队伍编号	335
赛道	（A）

基于 Kmeans 和 Kshape、LSTM 和 Cornish-Fisher 展式的基站流量分类与阈值设定研究

摘 要

时间序列特征提取与分类(问题一)：根据题目要求需要基于相关小区的历史流量数据提取时间序列数据特征进行“小区”分类，并描述每类的特点。但由于题目所提供的数据过于庞大，如果直接对原始数据进行清洗并在处理后的原始数据基础上进行数学建模，虽然得到的分类结果更加全面，但是考虑到时间效率与处理机器性能的限制可操作性较低。因此，本文考虑随机抽取 3 万个小区最为训练测试样本集进行初步聚类分析。再基于改进的 KNN 算法将剩余样本小区归为与其距离最近的类别。

本文的聚类流程主要由以下几步。第一步：对原始数据进行数据预处理，出于大样本大数据的考虑，采取随机抽样获得测试用样本小区数据集；第二步：利用 tsfresh 工具提取时间序列的统计特征、熵特征和分段特征等作为对应的特征向量进行聚类，得到特征向量 F。同时，基于随机森林法对构成的特征向量各个特征之间的重要性进行分析。第三步：根据轮廓系数和肘部法则获取最优聚类数，再利用 kmeans 方法进行基于特征的聚类分析并对各类数据特点进行描述；第四步：运用改进的 KNN 模型将非测试用样本小区分类。最终结果表明，上述方法可以有效的提高处理效率和精确度，具有一定的现实可行性。

基站流量阈值设置策略(问题二)：针对问题二中提出的上下行流量阈值设定的问题，本文根据处理后的流量数据及其分布特征，分别采用 Kshape 聚类方法和 Kmeans 聚类方法将时间维度上流量预期波动差异较大的小区和截面维度上流量分布差异较大的小区分离开来，分别得到聚类结果 I 和聚类结果 II。根据两种聚类结果，使用 LSTM 模型从聚类结果 I 中得到上下行流量的预期值，然后使用 Cornish-Fisher 展式从聚类结果 II 中得到去均值后的流量分布分位数，并将上下行流量的预期值与其去均值后的流量分布分位数相结合，得到不同分位数下的上下行流量的预期阈值。

问题二的实证研究结果表明：其一，采用 LSTM 模型和 Cornish-Fisher 展式相结合的方法能够根据上下行流量潮汐效应的变化和随机扰动分布特征的变化来动态地设定阈值，克服了静态阈值设定与流量潮汐效应的矛盾，实现了在对用户体验影响较小的条件下节约能源的目的；其二，上下行流量阈值的设定受到流量预期和各时点随机扰动项的共同影响，在应用过程中应当根据实际需要将阈值设定在不同的分位数水平；其三，采用分位数的方法来设定阈值，有利于决策者把控在多大概率不影响用户体验的基础上降低能源消耗。

关键词：Kmeans 聚类、LSTM 模型、Cornish-Fisher 展式、Kshape 聚类、改进的 KNN

目录

一、问题重述

1.1 研究的背景

互联网业务的快速发展，使得移动流量数据呈指数式增长。根据思科的流量预测报告显示，2020 年的移动数据流量相比 2016 年增长近 5 倍，达 30.6Ebyte 左右。因此，基站的负荷和能源消耗问题受关注度逐渐提升，有效设置自动开关优化基站运行成为了重要问题。此外，基于基站覆盖区域内用户和业务等的不同考量，不同的基站之间的流量模式有着显著的差异。例如，对于覆盖住宅区域活商业街的基站来说其流量传输高峰一般在晚间，而对于餐馆来说其流量传输高峰主要出现在中午和晚上的用餐时段。在对于基站的流量模式分析的过程中，传统的划分方法主要是通过依据基站所处的场景进行区分，但该方法并不具有代表性，无法对相似的场景或混合场景进行有效区分。有鉴于此，如何有效地对数据流量建模并对其进行精准分类，再根据分类结果合理设置基站流量阈值实现优化控制成为了研究的重点。

1.2 研究的问题

有鉴于上述背景，本文需要研究解决以下问题：

问题 1：提取“小区”样本时间序列数据特征并分类

依据附件 1 中提供的小区从 2018 年 3 月 1 日到 4 月 19 日的小时级训练用流量数据与其他相关资料，基于数理统计与移动通信流量理论，依靠数学建模知识提取有关的时间序列数据特征。根据提取的特征对附件 1 所有的小区进行分类，同时描述各类的特点。

问题 2：给出“小区”上下行流量的长期预测模型并预测

基于用户使用体验和资源节约的角度，并结合前文分析结果，给出能够有效优化基站运行、节约能源，达到自动开关限制部分载频目的的基站流量阈值设置策略和具体结果。

二、问题分析

2.1 针对问题一的分析

根据题目要求需要基于相关小区的历史流量数据提取时间序列数据特征进行“小区”分类，并描述每类的特点。问题一已给的条件是附件 1 中提供的 132279 个小区从 2018 年 3 月 1 日到 4 月 19 日共 50 天左右的基站上下行流量数据。对数据进行初步分析后发现提供的历史数据量极为庞大，同时存在大量的数据缺失和冗余信息等数据缺陷问题。由于题目所提供的数据过于庞大，如果直接对原始数据进行清洗并在处理后的原始数据基础上进行数学建模，虽然得到的分类结果更加全面，但是考虑到时间效率与处理机器性能的限制可操作性较低。因此，本文先针对原始数据进行初步的预处理和筛选，在此基础上从所有小区样本中随机抽取 30000 个小区作为样本数据集进行特征提取与分类。对于剩余的样本数据点，基于 K-近邻模型计算其与每类聚类中心的距离大小，将代划分

样本小区归为与其距离最近的类别。这样可以有效的缩小处理的样本量，并提高处理效率和精确度。

此外，还需要在抽取的样本数据集上作进一步的处理，将异常数据样本进行剔除，以前一天和后一天相同时间段的数据均值对空缺值进行插补，同时将冗余数据删除。然后，本文首先参考已有研究文献[1]内容考虑将时间序列数据的性质变化问题转化为一个静态处理问题。例如：根据样本数据的方差与均值将序列转化成具有二维属性的特征向量，再依据所获得的特征向量对时间序列进行描述。这种表述方法既可以有效地提取数据特征又可以对时间序列起到降维的效果。因此，本文参考已有研究文献后选择主要采用依靠特征向量来进行聚类分析[2]。其次，为了降低模型复杂度和提高运行效率、聚类效果，本文的聚类流程主要由以下几步。第一步：对原始数据进行数据预处理，出于大样本大数据的考虑，采取随机抽样获得测试用样本小区数据集；第二步：利用 tsfresh 工具提取时间序列的统计特征、熵特征和分段特征等作为对应的特征向量进行聚类，得到特征向量 F。同时，基于随机森林法对构成的特征向量各个特征之间的重要性进行分析。第三步：根据轮廓系数和肘部法则获取最优聚类数，再利用 kmeans 方法进行基于特征的聚类分析并对各类数据特点进行描述；第四步：运用 DTW 算法改进的 KNN 模型将非测试用样本小区分类。问题一的总体思路如图 2.1 所示。

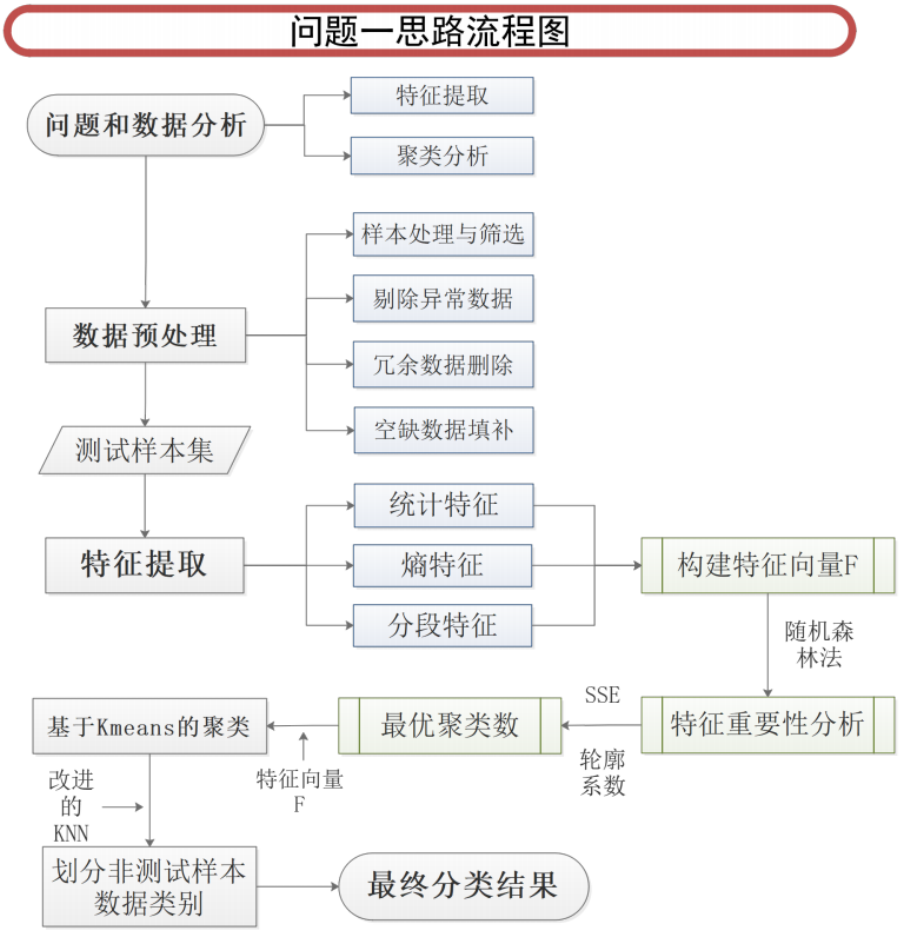

图 2.1 问题一思路流程图

2.2 针对问题二的分析

问题二要求根据流量的变化设定阈值来限制部分载频。一方面，阈值设定过高会造成基站的能源消耗过大，造成资源的浪费，且过高的载频冗余并不会提高用户的使用体验；另一方面，阈值设定过低虽然可以节约能源，但会使得用户的上网体验较差。因此，适当的阈值的设定需要在良好的用户体验和能源节约之间找到一个合理平衡点。

基于此，本文将基站的实际上下行流量的负载分为两个部分：流量使用的期望值部分和随机扰动部分。流量的期望值由 LSTM 模型进行预测，其表示上下行流量在不同时点预期的大小。但如果将阈值设定为期望值，则由于随机扰动的存在，在很大概率上实际的上下行流量会超过阈值，从而带来较差的用户体验。因此，还需要在预测值的基础之上加上一个随机扰动的分位数。一般而言，在很多研究中常常将随机项设定为正态分布的形式，但事实上并非如此。一方面，由于潮汐效应的存在，不同时点随机项的分布存在着很大的差异。例如，在基站负载较高的时间段，扰动项的方差也会相对较大，反之亦然。另一方面，随机项在偏度和峰度等分布特征方面与正态分布也存在着较大的差异。因此，为解决该问题，本文使用 Cornish-Fisher 展式来近似求解一天内（24 小时）不同时点的随机扰动项的分布形式，从而计算得到一个较为准确的随机扰动项分位数。

根据以上的问题分析，本文对于阈值的设定分为了两部分是合理的。第一部分为上下行流量预测值，也即是上下行流量均值的预测；第二部分为 0 均值下的分位数的计算。具体过程如下图所示：

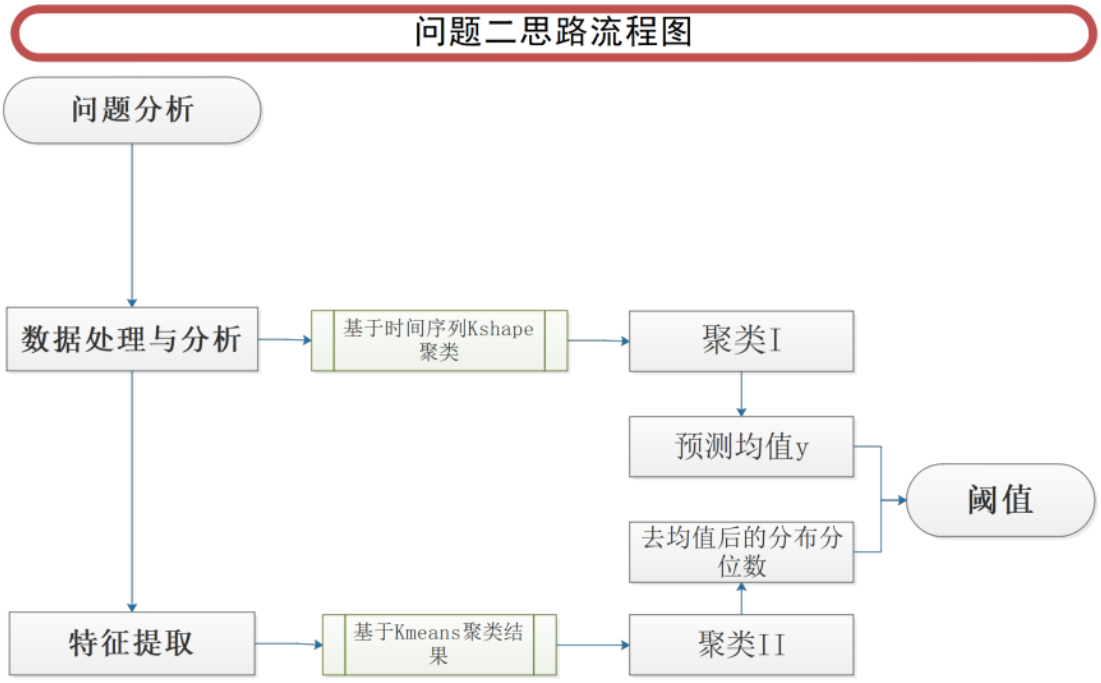

图 2.2 问题二思路流程图

首先，根据所提出的问题对各基站的数据进行基本的特征提取并对数据进行初步的特征分析；其次，对数据进行处理，处理过程包括：剔除缺失数据过多的样本、异常值数据的处理、冗余数据的处理和填补缺失值；然后，根据处理后的数据基于 Kshape 聚类法得到聚类 I，其目的在于将时序变化趋势差异较大基站分离开来，以便使用 LSTM

模型得到更好的上下行流量时序均值的预测结果；再次，从处理后的数据中提取出特征值并使用 Kmeans 聚类方法得到聚类 II，其目的在于将不同时点上下行流量分布差异较大的基站分离开来，以便使用 Cornish-Fisher 展式得到相对准确的上下行流量分位数值；最后，根据上下行流量的预测均值和去均值后的分布分位数计算出在一定概率（例如：95%）下，基站能够满足用户使用时的阈值。

三、模型假设

1. 样本缺失过多的小区能为有效预测提供的必要信息越少；

2. 同一小区在前后相邻日期的内同一时刻流量无显著性差异；

3. 在阈值设定时，假设用户满意度只与基站载频的大小有关；

4. 基于 Cornish-Fisher 展式，标准正态分布可以很好地拟合出标准化后的上下行流量随机扰动项的原分布。

四、符号定义与说明

符号	符号说明
Date	日期
Hours	时刻(小时)
Xiaoqu_num	小区编号
Up	上行业务量 GB
Down	下行业务量 GB
f_1	均值
f_2	方差
f_3	偏度
f_4	峰度
f_5	分组熵
f_6	近似熵
f_7	一阶差分绝对和
f_8	ADF 检测统计值
f_9	时序数据复杂度
f_{10}	傅里叶变换系数
f_{11}	傅里叶变换频谱统计量
f_{12}	各阶自相关系数的聚合统计特征
f_{13}	小波变换
s	轮廓系数
CC	互相关系数
SBD	形状距离测度
C	Kmeans 聚类中心

$F_{k,t}^{i}(x)$	标准化残差真实分布
$\Phi(v)$	标准正态分布函数
x_p，v_p	$F_{k,t}^{i}(x)$，$\Phi(v)$的 p 分位数
$z_{k,t}^{i,q}$	Q 分位数下的阈值
$F_{k,t}^{i\ -1}(x)$	$F_{k,t}^{i}(x)$的反函数
$y_{k,t}^{i}$	上、下行流量的预测值
h	神经元的输出结果
x	神经元当前输入
f	遗忘门输出矩阵
σ	Sigmoid 激活函数
W	权重参数矩阵

五、问题一的模型的建立与求解

5.1 数据预处理

5.1.1 样本的描述与筛选

表 5.1 样本数据的描述性说明

	符号	描述性说明
日期	Date	2018.03.01- 2018.04.19
时间	Hours	以小时为单位，0 到 23 时
小区编号	Xiaoqu_num	小区编号 1-132279
上行业务量 GB	Up	向网络发送的字节数
下行业务量 GB	Down	从网络中下载的字节数
样本总量	Sample_sum	样本总量 144138200

针对附件 1 中所给的数据进行初步描述性分析可得表 5.1。从表 5.1 中可知样本数据集一共有 132279 个小区(基站的扇区)，每个小区都有 50 天，每天 24 小时共 1200 个小时的数据样本。针对附件 1 对每个小区的样本数据量进行统计可以得到如图 5.1 所示的样本数据分布频数直方图。从图 5.1 可知，总得来看绝大部分小区样本数据量在 1000 以上，部分小区数据量大于 2000，说明这部分小区数据样本存在着重复值。同时，发现还有部分小区样本数据量远小于 1000。根据假设，样本数据量缺失过多的样本小区所蕴含的有用的信息越少。同时有鉴于样本小区数量充足，本文考虑删除样本量小于 1000 的所有小区，共剔除小区 18879 个，最后剩余 113440 个小区样本。

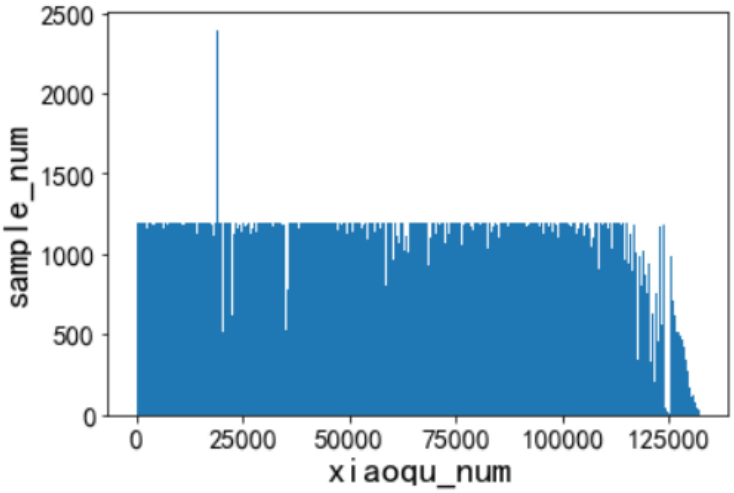

图 5.1 各小区样本数据量分布直方图

5.1.2 样本数据清洗

此外，可发现题目所给附件的数据量过于庞大。基于处理效率与机器运行性能的考量，本文从所有小区中随机抽取 30000 个小区作为样本集进行特征提取和聚类分析。同时，对其采取进一步的数据清洗与处理。针对随机抽取的 30000 个小区进一步的分析，发现部分小区数据仍然存在较多缺失值，且不同小区缺失值存在的时间点不同。其中，对于所有小区 4 月 15 号当天所有数据均缺失。基于同一小区在相邻日期内无显著差异的假设，针对空缺值本文选用同一时刻相邻日期的流量均值来填补。对数据样本重复情况进行统计分析可知，共有 23 个小区样本量大于 2000，通过 python 中 pandas 库自带的函数删除重复值。对于异常值，删除后同样用同一时刻相邻日期的流量均值填补，数据预处理流程图如图 5.2 所示。

