于该基站范围内的平均时间。
2. 凌晨人均停留时间：一般在00:00-07:00都是在住处休息，就计算所有用户在该时间段处于该基站范围的平均时间。利用这个指标可以表征出住宅区基站的人流特征。
3. 周末人均停留时间：指所有用户周末处于该基站范围内的平均时间。高价值商圈在周末的逛街人数和时间都会大幅增加，利用这个指标则可以表征出高价值商圈的人流特征。
4. 日均人流量：指平均每天曾经在该基站范围内的人数。日均人流量大说明经过该基站区域的人数多，利用这个指标可以表征出高价值商圈的人流特征。

这四个指标的计算直接从原始数据计算比较复杂，需要先处理成中间过程数据，再从中计算出这四个指标。

### 3.3 特征提取

中间过程数据的计算以单个用户在一天里的定位数据为基础，计算在各个基站范围下的工作日上班停留时间、凌晨停留时间、周末停留时间和是否处于基站范围。

假设整个原始数据中有  $M$  个用户， $N$  个基站，观测窗口期间  $L$  天。

用户  $i$  在  $j$  天中经过了三个基站  $a1$ ， $a2$  和  $a3$ ，则统计该用户  $i$  在  $j$  天内 在三个基站的：

1) 工作日上班时间时间停留时间为：

- 在基站  $a1$  的工作日上班时间停留时间为  $weekdaya1_{ij}$ ；
- 在基站  $a2$  的工作日上班时间停留时间为  $weekdaya2_{ij}$ ；
- 在基站  $a3$  的工作日上班时间停留时间为  $weekdaya3_{ij}$ ；

2) 凌晨停留时间为：

- 在基站  $a1$  的凌晨停留时间为  $nighta1_{ij}$ ；
- 在基站  $a2$  的凌晨停留时间为  $nighta2_{ij}$ ；
- 在基站  $a3$  的凌晨停留时间为  $nighta3_{ij}$ ；

3) 周末停留时间为：

- 在基站  $a1$  的周末停留时间为  $weekenda1_{ij}$ ；
- 在基站  $a2$  的周末停留时间为  $weekenda2_{ij}$ ；
- 在基站  $a3$  的周末停留时间为  $weekenda3_{ij}$ ；

4) 是否在基站停留：

- 在基站  $a1$  是否停留为  $staya1_{ij}$ ；
- 在基站  $a2$  是否停留为  $staya2_{ij}$ ；
- 在基站  $a3$  是否停留为  $staya3_{ij}$ ；

很明显，第四个指标是binary 类型；对于未停留的其他基站，这四个指标的值均为0。

因此，以 基站  $a1$  为例，这四个指标的特征计算公式如下：

1) 工作日上班时间时间停留时间为：
$$weekdaya1 = \frac{1}{LM} \sum_{j=1}^L \sum_{i=1}^M weekdaya1_{ij}$$

2) 凌晨停留时间为：
$$nighta1 = \frac{1}{LM} \sum_{j=1}^L \sum_{i=1}^M nighta1_{ij}$$

3) 周末停留时间为：
$$weekenda1 = \frac{1}{LM} \sum_{j=1}^L \sum_{i=1}^M weekenda1_{ij}$$

4) 日均人流量为：
$$staya1 = \frac{1}{L} \sum_{j=1}^L \sum_{i=1}^M staya1_{ij}$$

同样的，对其他基站，计算公式一样。

对采集到的数据，按基站覆盖范围区域的人流特征进行计算，得到各个基站的样本数据。见 data/business\_circle.xls 文件。

基站编号	工作日上班时间人均停留时间	凌晨人均停留时间	周末人均停留时间	日均人流量
36902	78	521	602	2863
36903	144	600	521	2245
36904	95	457	468	1283
36905	69	596	695	1054
36906	190	527	691	2051
36907	101	403	470	2487
36908	146	413	435	2571
36909	123	572	633	1897
36910	115	575	667	933