图 5.2 数据预处理流程图

5.2 基站一般分类与时间序列聚类

在一般的基站分类处理过程中，对于基站所属的类别往往是依靠其覆盖区域的主导业务场景来决定的，例如，当商业中心、餐馆和街道等场景被同一个基站覆盖时，该基站通常被归为商业中心。然而，对于传统的基站分类方法主要存在以下几个问题：首先，对于基站覆盖区域内主要场景的判别需要依靠地图或者已有信息条件，在实际实现上具有一定的困难度，尤其是对于大样本基站流量数据；其次，仅仅依靠场景去判别基站类别完备性较低，因为基站覆盖范围内的场景是多样化的，无法有效地识别出其主要场景；最后，即使业务场景是多种多样的，但是多样化的场景之间仍然具有一定的相似性，单纯依靠场景的类别划分可能会影响基站的最终分类效果。而基于流量数据时间序列的聚类只依赖于数据自身的信息，可以有效地避免上述问题。

时间序列聚类的方法主要有两种：一是针对时间序列数据的描述即时间域来进行聚类分析，主要通过计算序列之间的距离来实现。而这更多地取决于序列的长短和聚类算法。二是考虑将时间序列数据的性质变化问题转化为一个静态处理问题。例如：根据样本数据的方差与均值将序列转化成具有二维属性的特征向量，再依据所获得的特征向量对时间序列进行描述。这种表述方法既可以有效地提取数据特征又可以对时间序列起到降维的效果。有鉴于此，本文选择基于基站流量数据的时间序列特征的聚类方法来对基站进行分类研究。

5.3 基于 tsfresh 工具的特征提取

5.3.1 特征选择

针对基站流量数据的时间序列特征，本文主要考虑提取三大类数据特征，分别为时间序列的统计特征、熵特征和分段特征。

(1) 时间序列的统计特征

本文参考已有研究文献内容后，主要考虑选取均值、方差、偏度与峰度作为时序特征[3]，同时也包括序列数据的极值、中位数和标准差等。若将长度为 T 的时间序列数据表示为$X_T = \{x_1, \cdots, x_T\}$,则上述统计量的公式分别为：

$$\begin{aligned} &\mu = \frac{1}{T}\sum_{i=1}^{T} x_i, \sigma^2 = \sum_{i=1}^{T} \frac{1}{T}(x_i - \mu)^2 \\ &skewness(X) = \frac{1}{T} \\ &kurtosis(X) = \frac{1}{T} \end{aligned} \tag{5.1}$$

其中，μ和σ^2分别表示为均值和方差，$skewness(X)$表示偏度，$kurtosis(X)$表示峰度。

(2) 时间序列的熵特征

熵作为描述一组数据的确定性和不确定性的指标，可以有效地说明系统地混乱性。熵值越大数据越为混乱，熵值越小数据系统越为稳定。具体公式表示如下：

$$entropy(X) = -\sum_{i=1}^{\infty} P\{x = x_i\} \ln\left(P\{x = x_i\}\right) \tag{5.2}$$

此外，时间序列熵的主要有分组熵(Binned Entropy)、样本熵(Sample Entropy)、近似

熵(Approximate Entropy)。

1) 分组熵

分组熵用来表示序列的分布情况，若时间序列数据的分组熵越大，序列数据在最大与最小值之间的分布就越均匀，反之，则说明序列数据较为集中的在某一范围内。

2) 样本熵

样本熵主要用来表示该组序列的自相似性，样本熵越小表明这一序列的自相似性越强。

3) 近似熵

近似熵用来衡量序列的趋势是否是随机出现的，近似熵越小该时间序列中的重复片段越多，近似熵越大，则表明该条时间序列的趋势是随机出现的。

(3) 时间序列的分段特征

针对时间序列的分段特征主要是将时间序列划分成多段，对于每一段用一个线性函数表示，主要有分段线性逼近、分段聚合逼近和分段常数逼近三种方法。主要用来提取具有循环趋势的序列特征。

根据上述分析并参考已有相关文献[4][5]后本文构建的特征向量组成如表 2 所示。

表 5.1 构建的时间序列特征向量组成

符号	特征说明
f_1	均值
f_2	方差
f_3	偏度
f_4	峰度
f_5	分组熵
f_6	近似熵
f_7	一阶差分绝对和
f_8	ADF 检测统计值
f_9	时序数据复杂度
f_{10}	傅里叶变换系数
f_{11}	傅里叶变换频谱统计量
f_{12}	各阶自相关系数的聚合统计特征

(4) tsfresh 工具的使用

在特征提取的过程中，指数平滑和滑动平均等方法是较为常用的数据特征提取方法，然而对于大样本数据，在实际实现过程中具有一定的困难度同时效率较低。而 python 中的 tsfresh 工具可以搞笑快速的提取出大量的时间序列数据的特征，从中选择我们需要的特征构成特征向量。本文使用 tsfresh 工具提取的特征结果见附件 1。

5.3.2 基于随机森林的小区流量特征重要性分析

经过上面 5.3.1 节的特征提取，我们得到的小区上下行流量相关的 13 个特征，并对特征的相关性进行了分析，但是并不是所有的特征对小区流量的分类都能起到有效作用。数据和特征决定了机器学习的上限，而机器学习的算法只是努力地逼近该极限，如果将

过多冗余的特征放入分类器中，不但会影响计算效率，还会降低模型的分类准确性。因此筛选出有效的特征对实现小区基站的准确分类具有重要的意义。

本文使用随机森林的方法对小区特征的重要性进行分析，为下一步特征的筛选以及特征的改造提供理论依据和数据支持。随机森林算法是有多个决策树加权集合构成的，它有效地克服了决策树容易过拟合的特点，采用多个决策树的投票机制来改善决策树。根据特征选择的准测不同，决策树算法可以分为以下三种：ID3 算法、C4.5 算法和 CART 算法，它们所对应的特征选择准则分别为信息增益、信息增益率和基尼指数。本文所使用的随机森林分类器采用的是 CART 分类树的算法，需要在每次分叉的时候计算基尼指数，它反映了样本集合中一个随机选中的样本被分错的概率。基尼指数越小表示集合中被选中的样本被分错的概率越小，也就是说集合的纯度越高，反之，集合越不纯。在本文中，基尼指数等于小区样本被选中的概率乘以小区样本被分错的概率，可用公式表示为：

$$Gini(p)=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^2 \qquad (5.3)$$

其中 p_k 表示选中的样本属于 k 类别的概率，则这个样本被分错的概率是 $1-p_k$，样本中有 K 个类别，一个随机选中的样本可以属于这 k 个类别中的任意一个。基于特征 A 划分样本集合 D 的基尼指数可以表示为：

$$Gini(D)=1-\sum_{k=1}^{K}\left(\frac{C_k}{D}\right)^2 \qquad (5.4)$$

由于 CART 树是一种二叉树，即使用某个特征进行样本分类只有两个集合，一个是满足给定特征值的样本集合 D_1 和不满足给定特征值的样本集合 D_2，基于上述划分，可以计算出将样本集合划分为两个子集的纯度：

$$Gini(D,A)=\frac{|D_1|}{|D|}Gini(D_1)+\frac{|D_2|}{|D|}Gini(D_2) \qquad (5.5)$$

因而对于一个具有多个取值（超过 2 个）的特征，需要计算以每一个取值作为划分点，对样本 D 划分之后的子集的纯度 $Gini(D,A_i)$，其中 A_i 表示 A 可能取的值。然后从所有可能划分的 $Gini(D,A_i)$ 找出基尼指数最小的划分，这个划分的划分点，便是使用特征 A 对样本集合 D 进行划分的最佳划分点。

下面本文使用 python 中机器学习库的随机森林算法对上下行小区流量的特征进行特征重要性的计算。由于随机森林是一种有监督的学习算法，因此在输入变量的时候需要给分类器提供样本所属标签，本文使用的是在初赛阶段进行时间序列聚类所得到的标签。随机森林分类器的其他参数设定如下：

表 5.3　随机森林分类器参数设定

参数名	参数设定
criterion	Gini
splitter	best

min_samples_split	2
min_samples_leaf	1
min_weight_fraction_leaf	0
min_impurity_decrease	0
presort	deprecated
ccp_alpha	0

下面给出随机森林的分类基于特征进行样本分类的示意图，图中的 gini 表示的是基尼不纯度，是上文中基尼指数的指标的倒数，因此基尼不纯度越高，说明对分叉对子样本的区分会更精确。由于本文篇幅有限，因此只给出决策树分类的前三层示意图：

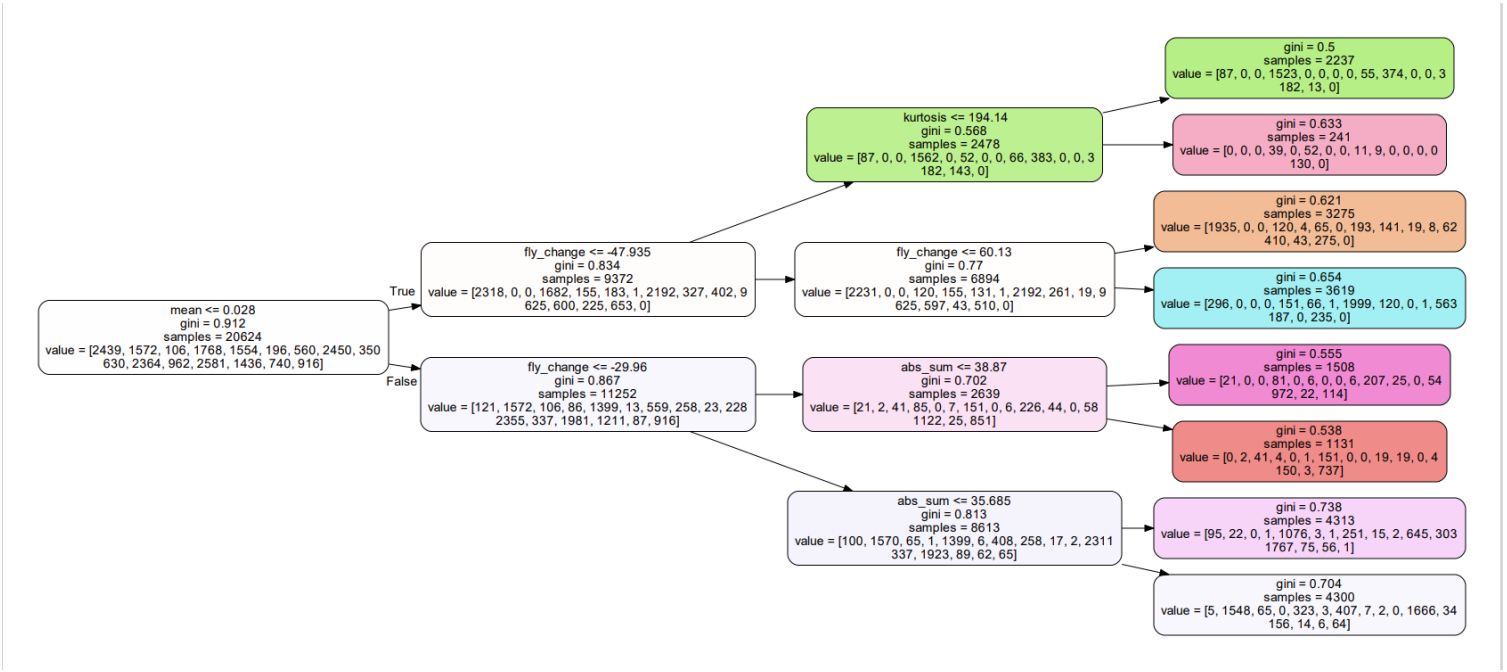

图 5.3 **上行流量随机森林分类图**

由上图可知，在根节点上，样本特征中的均值 mean<=0.028 节点处所计算得到基尼不纯度最高，达到 0.912，因此随机森林分类器选择了在此进行分叉；在一层节点上，傅里叶变换系数的 fly_change 的基尼不纯度最高，两个节点基尼不纯度分别达到 0.864 和 0.834，因此在此层上根据 fly_change<=-47.935 和 fly_change<=-29.96 进行分叉；在第二层节点上，峰度 kurtosis、一阶差分绝对和 abs_sum 的基尼不纯度最高，分类器根据计算得到的节点基尼不纯度进行分叉。

类似地，本文也使用了随机森林对下行流量的基站特征进行了分类：

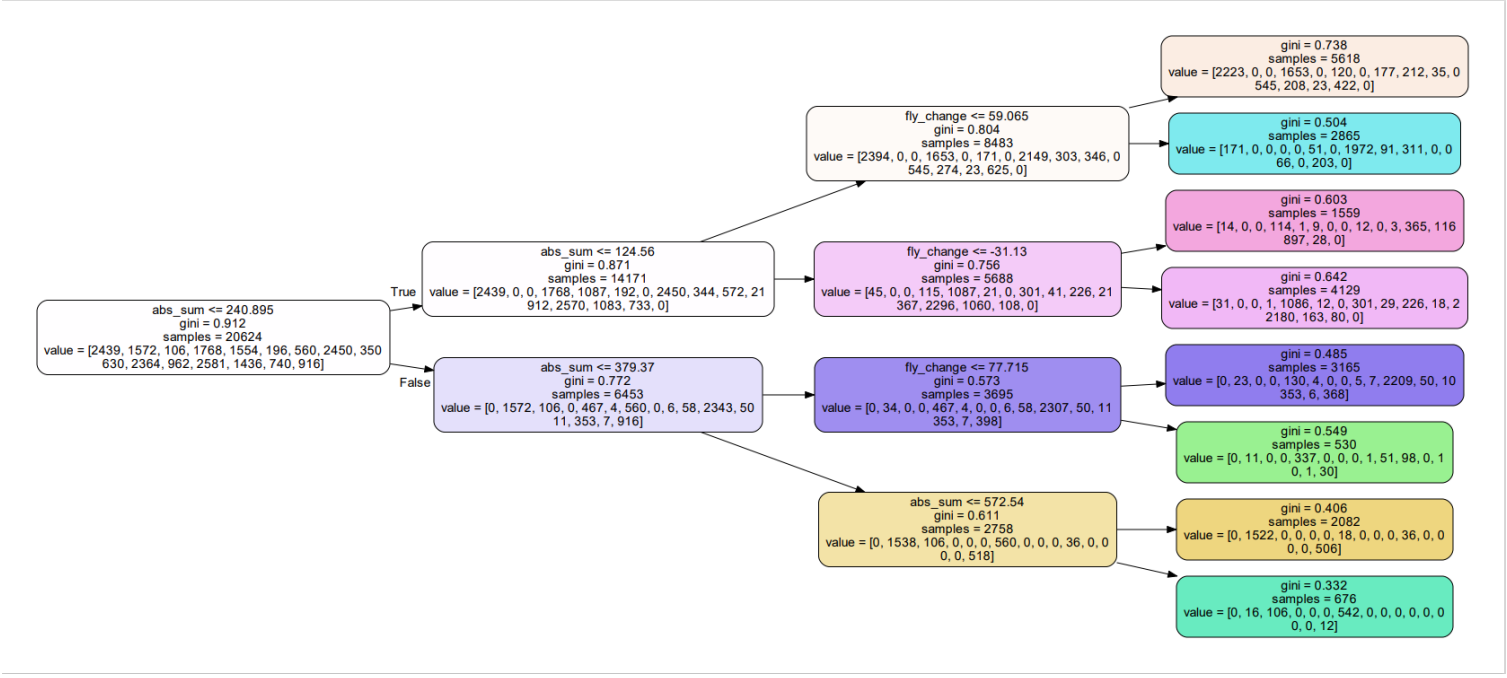

图 5.4 下行流量随机森林分类图

由上图可以看出，基站下行流量数据的特征分叉结果和上行流量数据有明显的区别，在根节点和第一层节点上都是一阶绝对差分 abs_sum 和的基尼不纯度最高，分别为 0.912，0.871 和 0.772，因此根节点和首层节点都根据特征 abs_sum 进行分叉；在第二层，傅里叶变换系数的基尼不纯度最高，这和上行流量结果类似。

随机森林的决策树根据基尼系数对样本进行切分，但是根节点基尼不纯度越高并不代表该特征在整体样本分类中发挥了最大的作用，因此还需要计算特征重要性。随即深林中某个特征 A 的特征重要性计算方法如下：对于随机森林中的每一刻决策树，使用相应的 OOB 袋外数据来计算它的袋外数据误差，记为$errOOB_1$。然后随机地对袋外数据 OOB 所有样本的特征 A 加入噪声干扰，再次计算它的袋外数据误差,记为$errOOB_2$。假设随机森林中有 N 棵决策树，那么对于特征 A 的重要性就可以表示为：

$$Feature\_impor = \sum \frac{errOOB_2 - errOOB_1}{N} \qquad (5.6)$$

当给某个特征随机加入噪声之后，若袋外的准确率大幅度降低，则说明这个特征对于样本的分类结果影响很大，也就是说它的重要程度比较高。本文通过以上方法计算得到小区基站流量的特征重要性排序如下图 5.5 所示。从小区基站的上行流量特征重要性排序可以看出，一阶差分绝对和 abs_sum 和 adf 检验统计量的分类特征重要性要远远大于其他变量，均值、标准差、偏度和峰度等传统统计量分列 2 到 6 名，其余变量的特征重要性较小，总体差距不大。由下图 5.6 可知，基站下行流量的特征重要性排序和上行流量的分布差别不大，都是一阶差分绝对和以及 adf 检验统计量远高于其他特征；而均值、标准差、偏度和峰度等统计量紧随其后。通过特征重要性的排序，我们可以识别出对小区分类贡献较大的特征，排除部分贡献较小的特征，构造更有效的特征，提高后续机器学习算法进行小区流量数据聚类的学习效率，能够获得更好的分类准确性。