表 5. 各个基站的训练数据

### 3.4 数据标准化

从表5的数据发现：各个属性之间的差异较大，为了消除数量级数据带来的影响，在进行聚类前，需要进行离差标准化处理，离差后的数据文件存储在 data/standardized.xls 文件。

In [2]:

```
#-*- coding: utf-8 -*-
#数据标准化到[0, 1]
import pandas as pd

#参数初始化
filename = "data/business_circle.xls" #原始数据文件
standardizedfile = "data/standardized.xls" #标准化后数据保存路径

data = pd.read_excel(filename, index_col = u'基站编号') #读取数据
data.head()
```

Out[2]:

	工作日上班时间人均停留时间	凌晨人均停留时间	周末人均停留时间	日均人流量
基站编号				
36902	78	521	602	2863
36903	144	600	521	2245
36904	95	457	468	1283
36905	69	596	695	1054
36906	190	527	691	2051

In [3]:

```
data = (data - data.min())/(data.max() - data.min()) #离差标准化
data = data.reset_index()
data.head()
```

Out[3]:

	基站编号	工作日上班时间人均停留时间	凌晨人均停留时间	周末人均停留时间	日均人流量
0	36902	0.103865	0.856364	0.850539	0.169153
1	36903	0.263285	1.000000	0.725732	0.118210
2	36904	0.144928	0.740000	0.644068	0.038909
3	36905	0.082126	0.992727	0.993837	0.020031
4	36906	0.374396	0.867273	0.987673	0.102217

In [4]:

```
# 保存结果数据文件
data.to_excel(standardizedfile, index = False) #保存结果
```

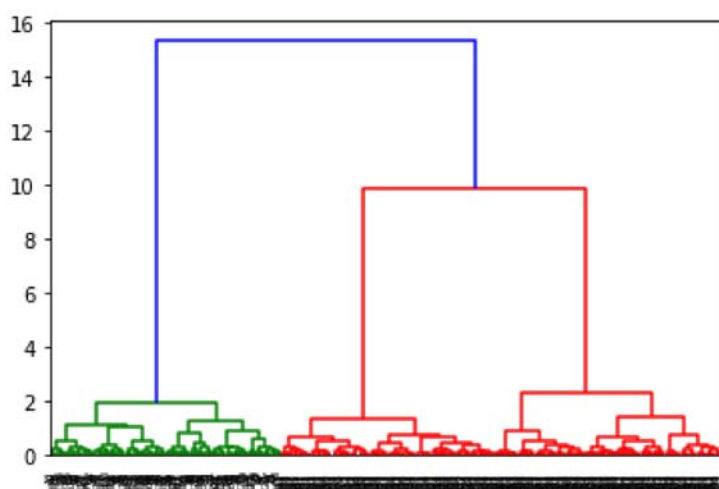
## 4 构建商圈聚类模型

### 4.1 画出谱系聚类图

数据经过上述的预处理之后，形成建模训练数据。采用层次聚类算法（ hierarchical clustering ）对训练数据进行商圈聚类，画出谱系聚类图（ dendrogram ），如下：

In [5]:

```
#-*- coding: utf-8 -*-  
#谱系聚类图  
import pandas as pd  
  
#参数初始化  
standardizedfile = "data/standardized.xls" #标准化后的数据文件  
data = pd.read_excel(standardizedfile, index_col = u'基站编号') #读取数据  
  
import matplotlib.pyplot as plt  
from scipy.cluster.hierarchy import linkage, dendrogram  
#这里使用scipy的层次聚类函数  
  
Z = linkage(data, method = 'ward', metric = 'euclidean') #谱系聚类图  
P = dendrogram(Z, 0) #画谱系聚类图  
plt.show()
```