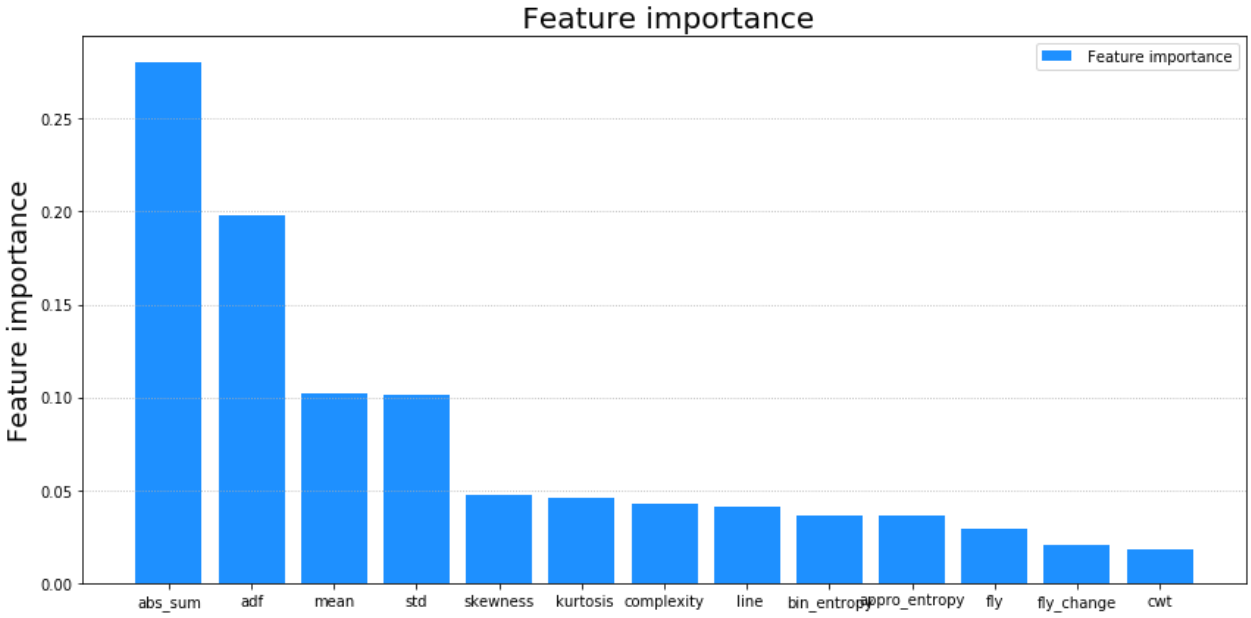

图 5.5 基站上行流量特征重要性排序

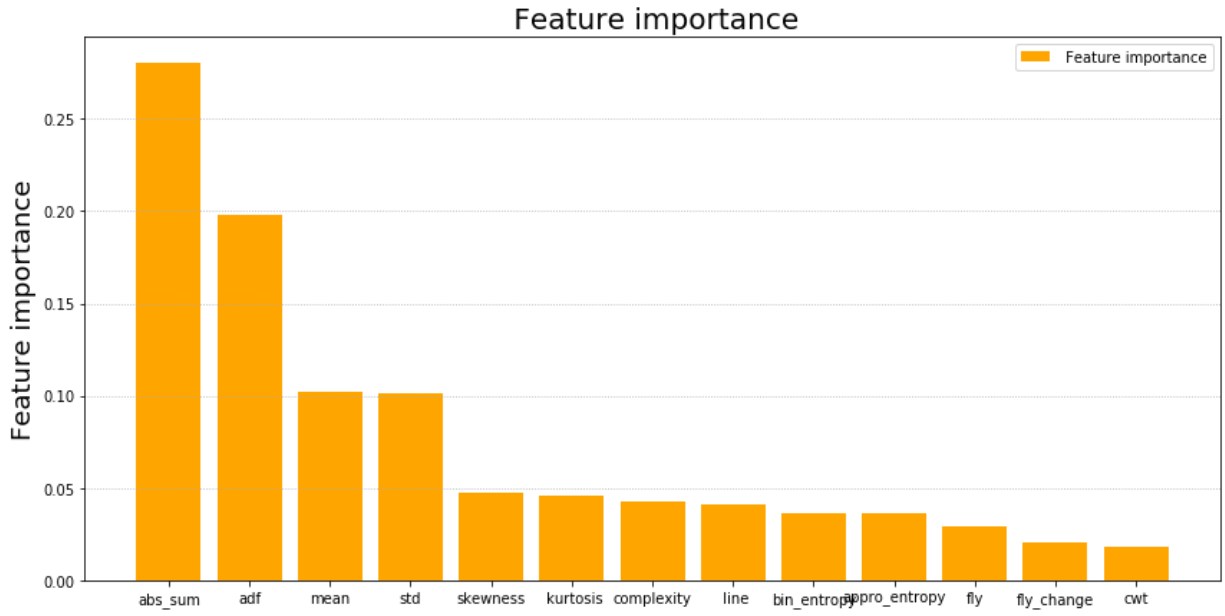

图 5.6 基站下行流量特征重要性排序

由上图可知，基站下行流量的特征重要性排序和上行流量的分布差别不大，都是一阶差分绝对和以及 adf 检验统计量远高于其他特征；而均值、标准差、偏度和峰度等统计量紧随其后。通过特征重要性的排序，我们可以识别出对小区分类贡献较大的特征，排除部分贡献较小的特征，构造更有效的特征，提高后续机器学习算法进行小区流量数据聚类的学习效率，能够获得更好的分类准确性。

5.4 基于特征的聚类结果分析

基于上述分析，本文采取的聚类分析的总体思路如下图所示：

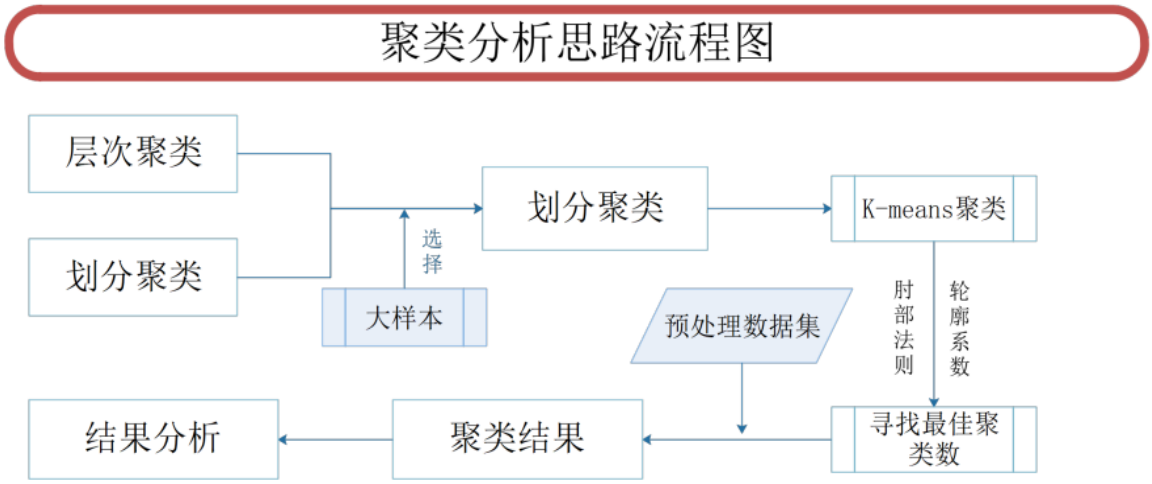

图 5.7 聚类分析思路流程图

5.4.1 最佳聚类数的确定

对小区基站流量数据进行聚类分析首先需要确定聚类的数目，本文根据 elbow 手肘法则和轮廓系数（Silhouette Coefficient）来确定时间序列聚类的最佳聚类数。

（1）首先介绍 elbow 法则的基本原理，根据时间序列聚类算法的核心思想，被聚为某一类的样本应该离该类的聚类中心比较接近。因此引入误差平方和来衡量某时间序列距离其聚类中心的远近：

$$SSE = \sum_{i=1}^{k} \sum_{X \in C_i} |X - m_i|^2 \qquad (5.7)$$

其中 C_i 是第 i 个时间序列簇，X 是 C_i 中的时间序列样本，m_i 是 C_i 的聚类中心（C_i 中所有样本的均值），SSE 是所有样本的聚类误差，代表了聚类效果的好坏。如果一个聚类结果是好的，那么对于每一类来说，该类中的时间序列样本距离该类的聚类中心的距离之和应该比较小，把所有类别的时间序列样本和其聚类中心距离加总就得到 SSE，SSE 越小，说明聚类的结果就越好。

Elbow 手肘法则的中心思想是：随着聚类数 k 的增大，样本划分会更加精细，每个簇的聚合程度会逐渐提高，那么误差平方和 SSE 自然会逐渐变小。并且，当 k 小于真实聚类数时，由于 k 的增大会大幅增加每个簇的聚合程度，故 SSE 的下降幅度会很大，而当 k 到达真实聚类数时，再增加 k 所得到的聚合程度回报会迅速变小，所以 SSE 的下降幅度会骤减，然后随着 k 值的继续增大而趋于平缓，也就是说 SSE 和 k 的关系图是一个手肘的形状，而这个肘部对应的 k 值就是数据的真实聚类数。如下图所示，当聚类数 K 小于 3 时，SSE 下降迅速；当聚类数大于 3 时，SSE 下降速度明显放缓，因此根据 elbow

手肘法则，当聚类数为 3 时即为最佳聚类数。

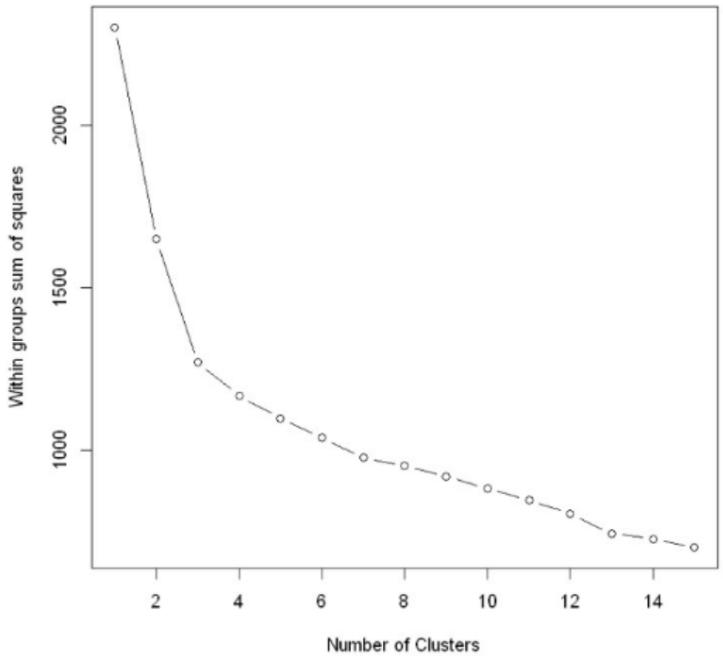

图 5.8 elbow 法则示意图

（2）接下来介绍轮廓系数法则，轮廓系数（Silhouette Coefficient）是另一种评价聚类结果好坏的指标，它最早由 Peter J. Rousseeuw 在 1986 提出。它结合了内聚度和分离度两种因素，可以在相同原始数据的基础上评价不同算法、或者算法不同运行方式对聚类结果所产生的影响。轮廓系数可以定义为：

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \qquad (5.8)$$

其中 a(i)为时间序列到同簇的其他时间序列的平均距离，a(i)越小，说明样本 i 越应该被聚类到该簇，因此将 a(i)称为时间序列样本 i 的簇内不相似度。而 b(i) 为时间序列

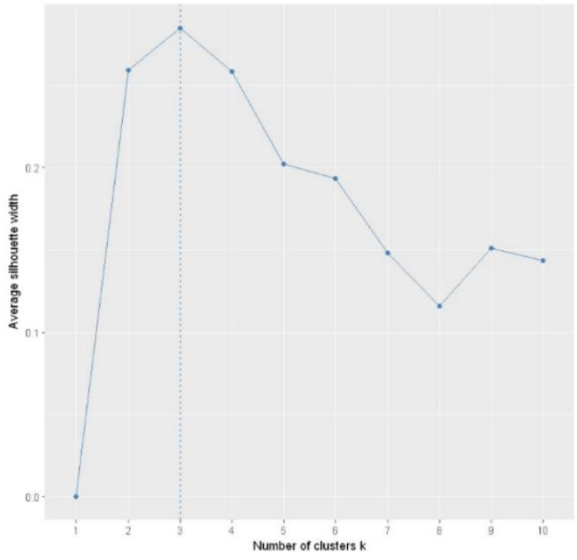

图 5.9 轮廓系数法则示意图

样本 i 到其他某簇 C_j 的所有样本的平均距离 b_{ij}，称为时间序列样本 i 与簇 C_j 的不相似度。那么时间序列样本 i 的簇间不相似度可以定义为：$b(i)=\min\{b_{i1},b_{i2},...,b_{ik}\}$，bi 越大，说明样本 i 越不属于其他簇。

求出所有样本的轮廓系数后再求平均值就得到了平均轮廓系数。平均轮廓系数的取值范围为[-1,1]，且簇内样本的距离越近，簇间样本距离越远，平均轮廓系数越大，聚类效果越好。那么，很自然地，平均轮廓系数最大的 k 便是最佳聚类数。如上图所示，当 k=3 时就是最佳聚类数。

下面使用 elbow 法则和轮廓系数法则，对本题的上行流量数据和下行流量数据进行最优聚类系数寻找，使用 python 程序遍历 2 至 30 的时间序列聚类簇数，得到如下的结果：

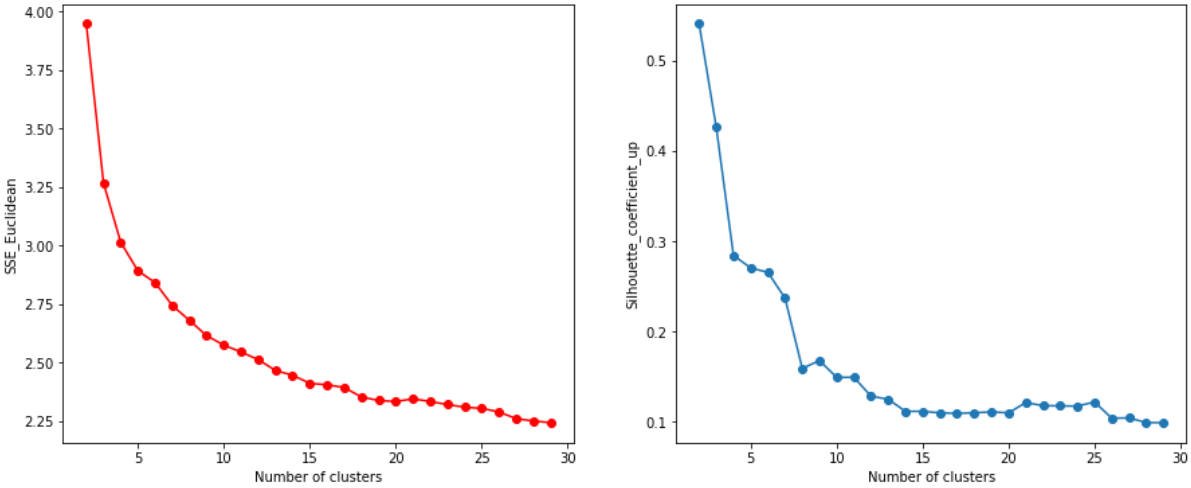

图 5.10 上行流量聚类的 SSE 和轮廓系数

表 5.4 上行流量聚类 SSE 和轮廓系数

聚类数	2	3	4	5	6	7	8
SSE	3.947039	3.267039	3.013997	2.891366	2.841355	2.743168	2.679369
轮廓系数	0.540256	0.425645	0.284065	0.270308	0.265356	0.236806	0.159058
聚类数	9	10	11	12	13	14	15
SSE	2.614142	2.573782	2.544621	2.511912	2.465644	2.444944	2.410573
轮廓系数	0.167815	0.149173	0.149367	0.128372	0.125	0.111653	0.111672
聚类数	16	17	18	19	20	21	22
SSE	2.404465	2.39262	2.352285	2.337288	2.332502	2.344757	2.332642
轮廓系数	0.109939	0.109317	0.109984	0.111036	0.109934	0.121367	0.117905
聚类数	23	24	25	26	27	28	29
SSE	2.320086	2.308597	2.303576	2.288312	2.259595	2.249561	2.242368
轮廓系数	0.117656	0.117438	0.12203	0.104009	0.104384	0.099215	0.099093

结合上图和上表可以看出，SSE 下降得较为平滑，没有表现出特别明显的 elbow 点。相对来说，当聚类数为 16 或者 18 时，SSE 下降速度的减缓幅度较大，因此根据手肘法则可以把聚类数 16 和 18 作为最佳聚类系数的备选择点。再观察轮廓系数图，在遍历聚类系数时，轮廓系数基本上呈现随着聚类系数上升而下降的趋势，当聚类数为 2 时，轮

廓系数达到最大，但是这和使用 elbow 手肘法则得出的结论明显是互相矛盾的。究其原因，这是因为轮廓系数考虑了簇间不相似度 b(i)，也就是样本与最近簇中所有样本的平均距离。从定义上看，轮廓系数大，不一定是簇内不相似度 a(i)（样本与同簇的其他样本的平均距离）小，而可能是 b(i)和 a(i)都很大的情况下 b(i)相对 a(i)大得多，由此导致 a(i)是有可能取得比较大的。当簇内不相似度 a(i)较大时，样本与同簇的其他样本的平均距离就大，簇的紧凑程度就弱，那么簇内样本离质心的距离也大，从而导致 SSE 较大。

因此，在选择最佳聚类系数时，需要优先考虑 SSE 的变化情况。另一方面，注意到当聚类数遍历到 16 至 18 的阶段时，轮廓系数的下降幅度不大，基本维持在同一水平；而一旦超过 25 则会大幅下降。因此综合考虑 elbow 法则和轮廓系数法则，最终确定 16 为小区上行流量数据的最佳聚类数，18 作为备选聚类数。