## 4.2 输出层次聚类模型结果

观察上面的谱系聚类图，可以把聚类类别数取为3类（代码中  $K=3$ ），输出结果typeindex为每个样本对应的类别号。层次聚类算法详见如下：

In [6]:

```
#-*- coding: utf-8 -*-  
#层次聚类算法  
import pandas as pd  
  
#参数初始化  
standardizedfile = "data/standardized.xls" #标准化后的数据文件  
k = 3 #聚类数  
data = pd.read_excel(standardizedfile, index_col = u'基站编号') #读取数据
```



In [7]:

```
data.head()
```

Out[7]:

	工作日上午时间人均停留时间	凌晨人均停留时间	周末人均停留时间	日均人流量
基站编号				
36902	0.103865	0.856364	0.850539	0.169153
36903	0.263285	1.000000	0.725732	0.118210
36904	0.144928	0.740000	0.644068	0.038909
36905	0.082126	0.992727	0.993837	0.020031
36906	0.374396	0.867273	0.987673	0.102217

In [8]:

```
#导入sklearn的层次聚类函数
from sklearn.cluster import AgglomerativeClustering
model = AgglomerativeClustering(n_clusters = k, linkage = 'ward')
model.fit(data) #训练模型
```

Out[8]:

```
AgglomerativeClustering(affinity='euclidean', compute_full_tree='auto',
                        connectivity=None, linkage='ward',
                        memory=Memory(cachedir=None), n_clusters=3,
                        pooling_func=<function mean at 0x0000000006321B38>)
```

In [9]:

```
#详细输出原始数据及其类别
r = pd.concat([data, pd.Series(model.labels_, index = data.index)], axis = 1) #详细输出每个样本对应的类别
r.columns = list(data.columns) + [u'聚类类别'] #重命名表头
```

In [10]:

```
r.head()
```

Out[10]:

	工作日上班时间人均停留时间	凌晨人均停留时间	周末人均停留时间	日均人流量	聚类类别
基站编号					
36902	0.103865	0.856364	0.850539	0.169153	1
36903	0.263285	1.000000	0.725732	0.118210	1
36904	0.144928	0.740000	0.644068	0.038909	1
36905	0.082126	0.992727	0.993837	0.020031	1
36906	0.374396	0.867273	0.987673	0.102217	1

In [11]:

```
r.tail()
```

Out[11]:

	工作日上班时间人均停留时间	凌晨人均停留时间	周末人均停留时间	日均人流量	聚类类别
基站编号					
35562	0.125604	0.081818	0.291217	0.608771	0
38624	0.152174	0.072727	0.354391	0.590718	0
36017	0.205314	0.003636	0.129430	0.973539	0
38827	0.154589	0.089091	0.118644	0.927129	0
37787	0.154589	0.001818	0.329738	0.802984	0

4.3 画出四个指标特征的折线图

针对聚类结果按不同类别画出4个特征的折线图，如下：

In [ ]:

```
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号

style = ['ro-', 'go-', 'bo-']
xlabels = [u'工作日人均停留时间', u'凌晨人均停留时间', u'周末人均停留时间', u'日均人流量']
pic_output = 'data/type_' #聚类图文件名前缀

for i in range(k): #逐一作图，作出不同样式
    plt.figure()
    tmp = r[r[u'聚类类别'] == i].iloc[:, :4] #提取每一类
    for j in range(len(tmp)):
        plt.plot(range(1, 5), tmp.iloc[j], style[i])

    plt.xticks(range(1, 5), xlabels, rotation = 20) #坐标标签
    plt.title(u'商圈类别%s' % (i+1)) #我们计数习惯从1开始
    plt.subplots_adjust(bottom=0.15) #调整底部
    plt.savefig(u'%s%s.png' % (pic_output, i+1)) #保存图片
```

#### 4.4 商业营销分析

从上面的聚类结果得到三个商圈的中4个特征指标的折线图：

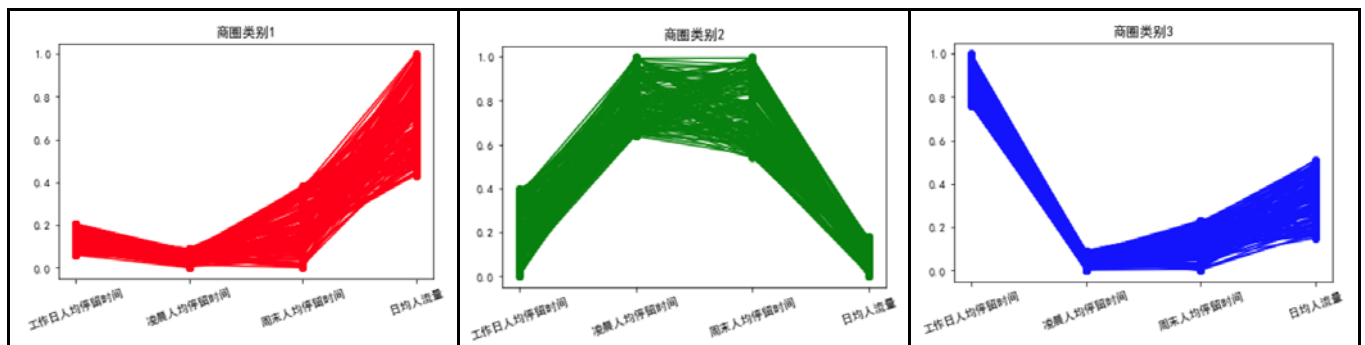


图 4. 三个商圈的折线图

观察：

1. 对于商圈1，日均人流量较大，工作日上午时间人均停留时间、凌晨人均停留时间和周末人均停留时间相对较短，该类别基站覆盖的区域类似于 商业区；
2. 对于商圈2，凌晨人均停留时间和周末人均停留时间较长，而工作日人均停留时间较短，日均人流量较少，该类别基站覆盖的区域类似于 住宅区；
3. 对于商圈3，工作日上午时间人均停留时间较长，而凌晨人均停留时间、周末人均停留时间较短，该类别基站覆盖的区域类似于 上班族的工作区域；

分析：

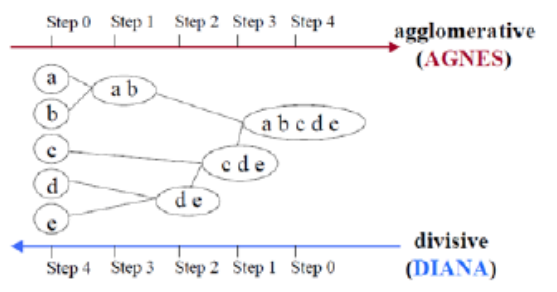
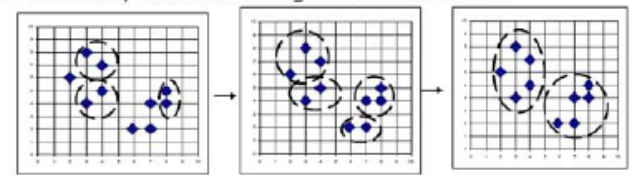
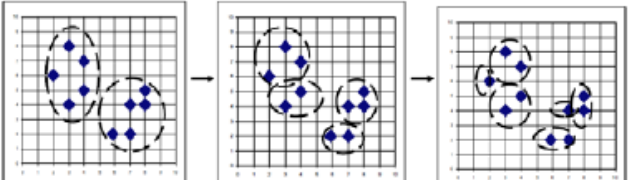
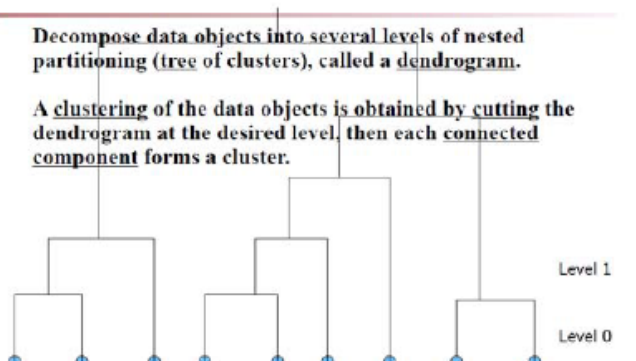
1. 商圈2 的人流量较少，商圈3的人流量一般，考虑上班族的工作区域通常人员流动集中在上、下班时间和午饭时间，这两类商圈不利于运营商展开促销活动；
2. 商圈1的人流量大，在这样的商业区有利于进行运营商的促销活动。