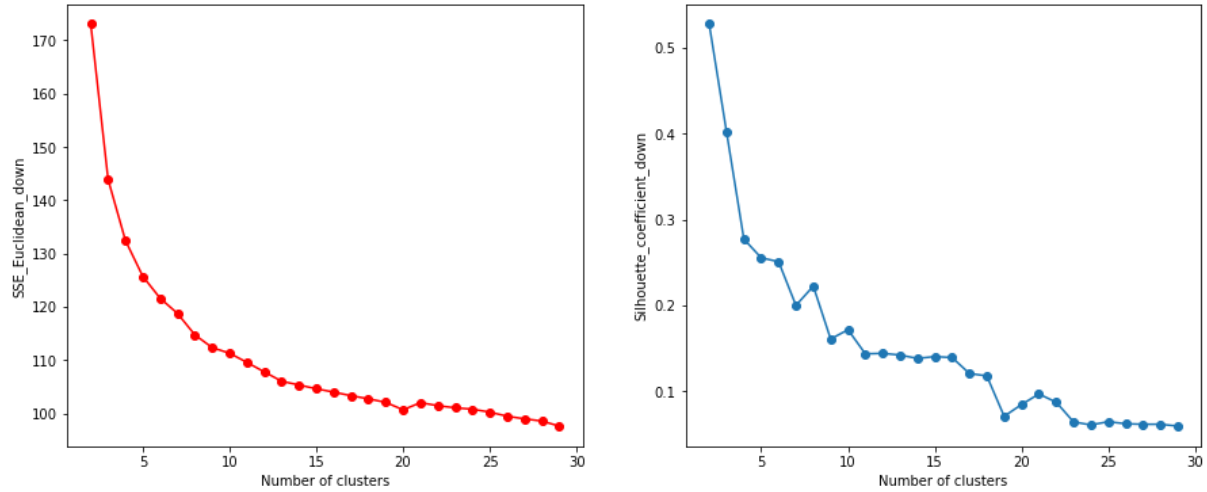

图 5.11 下行流量聚类的 SSE 和轮廓系数

表 5.5 下行流量聚类的 SSE 和轮廓系数

聚类数	2	3	4	5	6	7	8
SSE	173.0398	143.8722	132.4202	125.7008	121.5906	118.7341	114.7197
轮廓系数	0.52755	0.401989	0.277101	0.255347	0.25086	0.19995	0.222341
聚类数	9	10	11	12	13	14	15
SSE	112.3283	111.2762	109.556	107.7724	106.0182	105.3168	104.604
轮廓系数	0.160781	0.171613	0.143531	0.144088	0.14212	0.13818	0.140313
聚类数	16	17	18	19	20	21	22
SSE	103.9805	103.3006	102.7224	102.0537	100.6545	101.9966	101.4338
轮廓系数	0.139183	0.120636	0.117838	0.070791	0.084464	0.096794	0.08717
聚类数	23	24	25	26	27	28	29
SSE	101.0301	100.7696	100.216	99.4523	98.95901	98.543	97.59735
轮廓系数	0.064258	0.060711	0.064546	0.062101	0.061488	0.061558	0.059366

结合上图和上表可以发现，下行流量聚类的 SSE 的数值相比上行流量聚类要大得多，但是在变化趋势上是相类似的。可以考虑 16 或者 18 作为最佳聚类数。

5.4.2 聚类结果分析

根据前文分析，选定最佳聚类数和 kmeans 聚类的方法后，基于 python 对特征向量

集进行聚类运算，首先得到小区的各特征之间的相关关系如下两图所示：

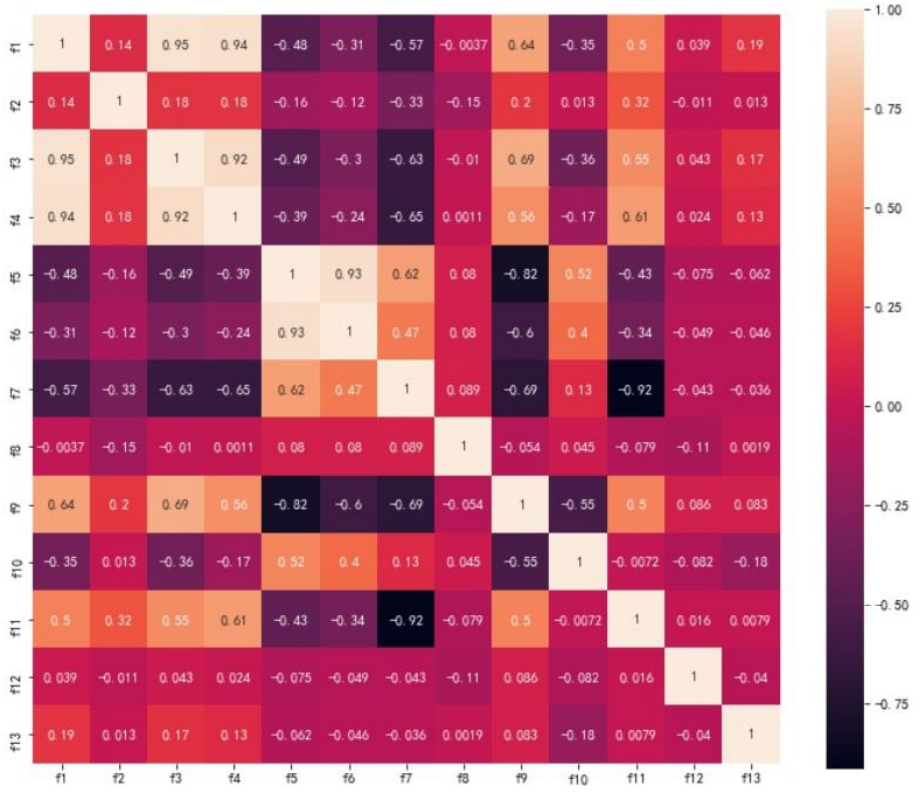

图 5.12 上行流量各特征相关性强弱图

图 5.13 下行流量各特征相关性强弱图

上述两图即为各特征之间的相关性关联图，对脚线上的元素代表自相关，其他位置上的元素代表互相相关，且不同的颜色代表着不同强弱的相关性，颜色越淡代表正向相关性越强，颜色越深代表负向相关性越强。

同时，由前文分析可知最佳聚类数为 16，基于 python 获得的部分聚类结果如下表所示，详细分类结果见附件 2，聚类中心结果详见附件 3。

表 5.6 样本小区集聚类结果情况统计表

聚类标签	小区编号	聚类标签	小区编号
0	658、694、867、1531、1614、2352…..	8	2545、3972…..
1	556、1064、1613、2018、2264….	9	918、1834、5373、5636…..
2	95、1336….	10	349 、546、672、697….
3	884、933、954、1874、2245、2689…..	11	338、2396、2701、3075、3258…
4	221、376、781、1305、2140、2593….	12	186、1663、1767、1933、2031….
5	7820、189、354、460、482、720…	13	495、586、616、728….
6	6906、8366….	14	2392、3141、3379、3711、6631….
7	420、833、1078、1100、1704、1997…..	15	583、1187、1469、2780….

从下图可知，各类别小区在样本小区集中的数量分布情况。其中，数量最多的前四类小区分别为第 12、0、3、4 类小区，数量占比总计达 50%左右，且第 12 类小区占比最大。同时，第 6 和 8 类的小区占比份额显著较小，都在 1%以下。此外，其他类别的小区份额占比集中在 3%到 8%之间。

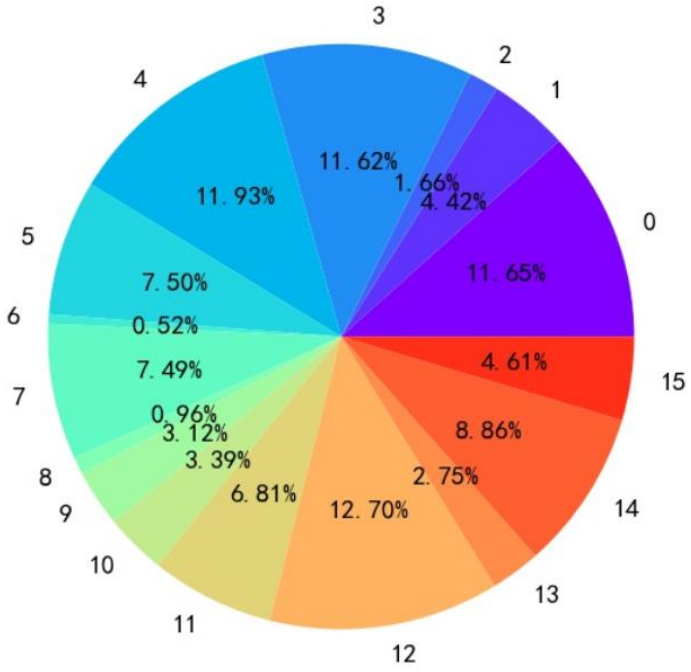

图 5.14 各类别小区数量分布扇形图

（3）聚类结果可视化

基于前文随机森林法关于特征重要性的分析，在 13 个特征中选取前 6 个最为重要的特征对各个类别的小区的特点进行分类对比。基于 python 对聚类结果的可视化实现

得到如下两个图所示:

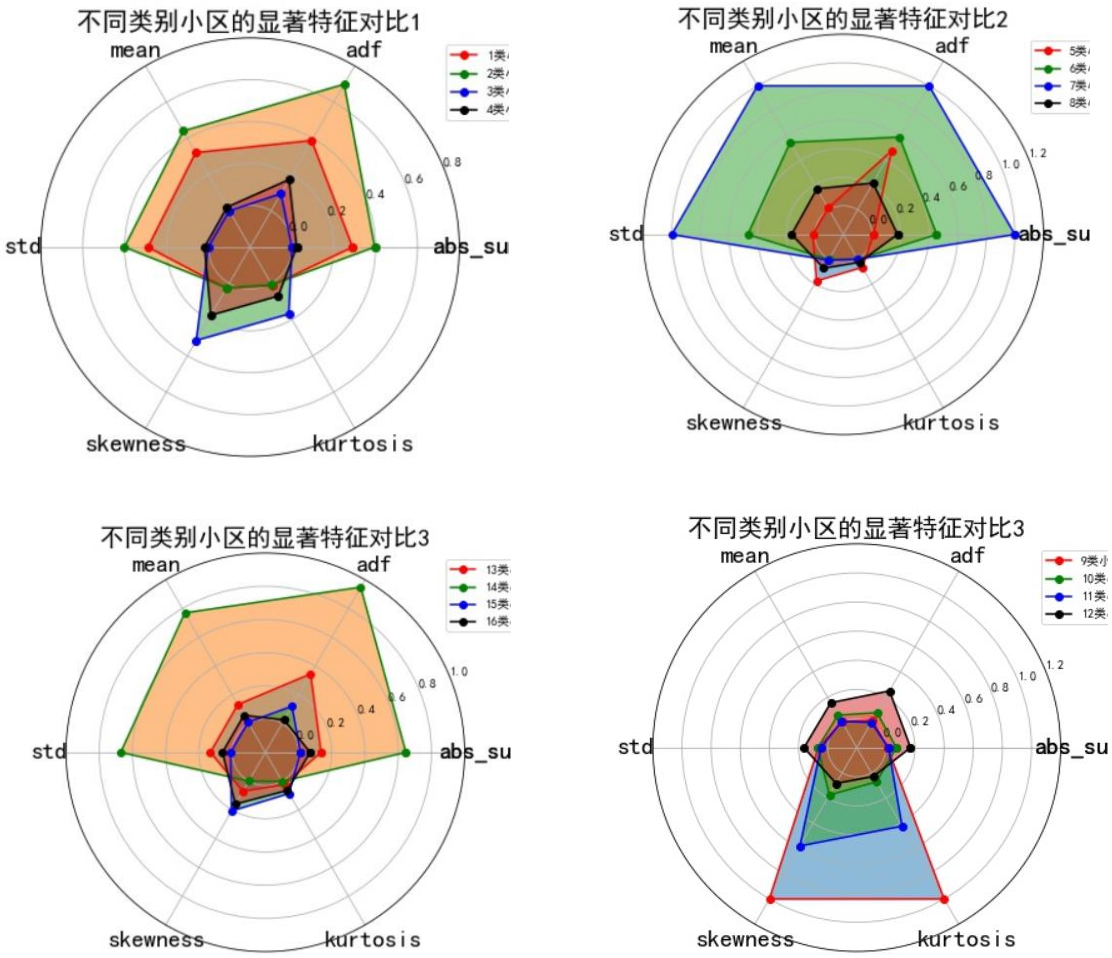

图 5.15 各类小区上行流量显著特征对比情况

基于雷达图可以清楚的认识到各个类别小区的特点：

从上、下行流量来看，第 2、7、6、14 类小区在 mean、adf、std 和 abs_sum 四个特征的相对值都明显大于其他对比小区，流量数据的波动性明显强于其他对比小区。说明这些类别的小区所在区域的基站服务人员具有较大的流动性。因此这类基站小区所处的业务场景可能是商场、步行街、火车站等。

而第 4，8，12，16，15 类小区 mean、adf、std 和 abs_sum 四个特征的相对值都较对比小区相对较小。说明基站流量的波动性相对较弱，这类基站小区所处的业务场景可能是学校、住宅小区或者写字楼等。

而对于第 3，9 类和 11 类小区来说可以看到相对其他对比的小区，偏度和峰度的值明显较大。偏度其描述的是某总体取值分布的对称性，偏度较大说明这些小区的基站流量数据呈现出较大的非对称性，说明此处基站的流量使用较为集中在一段时间。同时，峰度是描述总体中所有取值分布形态陡缓程度的统计量，峰度的绝对值数值越大表示其分布形态的陡缓程度与正态分布的差异程度越大，即数据较为集中。因此，对于这类小区可以采取按时间段设置开关基站来达到节能的目的。

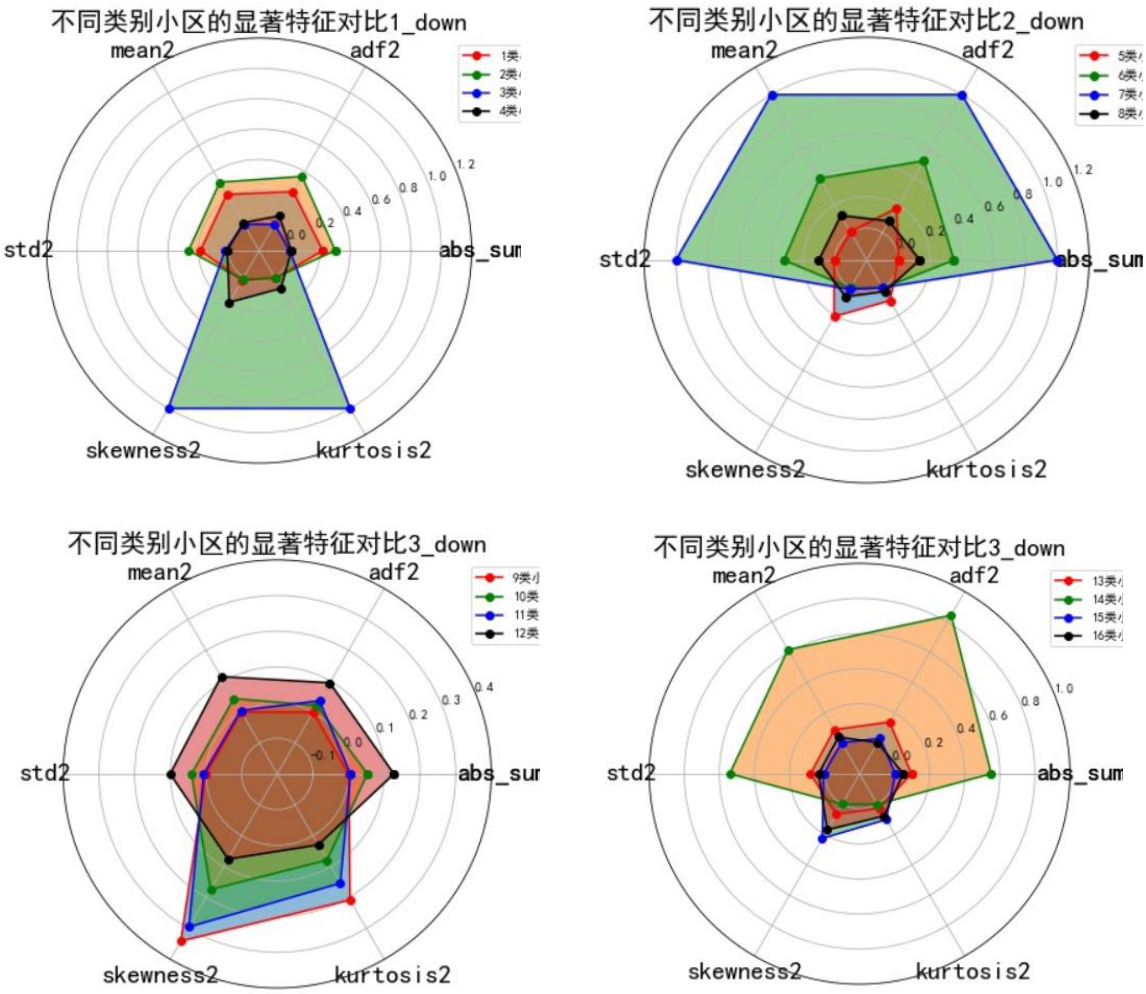

图 5.16 各类小区下行流量显著特征对比情况

而对于第 3，9 类和 11 类小区来说可以看到相对其他对比的小区，偏度和峰度的值明显较大。偏度其描述的是某总体取值分布的对称性，偏度较大说明这些小区的基站流量数据呈现出较大的非对称性，说明此处基站的流量使用较为集中在一段时间。同时，峰度是描述总体中所有取值分布形态陡缓程度的统计量，峰度的绝对值数值越大表示其分布形态的陡缓程度与正态分布的差异程度越大，即数据较为集中。因此，对于这类小区可以采取按时间段设置开关基站来达到节能的目的。

此外，对于第 1、5、10、13 类小区，从上、下行流量数据情况来看。总体上，对于各个特征，上述小区的各类特征分布都较为均衡，且各特征数值相对较小。说明这些基站小区所在位置的人员数量较少且人员具有的固定性，因此这类小区可能是工业园区、党政军机关、乡镇等。

5.5 基于近邻算法对其余样本进行划分

通过上面的聚类过程，我们得到了小区基站上下流量的 16 个类别。碍于本小组的计算机算力有限，本文的聚类结果是在附件一的原始数据中抽出 30000 个小区进行的，因此在完成聚类之后需要把剩余样本归类到这 16 类当中。具体过程是首先计算出基站流量的 16 类各类别的聚类中心，然后通过 DTW（Dynamic Time Warping）动态时间规整算法计算每个剩余小区基站流量样本和每个聚类中心的距离，最后通过改进版的 KNN 近邻算法构建分类器基于样本与聚类中心距离进行分类。

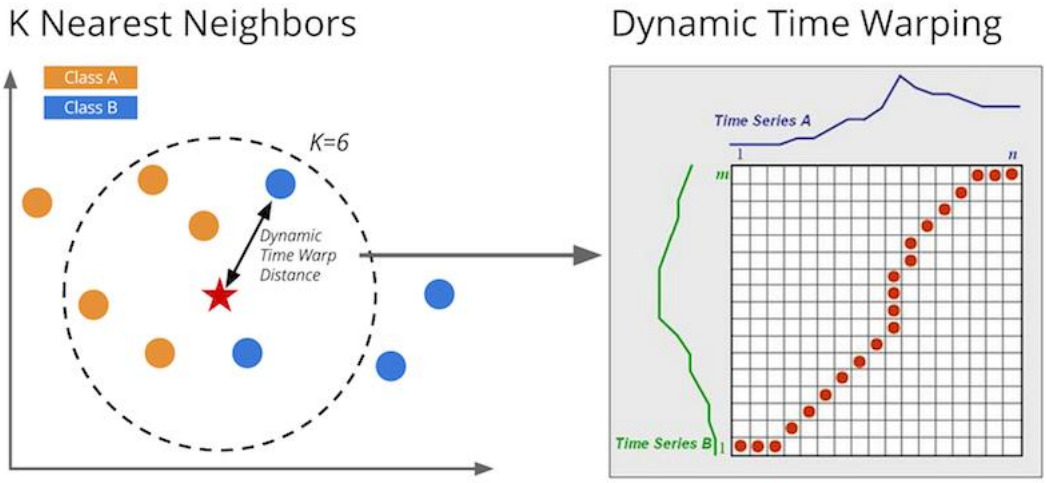

图 5.17 动态时间规整和 KNN 算法示意图

5.5.1 使用 DTW 算法计算样本距离

本文要对剩余小区进行分类，就要对剩余小区的时间序列与已完成聚类的小区进行相似性的度量，经典时间序列相似性的度量方法被分为两种锁步度量（lock-step measures）

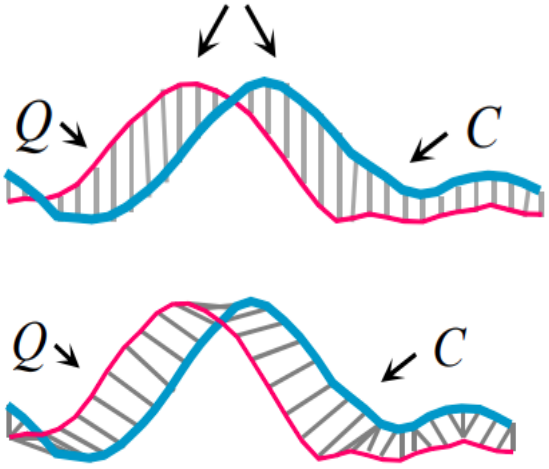

图 5.18 锁步度量和弹性度量示意图[8]

和弹性度量（elastic measures）。形象地来说，锁步度量是要对时间序列进行“一对一”的比较，而弹性度量则允许时间序列进行“一对多”的比较[9]。最一般的相似度计算方法是对两个时间序列计算欧氏距离，这是一种锁步度量的方式。假设有两个时间序列，Q 和 C，如果直接用欧氏距离计算相似度的话，就会存在时间步不对齐，序列长短不一等问题，如下图所示：

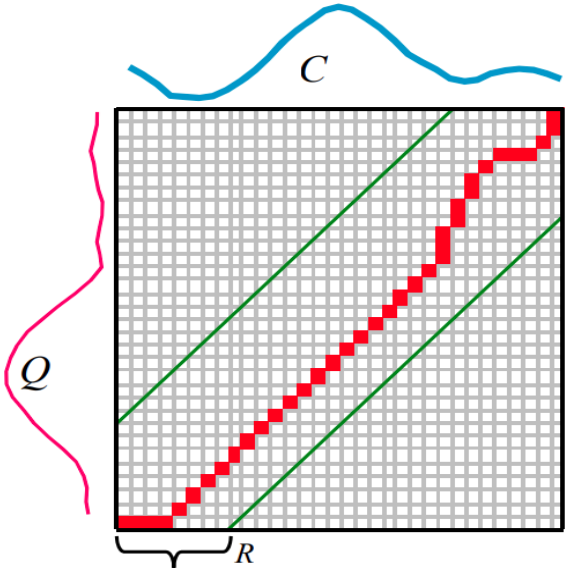

图 5.19 DTW Warping 路径[10]

可以看出 Q 和 C 两个时间序列是相似的，但是长短和相位有差异。如果序列长短不一，或时间步不对齐的时候，欧氏距离是无法有效计算两个时间序列的距离，特别是在峰值的时候。上图的第二部分则是 DTW 算法，首先将其中一个序列进行线性放缩进行某种“扭曲”操作，以达到更好的对齐效果，可以存在一对多匹配的情况，适用于复杂时间序列，属于弹性度量。

动态时间规整在 60 年代由日本学者 Itakura[11]提出，用于衡量两个长度不同的时间序列的相似度。把未知量伸长或缩短(压扩)，直到与参考模板的长度一致，在这一过程中，未知序列会产生扭曲或弯折，以便其特征量与标准模式对应。首先假设存在两个时间序列 Q 和 C，它们长度分别为 n 和 m，即 $Q=[q_1,q_2,...q_n]$ $C=[c_1,c_2,...c_m]$。用一个 n*m 矩阵来比对两个序列，Warping 路径会穿越这个矩阵，Warping 路径的第 k 个元素表示为 $w_k=(i,j)_k$，横纵代表两个序列对齐的点。

上述 Warping 路径的约束条件有三个：（1）边界条件：$w_1=(1,1)$ 和 $w_k=(n,m)$；（2）连续性：如果 $w_k=(a,b)$ 且 $w_{k-1}=(a',b')$，则必须满足 $a-a'\le 1$ 和 $b-b'\le 1$；（3）单调性：如果 $w_{k-1}=(a',b')$ 且 $w_k=(a,b)$，则必须满足 $a-a'\ge 0$ 和 $b-b'\ge 0$。满足以上三个条件的 Warping 路径并不唯一，因此需要找出最优的 Warping 路径是关键：

$$DTW(Q,C)=\min\left\{\frac{\sqrt{\sum_{k=1}^{K}w_k}}{K}\right\} \qquad (5.9)$$

主要的优化方法有平方距离优化方法、Lower Boundin 方法、Early Abandonin 方法和 Recording Early Abandoning 方法。本文所使用方法为 Lower Boundin 方法，它的主要思路是先通过计算 LB（lower bounding）处理掉不可能是最有匹配序列的序列，经过逐次筛选得到最优路径，计算 LB 的主要有 LB_Kim 和 LB_keogh 等方法，本文所使用的 LB_keogh 函数的计算方法如下：

$$LB\_keogh(Q,C)=\sum_{i=1}^{n}\begin{cases}(q_i-u_i)^2, q_i>u_i\\(q_i-l_i)^2, q_i<l_i\\0, other\end{cases} \quad (5.10)$$

5.5.2 使用改进的 KNN 算法对剩余小区分类

由上一节的 DTW 算法可以计算出剩余小区样本和已聚类的小区样本之间的距离矩阵，再通过 KNN 算法可以找出和每一个待分类小区最相似的 K 个已分类样本，最后通过这些样本的逆距离加权投票的方式决定该待分类样本属于哪一类。该过程可以用下面的公式来表示：假设已分类的小区样本有 N 个，它们分别属于 16 个分类$\omega_i, i=1,2,\cdots,16$考察待分类小区样本 x 在这些样本中的 k 个近邻，其中k_i个属于ω_i类，则ω_i类的判别函数可以表示为：

$$g_i(x)=\sum_{x_i\in w_i, x_i\in A}\frac{1}{d_{x_i}}, i=1,2,...16 \quad (5.11)$$

其中，A 表示待分类小区样本的 k 近邻样本集合，d_{X_i}表示由上面 DTW 算法计算得到的已分类小区样本X_i到待分类小区样本点的距离。则 KNN 算法的决策规则可以表示为：

$$if: g_a(x)=\max_{i=1,...16}\{g_i(x)\} then: x\in w_a \quad (5.12)$$

由于上述的传统 KNN 算法需要计算出每个待分类样本和所有已分类小区样本的 DTW 距离，计算量比较大。而且有可能有部分已分类样本和聚类中心偏离较大，再给

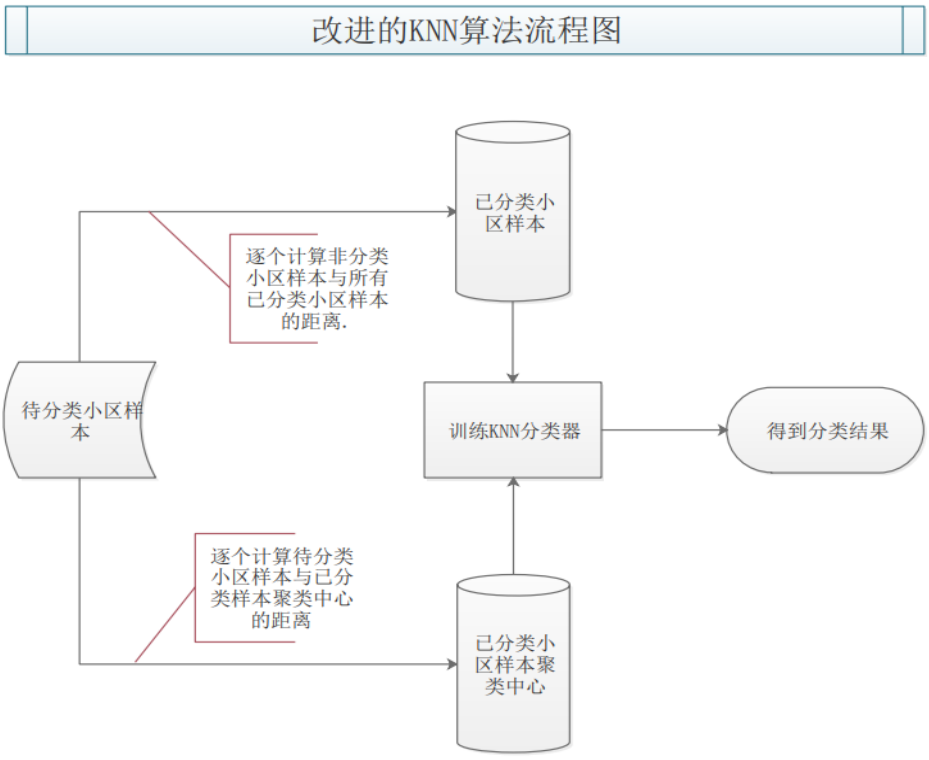

图 5.20 改进 KNN 算法与传统 KNN 算法对比示意图

待分类样本进行学习有可能得不到较好的分类结果。因此文本创新性地使用已分类样本的聚类中心作为训练集给 KNN 分类器进行学习，大幅度减少 KNN 分类器的训练时间。再通过直接计算待分类小区样本和 16 个聚类中心的 DTW 距离，提高待分类小区的分类准确性。改进 KNN 算法流程图如下所示：

通过改进的 KNN 算法，本文得到剩余待分类小区的所有分类标签，下面给出部分小区的聚类中心，以及被 KNN 分类器分到该类小区的流量时间序列图：

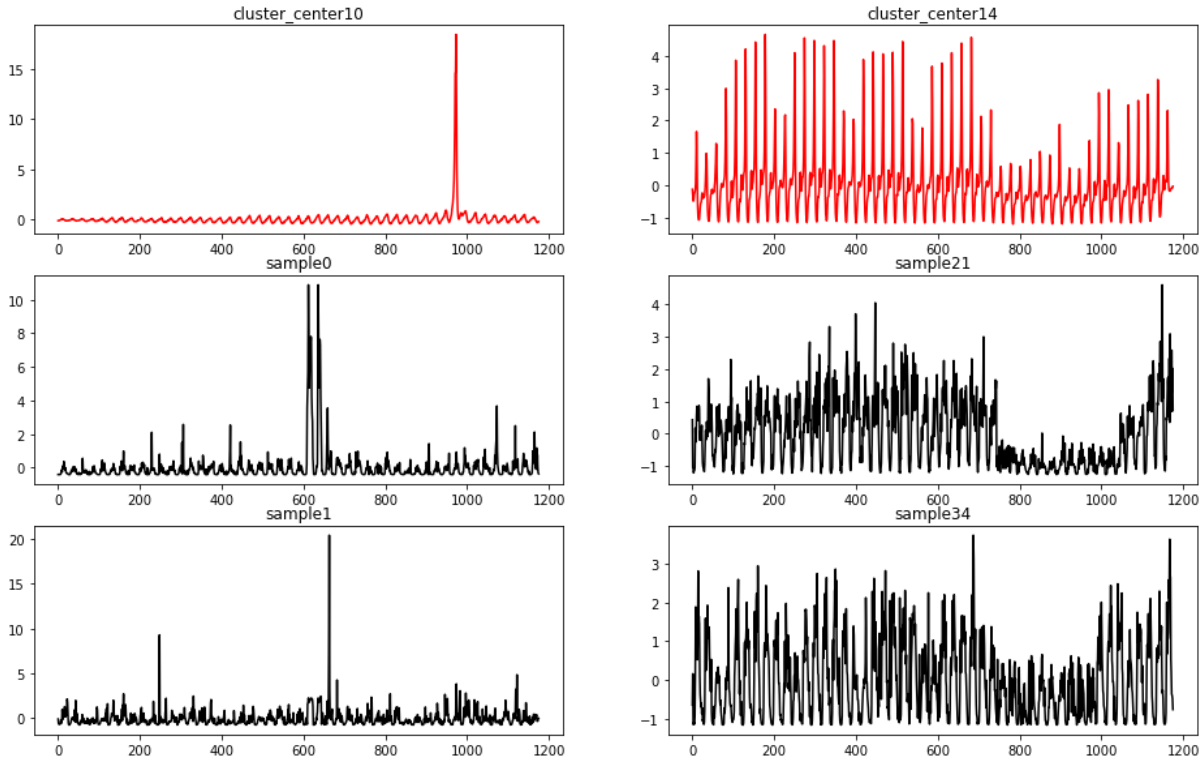

图 5.21　第 10 类和第 14 类小区 KNN 分类效果示意图

上图的左边区域展示了第 10 类小区聚类中心以及待分类样本中被分到第 10 类小区的 2 个样本的时间序列图。可以发现 KNN 分类准确地捕捉到了时间序列的特点，待分类小区中的 sample0 和 sample1 经过 DTW 矩阵的计算，得到和 16 个聚类中心的距离，经过 KNN 分类器的比对，最终被分类到相似性最高的 cluster_center10 所在类别当中。该类小区基站流量的特点是大部分时点都是处于较低的平稳状态，伴随着明显日周期性，但是在某个会出现远远高于平时的异常值。这部分小区的流量阈值设置较为困难，因为异常值出现缺乏规律性，而且次数非常少，调节流量阈值往往来不及，防不胜防。

上图的右边区域展示了第 14 类小区聚类中心以及待分类样本中被分到第 14 类小区的 2 个样本的时间序列图。可以发现这一类小区和第 10 类小区的特点完全不同，该类小区流量的时间序列展现出明显的工作日与周末的区别，具有显著的以一星期为周期的特点。且工作日的流量上限要显著大于周末，因此这类小区的场所应该在写字楼、中小学校或者工厂等区域。而且经过观察发现，在第 750 个小时到第 1000 小时的区间内，该类小区的流量出现了长期的异常情况。此时段的工作日和周末的区别不明显，且流量上限要显著低于大部分正常时点。因此推测该时间段应该是放长假时期造成的。

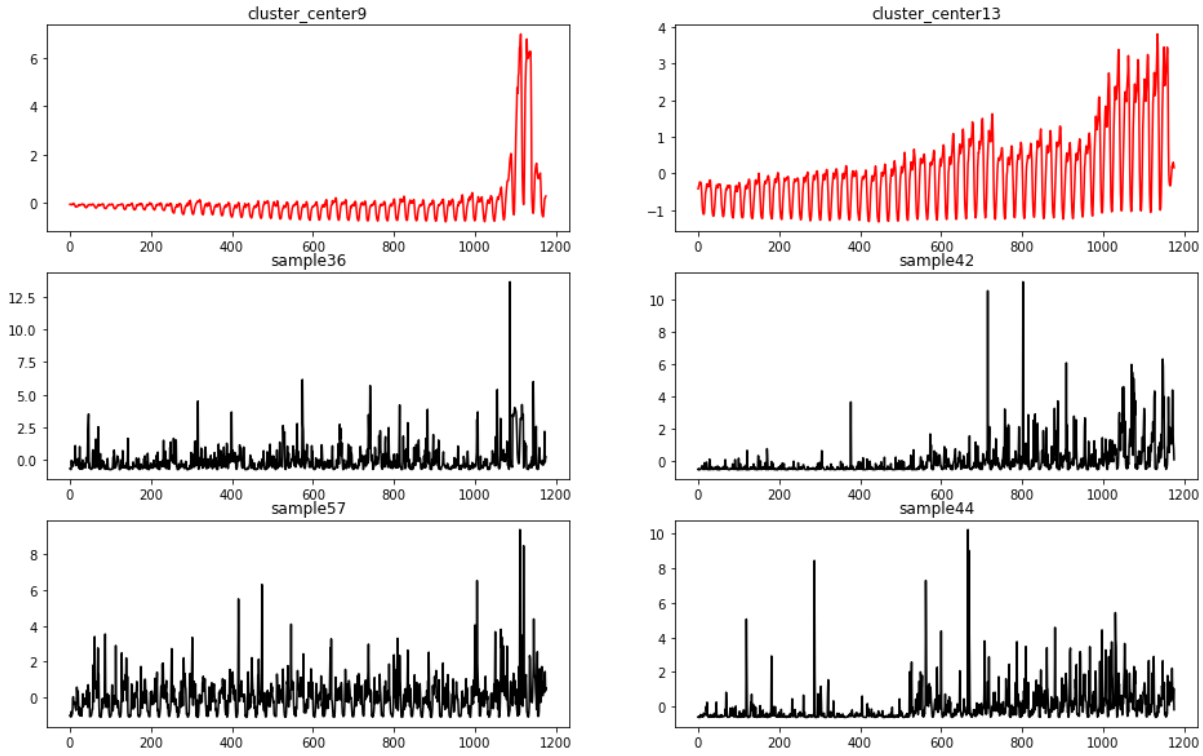

图 5.22 第 9 类和第 13 类小区 KNN 分类效果示意图

上图左边区域显示了第 9 类小区的聚类中心以及待分类样本中被分到第 9 类小区的 2 个样本的时间序列图。由图可知，第 9 类小区的大部分时点都处于较为平稳状态，但是在样本时间窗口末端出现了几天的异常值，出现异常值的时候小区流量要显著大于平时。

上图的右边区域显示了第 13 类小区的聚类中心以及待分类小区样本中被分到第 13 类的 2 个样本的时间序列图。由图可知，此类小区的流量上限在前期大部分处于较低的水平状态，随着时间往后，该类小区的流量逐渐增大。这类小区应该处于新开发地区，用户的需求正在被逐渐开发出来，流量的阈值会呈现出长期增长的趋势。