## 4.5 层次聚类算法概述

### 聚类算法中的基本概念

Distance between Clusters	Centroid, Radius and Diameter of a Cluster (for numerical data sets)
<ul style="list-style-type: none"> <li><b>Single link:</b> smallest distance between an element in one cluster and an element in the other, i.e., <math>\text{dis}(K_i, K_j) = \min(t_{ip}, t_{jq})</math></li> <li><b>Complete link:</b> largest distance between an element in one cluster and an element in the other, i.e., <math>\text{dis}(K_i, K_j) = \max(t_{ip}, t_{jq})</math></li> <li><b>Average:</b> avg distance between an element in one cluster and an element in the other, i.e., <math>\text{dis}(K_i, K_j) = \text{avg}(t_{ip}, t_{jq})</math></li> <li><b>Centroid:</b> distance between the centroids of two clusters, i.e., <math>\text{dis}(K_i, K_j) = \text{dis}(C_i, C_j)</math></li> <li><b>Medoid:</b> distance between the medoids of two clusters, i.e., <math>\text{dis}(K_i, K_j) = \text{dis}(M_i, M_j)</math> <ul style="list-style-type: none"> <li>Medoid: one chosen, centrally located object in the cluster</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li><b>Centroid:</b> the "middle" of a cluster <math display="block">C_m = \frac{\sum_{i=1}^N (t_{ip})}{N}</math></li> <li><b>Radius:</b> square root of average distance from any point of the cluster to its centroid <math display="block">R_m = \sqrt{\frac{\sum_{i=1}^N (t_{ip} - c_m)^2}{N}}</math></li> <li><b>Diameter:</b> square root of average mean squared distance between all pairs of points in the cluster <math display="block">D_m = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^N (t_{ip} - t_{jq})^2}{N(N-1)}}</math></li> </ul>

### 层次聚类算法概述

Hierarchical Clustering	AGNES (Agglomerative Nesting)
<ul style="list-style-type: none"> <li>Use distance matrix as clustering criteria. This method does not require the number of clusters <math>k</math> as an input, but needs a termination condition.</li> </ul> 	<ul style="list-style-type: none"> <li>Introduced in Kaufmann and Rousseeuw (1990)</li> <li>Implemented in statistical analysis packages, e.g., Splus</li> <li>Use the <b>Single-Link</b> method and the dissimilarity matrix.</li> <li>Merge nodes that have the least dissimilarity</li> <li>Go on in a non-descending fashion</li> <li>Eventually all nodes belong to the same cluster</li> </ul> 
DIANA (Divisive Analysis)	Dendrogram: Shows How the Clusters are Merged
<ul style="list-style-type: none"> <li>Introduced in Kaufmann and Rousseeuw (1990)</li> <li>Implemented in statistical analysis packages, e.g., Splus</li> <li>Inverse order of AGNES</li> <li>Eventually each node forms a cluster on its own</li> </ul> 	<p>Decompose data objects into several levels of nested partitioning (tree of clusters), called a <b>dendrogram</b>.</p> <p>A clustering of the data objects is obtained by <u>cutting the dendrogram at the desired level</u>, then each <u>connected component</u> forms a cluster.</p> 

### 对层次聚类算法的评价

## Extensions to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
  - can never undo what was done previously
  - do not scale well: time complexity of at least  $O(n^2)$ , where  $n$  is the number of total objects