由于本文篇幅有限，不再对其余的分类结果做展示。KNN 分类器所得到的所有分类标签将会在附件四中给出。

六、问题二的模型建立与求解

6.1 数据处理与聚类

阈值设定前，首先按照问题一中的数据处理方法对原始数据进行处理，然后根据问题二的需要共使用了两种聚类方法：Kshape 聚类方法和 Kmeans 聚类方法。考虑到不同基站上下行流量在时序波动特征上的差异，本文首先使用 Kshape 聚类方法将流量数据进行分类，得到聚类结果 I，以便在使用 LSTM 模型对上下行流量期望进行预测时得到的更好的预测结果。此外，考虑到在不同时点上流量分布特征上的异质性，本文又根据上下行流量的分布特征，使用 Kmeans 聚类方法聚类得到聚类结果 II，该种处理方法是为了在截面维度上将分布特征差异较大的小区分离开来，以便得到较好的分位数计算结果。

6.1.1Kmeans 聚类

下面介绍时间序列的 Kmeans 聚类方法。假设ξ={X_i|$X_i \in R^m$,i=1,2,...,n}为给定的小区流量数据集，则每个时间序列所在位置点可以用 d 维表示$x_i=(x_{i1},x_{i2},...,x_{id})$，$C=\{C_1,C_2,C_3,...,C_k\}$表示总样本集中小区流量数据的 k 个类别，其中每个类别的时间序列簇质心点由C_j表示，j=1,2,…k。假设时间序列$X_i=(X_{i1},X_{i2},...,X_{id})$和时间序列$X_j=(X_{j1},X_{j2},...,X_{jd})$分别代表两个小区的流量时间序列，那么它们之间的欧式距离为：

$$d(X_i,X_j)=\sqrt{\sum_{m=1}^{d}(X_{im}-X_{jm})^2} \tag{6.1}$$

则某一类小区基站的质心可以定义为：

$$C_j=\frac{1}{n_j}\sum_{X_i\in n_j}X_i \tag{6.2}$$

时间序列 k-means 算法是一种基于样本间相似性度量的间接聚类方法，也被称为K均值算法。算法的主要思想是通过迭代的过程把数据集划分为不同的类别，使得评价聚类性能的准则函数达到最优。算法描述如下：

输入：簇的个数 K 与时间序列集合 ξ。

输出：K 个簇，每个小区基站时间序列所属类别标签。

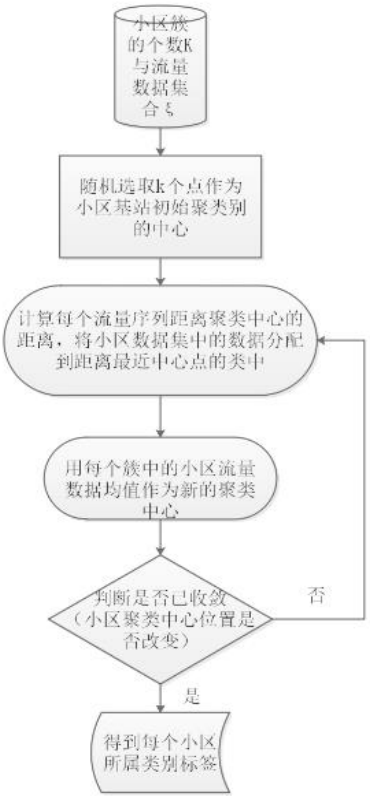

图 6.1 Kmeans 聚类流程图

6.1.1Kshape 聚类

KShape 聚类是一种基于形状计算距离的聚类方法，它由 John Paparrizos 和 Luis Gravano 于 2015 年首次提出。KShape 方法可以有效解决量纲不同、映射不变性以及序列相似但是存在相位漂移的问题。

KShape 聚类方法使用互相关度来测量时间序列之间的距离。假设存在两个时间序列 $x=(x_1,...,x_m)$， $y=(y_1,...,y_m)$，假设固定 y 不动，x 序列滑动 s 个单位，则有：

$$x_s=\begin{cases}(0,...,0,x_1,x_2,...x_{m-s}),s\geq 0\\(x_{1-s},...,x_{m-1},x_m,0,...,0),s\leq 0\end{cases} \tag{5.5}$$

当序列向左或者向右滑动，空缺的部分补 0，则互相关的计算结果也是一个序列。当所有可能得相位漂移都考虑进去得时候，s 的取值范围是 $s\in[-m,m]$，可以的得到互相关系数 $CC_w(x,y)=(c_1,...,c_w)$，互相关系列的长度为 2m-1，它可以被定义为：

$$CC_w(x,y)=R_{w-m}(x,y),w\in\{1,2,...,2m-1\} \tag{5.6}$$

当相应的 $R_{w-m}(x,y)$ 被计算出来，就可以得到：

$$R_k(x,y)=\begin{cases}\sum_{l=1}^{m-k}x_{l+k}\cdot y_l,k\geq 0\\R_{-k}(y,x),k<0\end{cases} \tag{5.7}$$

我们的目标是计算 w 使得互相关系数 $CC_w(x,y)$ 达到最大。此时，在最优 w 值的基础上，x 对于 y 的最优漂移项是 x(s)，其中 s=w-m。接下来定义 KShape 聚类方法的核心指标之一，形状距离测度 SBD（Shape-Based Distance）：

$$SBD(x,y)=1-\max_w\left(\frac{CC_w(x,y)}{\sqrt{R_0(x,x)\cdot R_0(y,y)}}\right) \tag{5.8}$$

SBD 的取值在[0,2]之间，值越小，说明两个时间序列越相似。接下来介绍 KShape 算法的第二个核心：根据 SBD 来计算时间序列类的质心，首先要按照误差平方和最小的原理，使得各类时间序列到质心的 SBD 平方和最小：

$$\mu_k=\arg\max_{\mu_k}\sum_{x_i\in P_k}\left(\frac{\max_w CC_w(x_i,\mu_k)}{\sqrt{R_0(x_i,x_i)\cdot R_0(\mu_k,\mu_k)}}\right)^2 \tag{5.9}$$

根据上面式子(6)和(7)的结果，可以化简为：

$$\mu_k=\arg\max_{\mu_k}\sum_{x_i\in P_k}\left(\sum_{l\in[1,m]}x_{il}\cdot\mu_{kl}\right)^2 \tag{5.10}$$

写成向量的形式可进一步化为：

$$\mu_k=\arg\max_{\mu_k}\sum_{x_i\in P_k}(x_i^T\cdot\mu_k)^2=\arg\max_{\mu_k}\mu_k^T\sum_{x_i\in P_k}(x_i\cdot x_i^T)\mu_k \tag{5.11}$$

引入矩阵 $M=Q^T\cdot S\cdot Q$，则有：

$$\mu_k = \arg\max_{\mu_k} \frac{\mu_k{}^T \cdot Q^T \cdot S \cdot Q \cdot \mu_k}{\mu_k{}^T \cdot \mu_k} = \arg\max_{\mu_k} \frac{\mu_k{}^T \cdot M \cdot \mu_k}{\mu_k{}^T \cdot \mu_k} \tag{5.12}$$

到此为止，时间序列类质心的计算就转化为矩阵 M 特征值和特征向量的求解了。通过计算 SBD 距离以及上述的质心计算过程，就可以对本题小区的数据进行 KShape 聚类了。下面对样本小区基站的上行数据进行 K=16 的 KShape 聚类，得到结果如下：

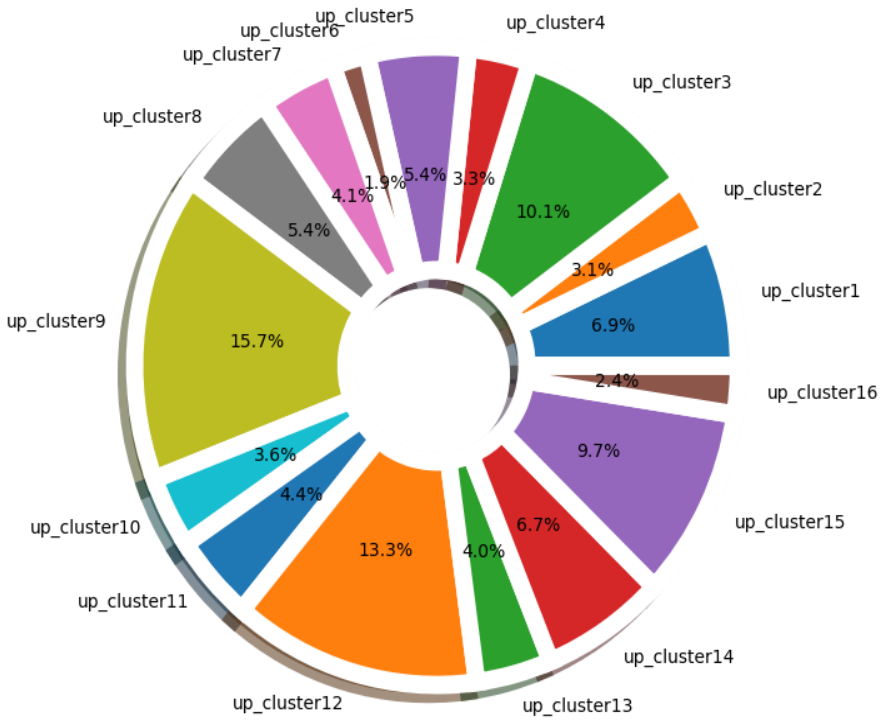

图 6.2 上行流量 Kshape 聚类各类小区占比

从上图可以看出，将上行流量聚类成 16 类小区，样本量最多的“up_cluster9”的小区占比为 15.7%，而样本量最少的“up_cluster6”的小区占比也有 1.9%，从 KShape 聚类结果来看，样本的分布较为均衡。将每一类所有小区基站流量的时间序列显示在一幅图中，并用红线画出它们的聚类中心，得到下面两幅图。

由图 6.3 可知，各类别的小区的时间序列有显著差异，其中第 1、5、7、6 类小区在时间窗口期内有显著的异常值，而其中第 6 类小区异常值在紧邻的两天之内出现了两次。在非异常值时期，这四类小区的流量水平要远远低于异常值水平，呈现出显著日周期性波动，但波动幅度有所差异。第 3 类和第 4 类小区的呈现出显著的工作日和周末的差别，周末流量阈值显著低于工作日，其中第 4 类小区工作日存在两个显著的波峰突起。第 2 类和第 8 类小区的流量阈值分别呈现出显著的逐渐下降和逐渐上升趋势。

由图 6.4 可知，第 9 类至第 16 类小区的时间序列也有显著的差异。第 11、13、16 类小区有显著单峰异常值，而且它们的位置都不相同。第 10 类小区在时间窗口的末端出现了多峰异常值，其余时间都处于较为平稳的周期波动。第 12 类和第 15 类小区的日内傍晚时分会出现瞬间的流量高峰时段。第 14 类小区呈现出显著的工作日和周末的差异，与前面不同的是，第 14 类小区的周末流量要显著比工作日流量高，推测该类小区应处于娱乐场所附近。

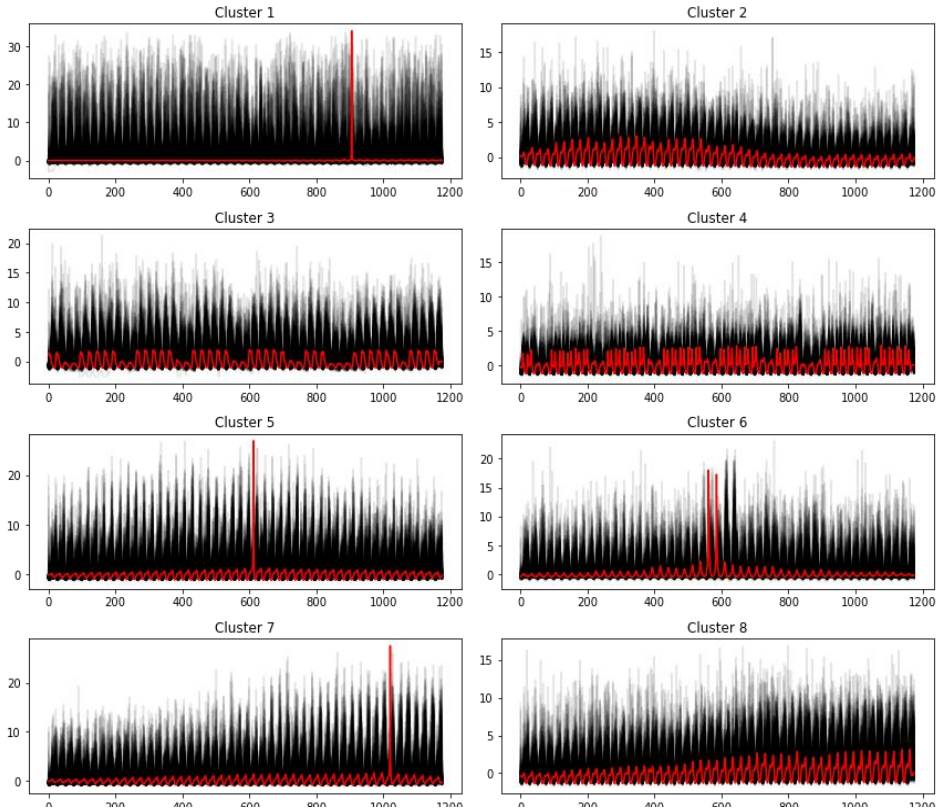

图 6.3 Kshape 上行流量第 1 至第 8 类小区

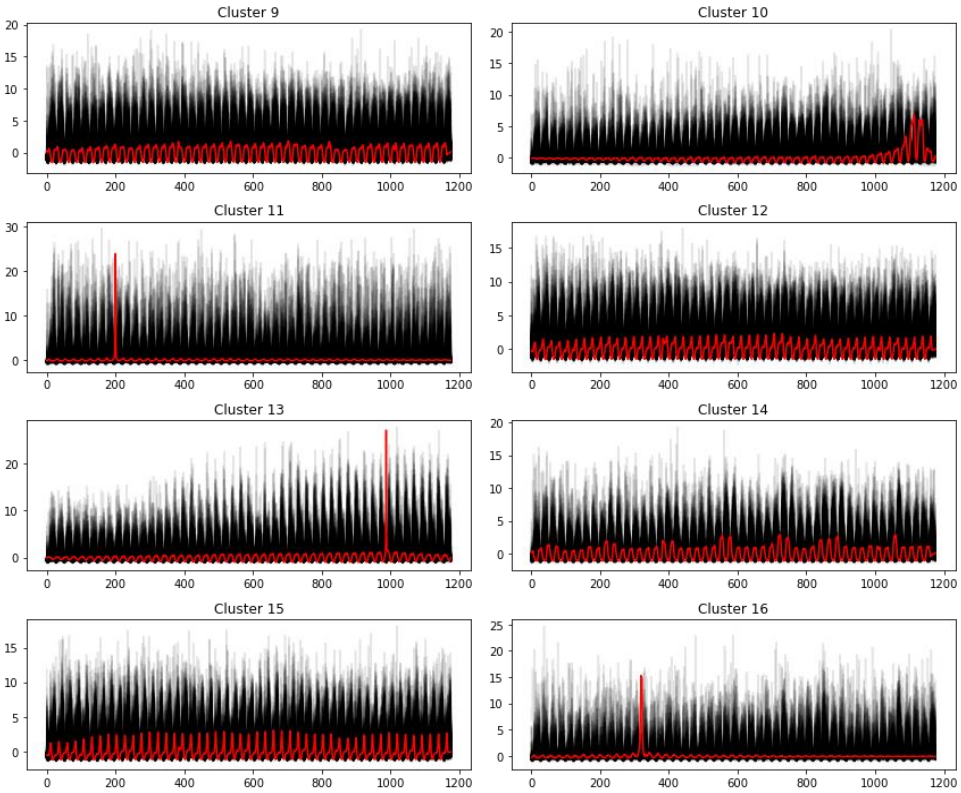

图 6.4 Kshape 上行流量第 9 至第 16 类小区

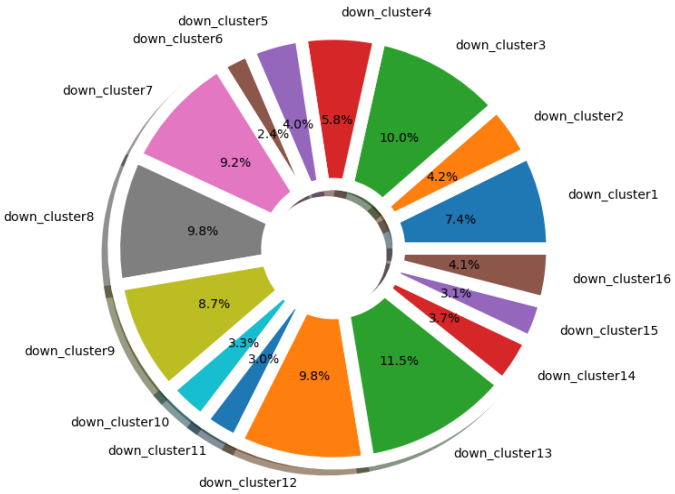

图 6.5 下行流量 Kshape 聚类各类小区占比

从上图可以看出，将下行流量聚类成 16 类小区，样本量最多的“up_cluster13”的小区占比为 11.5%，而样本量最少的“up_cluster6”的小区占比也有 2.4%，从 KShape 聚类结果来看，第 3、12、13 类小区的占比都超过了 10%，总体分布较为均衡。将每一类所有小区基站流量的时间序列显示在一幅图中，并用红线画出它们的聚类中心，得到下面两幅图：

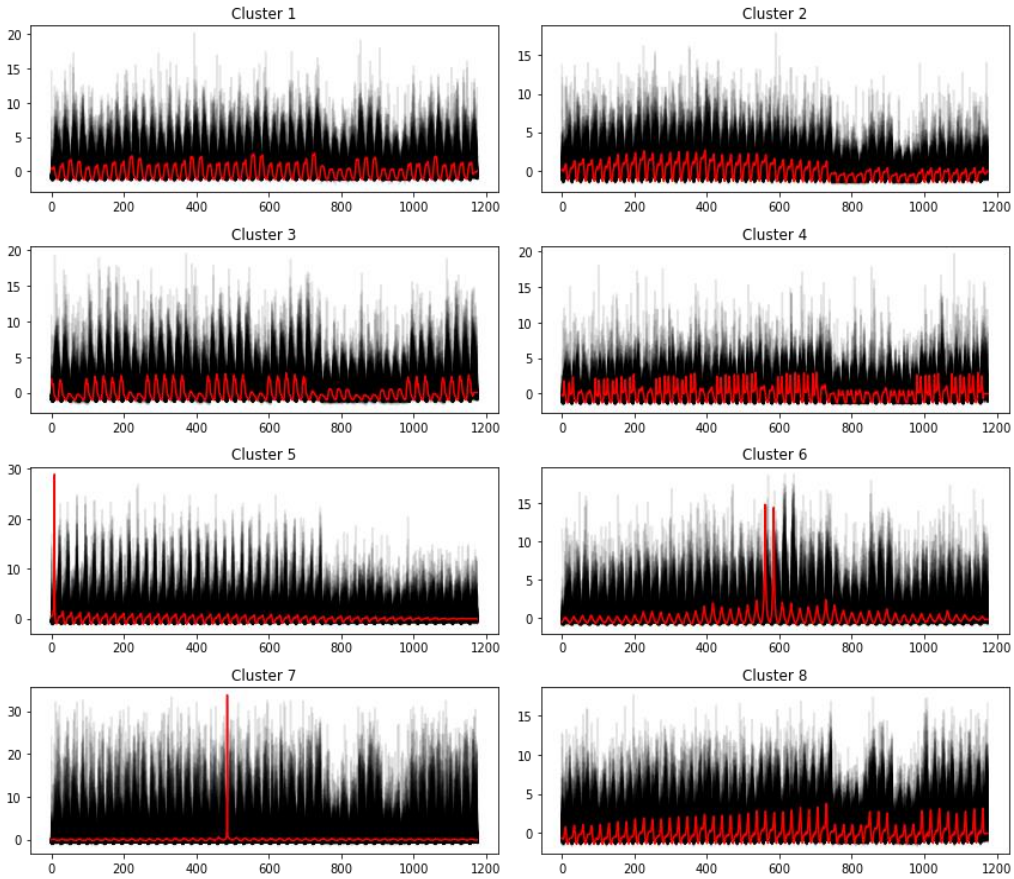

图 6.6 Kshape 下行流量第 1 至第 8 类小区

由上图可知，第 5、7 类小区在时间窗口期间出现了单峰的异常值，而第 6 类出现了紧邻两天的双峰的异常，异常值当天的流量要显著大于平常时期。第 1、3、4 类小区出现了明显的工作日和周末的差异，第 1 类小区是周末流量要显著大于工作日流量，而第 3、4 类小区是工作日流量显著大于周末流量，而且在第 750 小时至第 950 小时之间出现低峰异常值的情况。第二类小区流量呈现出明显的分阶段，从第 0 时到第 750 小时的阶段流量较高，而第 750 小时后切换到较低的第二阶段。

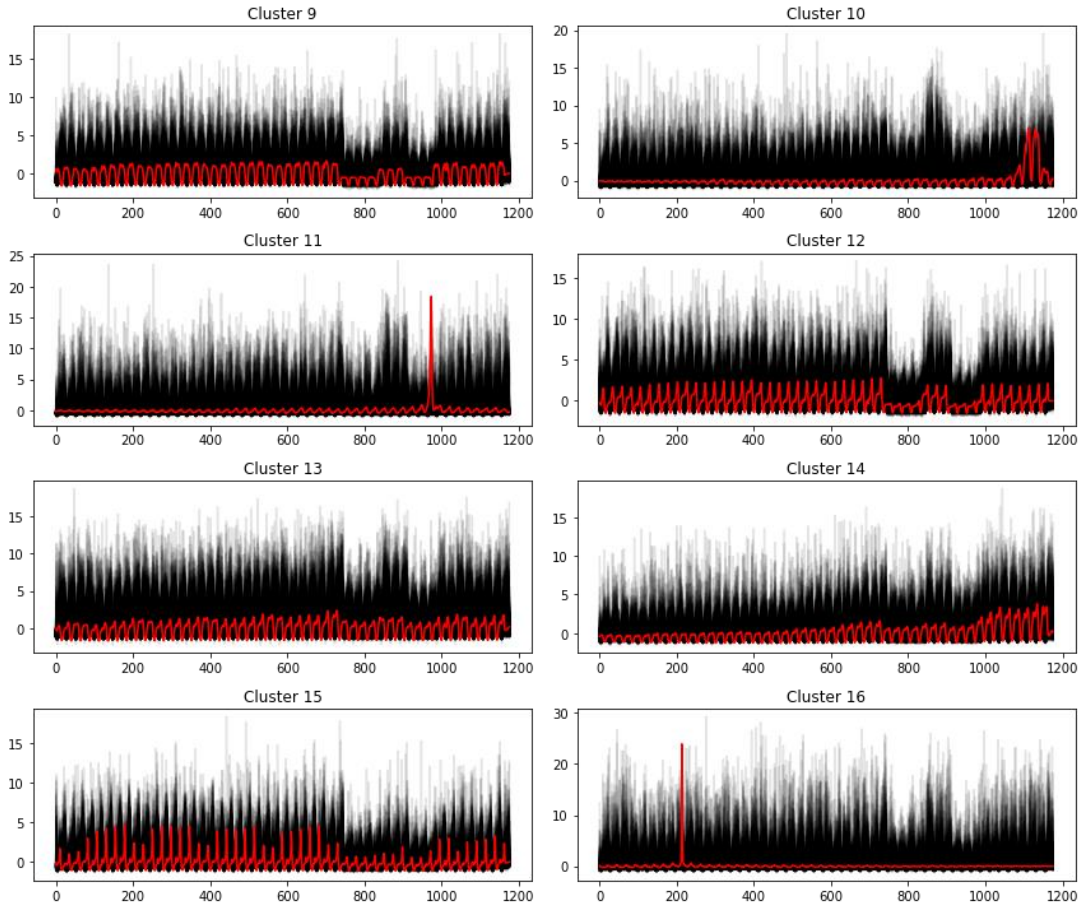

图 6.7　Kshape **下行流量第** 9 **至第** 16 **类小区**

由上图可知，第 9 至第 16 类小区的下行流量存在显著差异。第 11、16 类小区分别在时间窗口期的后段和前段出现单峰的异常值。第 9 类和第 12 类在第 800 小时和第 970 小时附近有连续几天的异常值，异常值的流量水平要显著低于平时，其中第 12 类小区还会出现瞬时的日内峰值。第 10 类小区在时间窗口期末端出现连续几天的高峰异常值。第 14 类小区流量呈现出逐步上升的趋势，在计算该类小区阈值的时候需要考虑基站扩容。第 15 类小区出现了显著的工作日和周末的差异，工作日的流量要显著高于周末的水平。

6.2 模型构建

本文所使用的动态阈值模型由均值预测模型和分位数模型共同构成。其中，均值预测由 LSTM 模型实现，分位数模型的求解则由 Cornish-Fisher 展式计算求得。该模型具有以下优势：其一，不同基站的阈值是根据基站的实际上下行流量动态调整的，相较于静态的阈值设定能够基于基站实际状况进行调整；其二，采用 Cornish-Fisher 展式计算上下行流量的分位数，可以基于有限的样本数据近似求解上下行流量的分位数，避免

了分类后不同类别基站的数据量有限的问题，具有很强的实用性；其三，在给出基站实际上下行流量不超过阈值某一概率的条件下计算阈值，避免了关于基站用户满意度评价和能源消耗之间的损失函数的设定难题。

6.2.1 Cornish-Fisher 展式[12]

Cornish-Fisher 展式是 Cornish 和 Fisher 提出的一种在已知部分样本点的情况下近似求解原分布及其分位数的方法。该方法假设随机变量 X 的数学期望为 $\mu = E(X)$，标准差为 $\sigma = 1$，则记 $F_{k,t}^{i}(x)$ 为第 i 类，第 t$(t = 0,1,\cdots,23)$时的上下行流量预测值的标准化残差的真实分布，$\Phi(\upsilon)$ 为标准正态分布的分布函数；x_p、υ_p 分别为 $F_{i,t}(x)$、$\Phi(\upsilon)$ 的 p 分位数（0<p<1）。根据 Cornish-Fisher 展式，保留前 4 项则有：

$$\begin{aligned} x_p &= \upsilon_p + \frac{1}{6}\left(\upsilon_p^2 - 1\right)\kappa_3 + \frac{1}{24}\left(\upsilon_p^3 - 3\upsilon_p\right)\kappa_3 - \frac{1}{36}\left(2\upsilon_p^3 - 5\upsilon_p\right)\kappa_3^2 \\ &\quad + \frac{1}{120}\left(\upsilon_p^4 - 6\upsilon_p^2 + 3\right)\kappa_5 - \frac{1}{24}\left(\upsilon_p^4 - 5\upsilon_p^2 + 2\right)\kappa_3\kappa_4 + \cdots \\ &\approx \upsilon_p + \frac{1}{6}\left(\upsilon_p^2 - 1\right)\mathrm{s} + \frac{1}{24}\left(\upsilon_p^3 - 3\upsilon_p\right)k - \frac{1}{36}\left(2\upsilon_p^3 - 5\upsilon_p\right)s^2 \end{aligned} \tag{6.3}$$

其中，累量 κ_n 为矩函数 $g(x) = \sum_{n=1}^{\infty}\kappa_n \frac{x^n}{n!} = \mu x + \sigma^2\frac{x^2}{2} + \cdots$，在原点处的导数，即：

$$\begin{cases} \kappa_1 = \mu \\ \kappa_2 = \mu_2 \\ \kappa_3 = \mu_3 \\ \kappa_4 = \mu_4 - 3\mu_2^2 \\ \kappa_5 = \mu_5 - 10\mu_3\mu_2 \end{cases} \tag{6.4}$$

其中，$\mu = E(X)$；$\mu_n = E\left[\left(X - \mu\right)^n\right]$。

由于在实际的上下行流量预测过程中，上下行流量的预测残差未必满足残差分布的标准差为 1 的，均值为 0 的假设，因此，参考于孝建等[6]（2018）中关于动态风险价值 VaR（value at risk）的计算方法计算上下行流量的分位数，将预测得到的上下行流量的分位数作为不同分位数下的阈值。具体的计算方法可用如下公式表示：

$$\begin{aligned} z_{k,t}^{i,q} &= y_{k,t}^{i} + \sigma_{i,t}F_{k,t}^{i\ -1}(q) \\ &\approx y_{k,t}^{i} + \sigma_{i,t}x_q \\ &= y_{k,t}^{i} + \sigma_{i,t}\left(\upsilon_p + \frac{1}{6}\left(\upsilon_p^2 - 1\right)\mathrm{s} + \frac{1}{24}\left(\upsilon_p^3 - 3\upsilon_p\right)k - \frac{1}{36}\left(2\upsilon_p^3 - 5\upsilon_p\right)s^2\right) \end{aligned} \tag{6.5}$$

其中，i 表示第 i 个大类；$z_{k,t}^{i,q}$ 表示所求的 q 分位数下的阈值；k 表示第 k 日；t 为第 t 个小时；$\sigma_{i,t}$ 预测值的标准差，这里使用样本的标准差 $\hat{\sigma}_{i,t}$ 替代；$y_{k,t}^{i}$ 即为上行或下行流量的预测值，由 LSTM 模型预测得到；$F_{k,t}^{i}(x)$ 为样本的标准化残差的真实分布，$F_{k,t}^{i\ -1}(x)$ 为其反函数。

6.2.2 LSTM 模型

LSTM 模型是基于 RNN 递归神经网络的一种变体。RNN 作为用于时间序列处理的神经网络，在不断学习的过程中 t 时刻的状态包含了 t 以前时刻的所有信息，考虑了序列的相关性[7]。但是，对于 RNN 的标准结构来说，存在梯度消失或者爆炸问题的可能，因此对于时间序列长期依赖关系的学习拥有一定的难度。而长短期记忆网络 LSTM 模型改变了 RNN 内部的隐藏层神经元结构，有效地解决了序列存在的长期依赖问题。长短期记忆网络神经元结构如下图 6.1 所示。

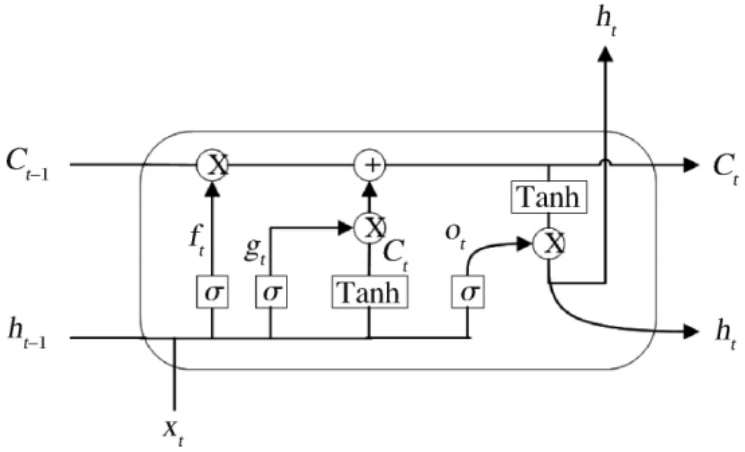

图 6.8 长短期记忆网络 LSTM 模型的结构[①]

长短期神经网络的结构含有一系列循环连接的记忆模块，其中连接细胞 cell 是长短期记忆网络模型中具有记忆信息储存的关键结构，LSTM 通过输入门、输出门和遗忘门调节 cell 的状态变化和输出当前结果。LSTM 网络模型的具体执行过程如下图 6.2 所示：

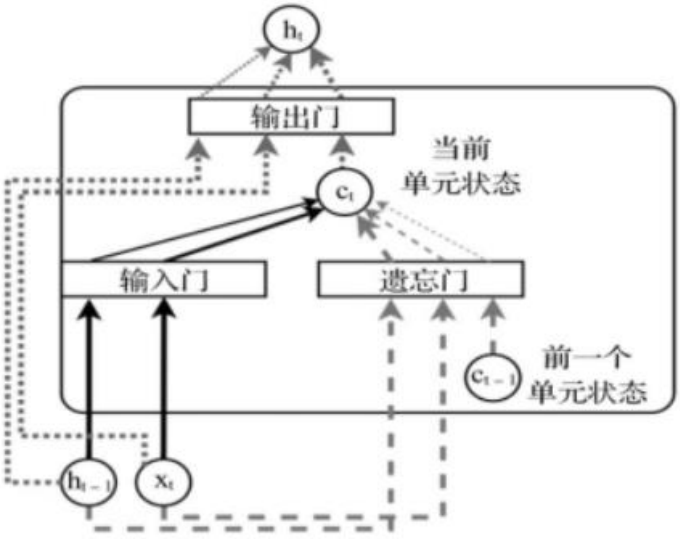

图 6.9 LSTM 模型的执行过程[②]

[①] 任嘉鹏. 基于机器学习的流量预测及基站休眠方法研究[D].吉林大学,2020.

[②]资料来源：https://www.csdn.net/tags/NtTaIg0sMTM1OC1ibG9n.html

（1）遗忘门

长短期记忆网络模型结构中的遗忘门是决定连接细胞(cell)状态丢弃哪些信息。遗忘门首先通过读取神经元前一时刻的输出结果 h_{t-1} 以及当前时刻的输入 x_t，其次运用 sigmoid 激活函数计算得到输出矩阵 f_t，矩阵 f_t 中的元素介于 0 到 1 之间，最后将矩阵 f_t 乘以前一时刻连接细胞 cell 的状态 C_{t-1}，从而决定遗忘哪些信息。主要公式如下所示：

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f) \tag{6.6}$$

其中中σ表示激活函数 sigmoid，W_f 代表遗忘门的权重参数矩阵，偏置参数表示为 b_f。

（2）输入门

输入门决定了连接细胞 cell 的信息更新方式。输入门主要通过读取神经元前一时刻的输出结果 h_{t-1} 以及当前时刻的输入 x_t，并计算获得连接细胞的最新状态的信息 C_t。主要公式如下所示：

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i) \tag{6.7}$$

$$C_t^{'} = \sigma(W_c \bullet [h_{t-1}, x_t] + b_c) \tag{6.8}$$

$$C_t = f_t * C_{t-1} + i_t * C_t^{'} \tag{6.9}$$

其中，输入门的输出用 i_t 表示，输入门和连接细胞状态的权重参数矩阵分别用 W_i 和 Wc 表示，$C_t^{'}$和b_i、b_c分别代表连接细胞状态的更新量和偏置参数。

（3）输出门

LSTM 模型中的输出同时受到当前输入与前一时刻输出的影响，其主要通过读取神经元前一时刻的输出结果 h_{t-1} 以及当前时刻的输入 x_t，并联合连接细胞状态信息 C_t 计算，得到神经元当前时刻的输出结果 h_t。有关计算公式如下所示：

$$O_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o) \tag{6.10}$$

$$h_t = O_t \times \tanh(C_t) \tag{6.11}$$

其中，输入门的输出用 O_t 表示，输入门权重参数矩阵分别用 W_o 表示，b_o 代表偏置参数。

LSTM 模型通过记忆信息的结构 cell 和特殊的“三门”控制机制，使得长短期记忆网络模型在面对序列的长期依赖问题时表现出了良好的模型效果，弥补了标准 RNN 模型的不足。总的来说，LSTM 模型同时包含了标准 RNN 模型中隐藏层单元间的外部循环和细胞内部的内循环。因此，LSTM 模型对时间序列的基站流量历史数据挖掘和对序列的长期依赖性的考虑更加完备，建立 LSTM 模型对基站流量数据进行预测具有一定的理论可行性。

长短期记忆网络 LSTM 模型的具体参数设置如下表所示。

表 6.1 长短期记忆网络参数设置说明

参数名称	参数设置说明
输入层特征维数	4
优化算法	Adam
迭代次数	1000
时间步长	24
Minibatch	24
梯度阈值	1
损失函数 loss	MSE
LSTM 层数	4
LSTM 层节点数	128

6.3 模型结果分析与检验

6.3.1 模型结果分析

将 LSTM 模型对各时点上行流量的预测值与各时点去均值后的分位数相结合便可得到不同分位数下的上行流量的阈值。75%分位数下与 90%分位数下各时点的阈值与其真实值如下图所示：

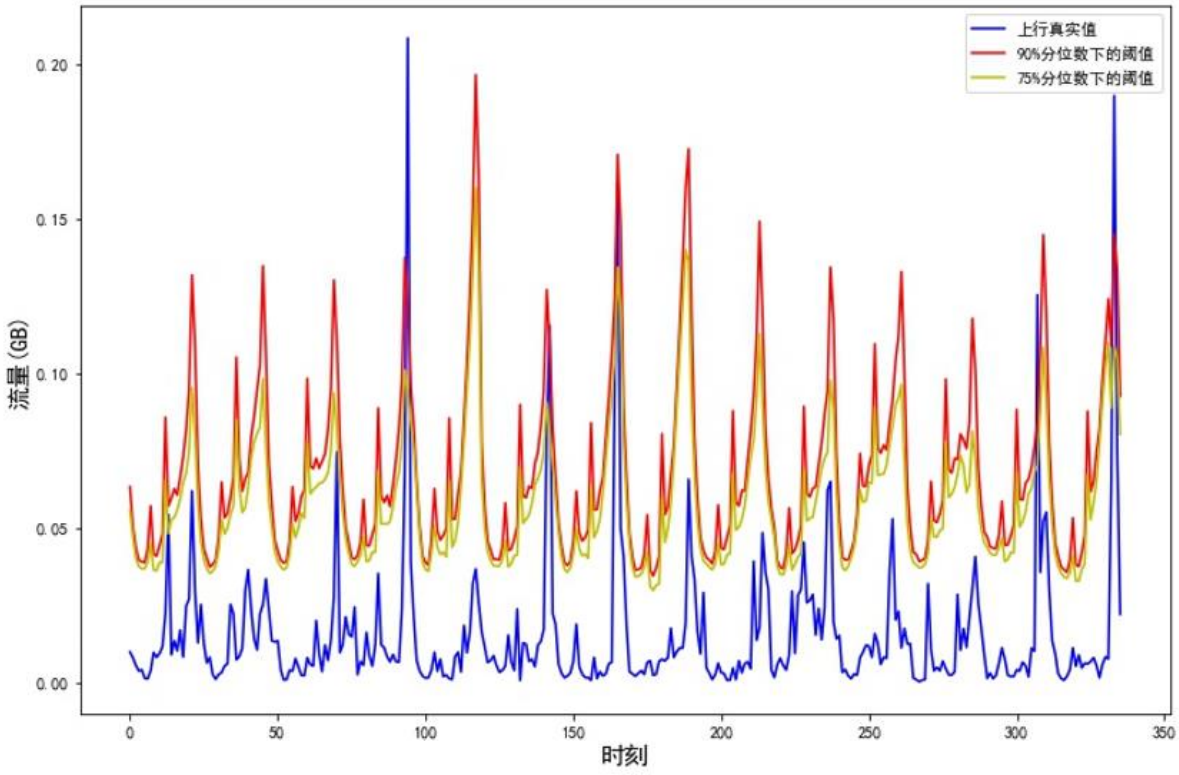

图 6.10 上行流量不同分位数下阈值与真实值对比情况

上图中，95%分位数下的上行流量阈值表示在 95%的概率下该时点的上行流量不会超过该阈值，同理，75%分位数下的上行流量阈值表示在 75%的概率下该时点的上行流量不会超过该阈值，因此，95%分位数下的上行流量阈值要高于 75%分位数下的阈值。

通过观察上图不难发现一下两点：其一，由于流量具有潮汐效应，基于本文所提出

的方法得到的阈值能够很好地拟合出潮汐效应的变化趋势，在基站负载高时设定较高的阈值，在基站负载低时设定较低的阈值，在不影响或对用户影响较小的条件下达到了尽可能节约能源的目的；其二，在绝大多数时刻，75%分位数的阈值和 90%分位数下的阈值要高于真实值，只有在个别时点真实值会超过阈值，这表明在大多数时刻，阈值的设定并不会降低用户的体验。

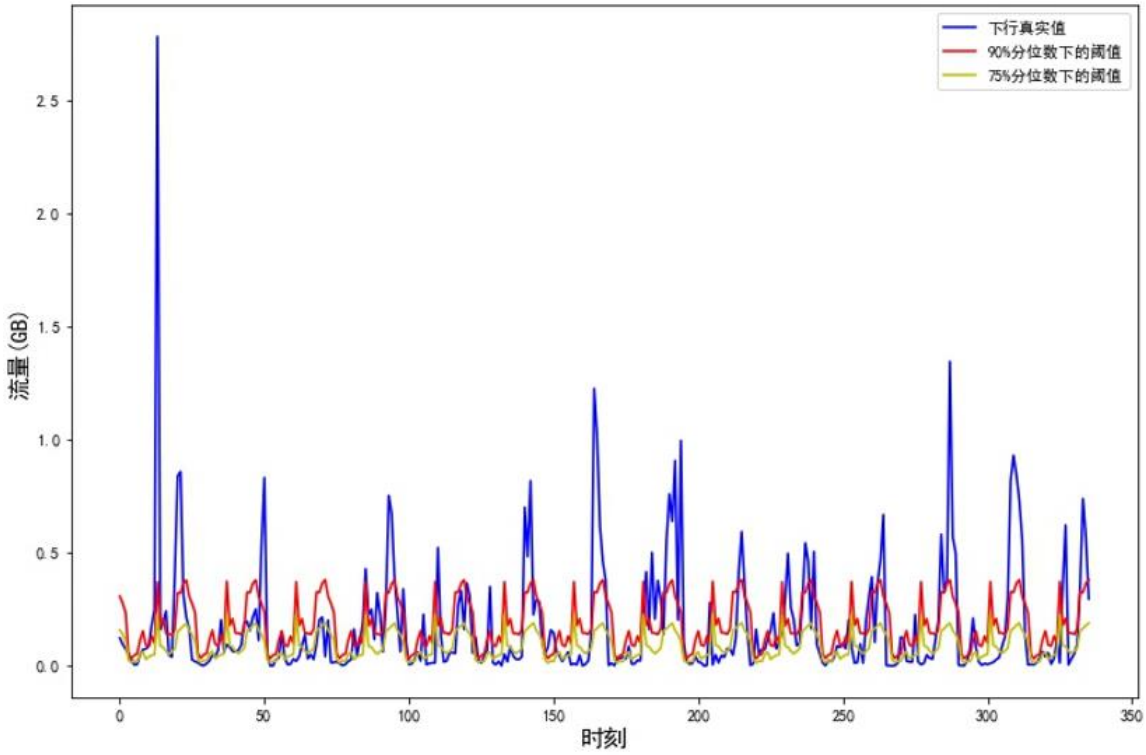

图 6.11 下行流量不同分位数下阈值与真实值对比情况

由上图可以看出，与不同分位数下的上行流量阈值类似，95%分位数下的下行流量阈值要高于 75%分位数下的下行流量阈值，且基于本文提出的方法计算出的不同分位数下的下行流量阈值也能够很好地拟合出下行流量的时变趋势，在基站负载高时设定较高的阈值，在基站负载低时设定较低的阈值，在不影响或对用户影响较小的条件下达到了尽可能节约能源的目的。

与上行流量分位数阈值不同的是，虽然在大多数时点下行流量的真实值要低于阈值，但在流量负载较高时，下行流量的实际值可能会超过 75%和 95%分位数下阈值，这可能是由于预测偏差导致的。因此，本文建议下行流量的阈值可以设定在一个相对较高的分位数的基础之上。

6.3.1 模型检验

对不同分位数下的上下行流量阈值进行失败率检验，检验结果如下表所示：

表 6.2 分位数失败率检验表

分位数	75%上行分位数	90%上行分位数	75%下行分位数	90%下行分位数
失败率	0.006	0.003	0.533	0.298

由上表中的失败率检验结果可知，对于 75%分位数下的上行流量分位数阈值设定和

90%分位数下的上行流量分位数阈值设定而言，失败率分别低于其理论值（0.25、0.10）。而对于 75%分位数下的下行流量分位数阈值设定和 90%分位数下的下行流量分位数阈值设定而言，失败率则分别高于其理论值。结合 6.3 节的分位数图来看，这可能是由于预测值的偏差造成的。因此，对于上下行流量分位数阈值的设定，本文建议对于下行流量阈值的设定选择一个相对较高的分位数，例如 90%或 95%。

七、模型评价与推广

7.1 问题一的模型评价与推广

针对问题一，为了降低模型复杂度和提高运行效率、聚类效果，本文出于大样本大数据的考虑，首先采取随机抽样获得测试用样本小区数据集；其次利用 tsfresh 工具提取时间序列的统计特征、熵特征和分段特征等作为对应的特征向量进行聚类，得到特征向量 F。同时，基于随机森林法对构成的特征向量各个特征之间的重要性进行分析。再利用 kmeans 方法进行基于特征的聚类分析并对各类数据特点进行描述；最后，运用 DTW 算法改进的 KNN 模型将非测试用样本小区分类。构建的模型的优缺点如下：

优点：

（1） 模型将大样本转换成小样本测试，可以有效地提升了效率与速度。

（2） 模型采取基于特征的 kmeans 聚类，将时间序列数据处理问题转换成了静态的问题，同时起到了数据降维的效果。

（3） 采用基于 DTW 算法改进的 KNN 模型对剩余样本进行类别划分，提高了最终模型的分类准确率。

（4） 对大样本、大数据分类问题有一定的现实可操作性。

缺点：

（1） 模型训练测试利用的是随机抽样的相对较小的样本，可能不如用大样本直接聚类获得的结果更加精准。

（2） Kmeans 算法的效果在一定程度取决于初始质心的选择，若初始质心的选择可能会对模型的最终效果产生影响。

（3） 模型在使用过程中，对于特征的选取并不全面，仅仅是根据时间序列的一般特征进行选取，可能在实际运用过程中还需要进一步的改进。

7.2 问题二的模型评价与推广

在处理问题二中的上下行流量阈值设定问题上，本文创新性地采用了 LSTM 模型与 Cornish-Fisher 展式相结合的方法来设定阈值，该方法主要具有以下优点：

（1）Cornish-Fisher 展式在拟合扰动项的分布时对数据量的要求不高且对计算机运算能力的要求较低。

（2） LSTM 模型改变了 RNN 内部的隐藏层神经元结构，采用 LSTM 模型来预测流量的期望值可以有效地解决时间序列存在的长期依赖问题。

（3）采用 LSTM 模型和 Cornish-Fisher 展式相结合的方法能够根据上下行流量潮汐效应的变化和随机扰动分布特征的变化来动态地设定阈值，克服了静态阈值设定与流量潮汐效应的矛盾。

（4）采用分位数的方法来设定阈值，有利于决策者把控在多大概率不影响用户体验的基础上降低能源消耗，在用户体验和能源消耗之间找到平衡点。

该方法的不足之处在于：

阈值的设定会受到预期上下行流量预测准确性和随机扰动项原分布拟合准确性的影响，在较高的流量预期准确性和优良的随机扰动项原分布拟合效果的基础之上，通过该方法可以计算出一个理想的阈值，反之，阈值的设定则难以达到理想效果。

八、参考文献

[1] Wang L,Wang X,Leckie C,et al.Characteristic-based descriptors for motion sequence recognition//Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg,2008:369-380.

[2]纪勇.基于时间序列特征的基站聚类[J].电子世界,2016(14):165-167.

[3] Nanopoulos A,Alcock R,Manolopoulos Y.Feature-based classification of time-series data. International Journal of Computer Rese arch,2001,10(3):49-61.

[4] Wang X,Smith K,Hyndman R.Characteristic-based clustering for time series data.Data mining and knowledge Discovery,2006,13(3): 335-364.

[5] Mörchen F.Time series feature extraction for data mining using DWT and DFT.2003.

[6] 于孝建,王秀花.基于混频已实现 GARCH 模型的波动预测与 VaR 度量[J].统计研究,2018,35(01):104-116.

[7]欧阳红兵,黄亢,闫洪举.基于 LSTM 神经网络的金融时间序列预测[J].中国管理科学,2020,28(04):27-35.

[8][10]《Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping 》——Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista2 , Brandon Westover1 , Qiang Zhu, Jesin Zakaria, Eamonn Keogh

[9]李海林,梁叶,王少春.时间序列数据挖掘中的动态时间弯曲研究综述[J]. 控制与决策, 2018, 033(008):1345-1353.

[11]C. S. Myers,L. R. Rabiner. A Comparative Study of Several Dynamic Time‐Warping Algorithms for Connected‐Word Recognition[J]. John Wiley & Sons, Ltd,1981,60(7).

[12]李美霖,崔雪松,姜今锡.基于Cornish-Fisher展开的样本分位数渐进展开算法[J].延边大学学报(自然科学版),2011,37(04):306-310+344.

[13]郑康,段然,吴杰,袁宇恒.基于AI和O-RAN架构的5G网络容量自适应算法[J].电信工程技术与标准化,2020,33(01):19-24.

[14]沈瑶. 基于大数据的基站流量预测与网络规划算法研究[D].南京邮电大学,2019.

[15]彭铎,周建国,羿舒文,江昊.基于空间合作关系的基站流量预测模型[J].计算机应用,2019,39(01):154-159.

[16]仝宗健. 5G超密集网络中基于流量地图的基站节能算法[D].北京邮电大学,2018.

[17]廖伟琛. 基于聚类的数据挖掘技术在未来网络基站部署策略中的应用[D].北京邮电大学,2018.

[18]陈薇,袁中原,高波.基于 LSTM 神经网络预测低温热源动态特性[J].制冷与空调(四川),2020,34(06):670-675.

[19]张宇峰.基于LSTM神经网络对深证综指预测分析[J].商讯,2020(36):81-82.

[20]胡铮,袁浩,朱新宁,倪万里.面向 5G 需求的人群流量预测模型研究[J].通信学报,2019,40(02):1-10.
[21]孙晓爽.异构网络中的基站休眠——基于流量预测的方法[J].电子技术与软件工程,2017(10):28-29.
[22]张佳鑫,张兴,李永竞,王硕,杨居沃,梅承力,王文博.蜂窝网络中基站关系与业务关系网络与应用[J].中国科学:信息科学,2017,47(05):648-663.

附录

所用程序软件:python ； 详细代码见附件

基于特征的 Kmeans 聚类

```
from sklearn.cluster import KMeans
from scipy.spatial.distance import cdist
from sklearn import metrics
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
#解决画图时标题不能为中文的问题
plt.rcParams['font.sans-serif']=['SimHei'] #替换成中文字体
plt.rcParams['axes.unicode_minus'] = False #解决坐标轴负数的负号显示问题
os.chdir(r"C:\Users\fht\Desktop\Mathorcup 复赛")
data=pd.read_csv('newfeatures.csv')
data=data.drop(['id'],axis=1)
data.head()
# 肘部法则确定最优聚类数
K=range(2,30)
meandistortions=[]
for k in K:
    kmeans=KMeans(n_clusters=k)
    kmeans.fit(X)
    meandistortions.append(sum(np.min(cdist(
            X,kmeans.cluster_centers_,"euclidean"),axis=1))/X.shape[0])
plt.figure(figsize=(10,8))
plt.plot(K,meandistortions,'-*')
plt.xlabel('聚类数 k',fontsize=15)
plt.ylabel('平均畸变程度',fontsize=15)
plt.savefig('寻找的最佳聚类数.jpg')
# plt.title('用肘部法则来确定最佳的 K 值')
# k-means 聚类
estimator=KMeans(n_clusters=16)
estimator.fit(data)
label_pred=estimator.labels_
label_pred=pd.DataFrame(label_pred).T
centroids=estimator.cluster_centers_
```

```
centroids=pd.DataFrame(centroids)
centroids.to_csv('centroids.csv')
# 加载数据
# 上行流量特征
data_fea_up=pd.read_csv('up_features.csv')
data_fea_down=pd.read_csv('down_features.csv')
# 求不同特征间的相关性
data_up_corr=data_fea_up.corr()
data_down_corr=data_fea_down.corr()
# 绘制不同特征间的相关性热力图
# up_hot
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus'] = False

plt.figure(figsize=(12,10))
sns.heatmap(data_up_corr,annot=True, vmax=1, square=True)#绘制 new_df 的矩阵热力图
plt.savefig('up_hot.jpg')
plt.show()#显示图片
聚类结果与可视化
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
from  sklearn.preprocessing import  MinMaxScaler,OneHotEncoder
# 计算每个聚类下样本数量和占比情况
#计算每个聚类类别的样本量
clusters_count2 = pd.DataFrame(down_he['xiaoqubianhao'].groupby(down_he['sort']).count()).T.rename({'xiaoqubianhao':'counts'})
#计算每个聚类类别的样本占比
clusters_rat2 = (clusters_count2/ len(down_he)).round(2).rename({'counts':'percentage'})
print('每个聚类下的样本量：\n',clusters_count2,'\n\n 每个聚类下的样本占比：\n',clusters_rat2)
cluster_features2 = [] # 空列表，用于存储最终合并后的所有特征信息
for line in range(0,16): #读取每个类索引
    label_data =down_he[down_he['sort'] == line] # 获得特定类的数据
    part1_data = label_data.iloc[:,2:15] # 获得数值型数据特征
```

```
        part1_desc = part1_data.describe().round(3) # # 得到数值型特征的描述性统计信息
        merge_data1 = part1_desc.iloc[1,:]  # 得到数值型特征的均值

    #       part2_data = label_data.iloc[:, 7:-1]  # 获得字符串型数据特征
    #       part2_desc = part2_data.describe(include='all')  # 获得字符串型数据特征的描述性统计信息
    #       merge_data2 = part2_desc.iloc[2, :]  # 获得字符串型数据特征的最频繁值

    #       merge_line = pd.concat((merge_data1,merge_data2),axis=0)# 将数值型和字符串型典型特征沿列合并
        cluster_features2.append(merge_data1) # 将每个类别下的数据特征追加到列表

    # 输出完整的类别特征信息
    clusters_pd2 = pd.DataFrame(cluster_features2).T # 将列表转化为 DataFrame
    clusters_pd2.to_csv('clusters_pd2.csv')
    clusters_pd2=pd.read_csv('clusters_pd2.csv')
    clusters_count2=pd.read_csv('clusters_count2.csv')
    clusters_rat2=pd.read_csv('clusters_rat2.csv')

    #解决画图时标题不能为中文的问题
    plt.rcParams['font.sans-serif']=['SimHei']
    plt.rcParams['axes.unicode_minus'] = False
    fig = plt.figure(figsize=(6,6))#建立画布
    ax = plt.subplot(111,polar = True) # 增加子网格，设置 polar 参数绘制极坐标
    labels = np.array(merge_data1.iloc[0:6].index)  #设置要展示的数值数据标签
    cor_list = ['r','g','b','k'] # 定义不同类别的颜色
    angles = np.linspace(0,2 * np.pi,len(labels),endpoint = False) # 计算各个区间的角度
    angles = np.concatenate((angles,[angles[0]])) #建立相同首尾字段以便于闭合
    labels=np.concatenate((labels,[labels[0]])) #建立相同首尾字段以便于闭合
    for i in range(4): #循环每个类别
        data_tmp = num_sets_max_min[i,:] #获得对应类数据
        data = np.concatenate((data_tmp,[data_tmp[0]])) # 建立相同首尾字段以便于闭合
        ax.plot(angles,data,'o-',c=cor_list[i],label = '%i 类小区'%(i+1))# 画线
        ax.fill(angles,data,alpha = 0.5) # 区域填充颜色

    #设置图像显示格式
    ax.set_thetagrids(angles * 180 / np.pi,labels,fontsize=20)# 设置极坐标轴
```

```
ax.set_title("不同类别小区的显著特征对比 1_down",fontsize=20)  # 设置标题放置
ax.set_rlim(-0.2,1.2) #设置坐标轴尺度范围
plt.legend(loc = 'best',bbox_to_anchor = (1.2,1.0))  #设置图例位置
plt.savefig('不同类别小区的显著特征对比 1_down.jpg')
plt.show()
```