中国研究生创新实践系列大赛
“华为杯”第十六届中国研究生数学建模竞赛

学　　校	湖南大学
参赛队号	19105320030
队员姓名	1.许晓玥
	2.邓　彬
	3.鲁文格

中国研究生创新实践系列大赛
“华为杯”第十六届中国研究生数学建模竞赛

题 目 汽车行驶工况构建

摘 要：

汽车行驶工况是描述汽车在不同工况下运动特征的速度时间曲线，广泛使用于汽车能效分析、动力性能分析和运动特征分析。我国目前使用欧洲提出的 NEDC 工况，然而由于汽车行驶工况与地理环境、自然气候、驾驶习惯等具有较大关系，所以国外的汽车行驶工况标准并不能很好地对我国汽车的实际行驶工况进行分析和预测。与此同时，随着新能源汽车的快速发展，新能源动力汽车将逐渐取代传统燃油汽车，因此用近 20 年前的标准也明显无法适应新的发展要求。所以，构建具有我国特色的汽车行驶工况具有十分重要的意义。本文按照数据预处理、运动片段提取、汽车运动特征评估体系构建、主成分分析降维、K-Means 聚类对运动片段进行分类、基于指标偏差的运动片段提取、汽车行驶工况分析、误差分析这几个步骤，建立了一套完整的汽车行驶工况构建方法，并按照题目要求解决相关问题。

针对问题一：根据题中所提出的五种不良数据类型，本文对原始数据进行了变速异常数据处理、数据填充处理及异常怠速和停车状态处理三个步骤，首先，进行变速异常数据处理，筛选出加速度和减速度超过最大阈值的数据行，予以删除；然后，进行数据填充处理，找到原始数据中时间不连续的点，统计不连续点之间的间隔，对缺失时间较短的数据利用缺失前后数据信息进行数据填补；最后，对异常怠速和停车状态进行判别，对于异常怠速数据，超过怠速最长时间（180s）以上的数据全部予以删除，对于停车状态，前后保留 5s 的 0 值余量，把中间数据全部删除。**最终，经过数据预处理过后，文件 1、文件 2 和文件 3 剩余的数据总量分别为 177760 行，141378 行和 154302 行**。

针对问题二：通过对预处理后数据进行分析和参考相关文献，本文为运动学片段的筛选制定了 3 条规则：（1）满足运动学片段的基本定义，即从一个怠速状态开始到下一个怠速状态开始间的行驶区间里，包含加速、减速和匀速三个过程；（2）片段时长限制，为保证片段的有效性，运动学片段的时长必须大于 20s；（3）运动学片段数据缺失率限制，保障片段的完整性， GPS 车速数据缺失率小于 10%；基于 3 条规则对预处理后数据进行运动学片段提取，**文件 1、文件 2 和文件 3 提取出的运动学片段个数分别为 876, 668 和 560。**

针对问题三：本文首先建立能广泛评估汽车运行特征的 11 个运动学指标（运行时间、平均速度、平均行驶速度、速度标准差、最大速度、最大加速度、最大减速度、平均加速度、平均减速度、加速度标准差、运行路程）和 4 个统计学指标（怠速时间比、加速时间比、减速时间比、匀速时间比），以便描述和比较运动片段、运动工况的运动特征。在此基础上使用主成分分析法简化数据结构，最终将 15 个运动特征指标降为 3 个综合指标，且这 3 个综合指标的累计贡献率达到 99%。接着利用 K-Means 聚类方法，以综合指标间的

距离为分类标准，将提取出的 2104 个运动片段分为 3 类，3 个聚类集合中分别包含 1904、184、16 个运动片段。在聚类的基础上，本文以一个运动片段的去量纲指标和与该片段所属聚类集合的去量纲指标和的偏差大小为依据，优先选取偏差小的运动片段作为所属聚类集合最具代表性的运动片段。按照汽车运行工况时间范围 1200s~1300s 的规定，**最终得到总长 1268s 的汽车行驶工况曲线，其中第一类运动片段（低速）占时 301s、第二类运动片段（中速）占时 392s、第三类运动片段（高速）占时 575s**。观察所构建汽车行驶工况，发现提取片段速度时间曲线所包含汽车运动特征信息与通过 15 个指标分析出的三种工况基本吻合。同时，**本文比较所构建汽车运行工况指标值与经处理后的采集数据指标值，各指标误差率集中在 2%~30%，其中速度标准差误差率为 2.046%、最大速度误差率为 5.651%，相似系数为 0.983，欧氏距离偏差率为 1.199%。**综合上述结果，可以说明本文所构建汽车运行工况具有一定代表性。

关键词：运动学片段；汽车行驶工况；主成分分析；K-Means 聚类；欧氏距离偏差；

一、问题重述

1.1 问题背景

自 2009 年起，我国乘用车市场蓬勃发展，乘用车销量逐年上涨，在 2016 年突破 2000 万大关，预计到 2025 年，中国汽车销量将突破 3500 万辆，届时全球汽车销量更是将达到 1.2 亿辆[1,2]。

汽车产业的快速发展虽然节省了出行时间，加快了工作效率，但与此同时也带来了城市交通道路拥堵、化石能源消耗量上升、温室气体排放量快速增长等一系列问题。对于汽车数量快速增长带来的道路拥堵问题，可以通过合理规划城市道路、桥梁网络和设计智能化指挥系统来解决或者缓解；对于汽车运行带来环境污染和能源消耗的问题，可以改用“零排放”、能源转化效率高的清洁能源汽车来解决或缓解，汽车动力能源类型由不可再生能源向清洁能源的成功转型更是未来汽车产业可持续发展的关键因素。于是，如何计算、预测汽车污染物排放量、燃油消耗量以及动力性能，如何统计不同交通环境下不同类型车辆的行驶特征和运行情况，成为城市交通网络合理规划和车辆技术开发、评估的基础。

汽车行驶工况可以用来描述特定类别汽车在不同交通环境下的运动学特征，具体表现为速度-时间曲线。该曲线时间长度可以从几百秒到几千秒，常用的 NEDC 工况曲线时间长度为 1180s，WLTC 工况曲线时间长度为 1800s[3]。此外，汽车行驶工况可根据交通环境（NEDC 工况采用分类标准）或者比功率（WLTC 工况采用分类标准）等分为不同循环，每个循环内又有包含一个或者多个工况，以此来描述在不同交通环境（如城市、郊区等）或比功率不同（如 PMR>34，表示高比功率）的汽车的行驶运动特征。汽车行驶工况广泛应用于车辆排放性、燃油经济性和动力性等测试中[4]，是汽车行业标准制定和汽车开发、评价、性能优化的基础。因此，研究汽车行驶工况与分析某地区交通拥堵情况、车辆运行情况，设计合理交通网络，计算汽车有害物质排放情况、油耗情况、能源利用率，制定汽车能效、环保、动力性能标准，评定车辆各类性能指标等级，研发和检验新型汽车等环节关系紧密，是汽车行业乃至道路规划行业的共性基础技术。

在汽车行驶工况的研究上，美国、欧洲和日本提出的三个标准工况体系最具代表性，目前我国大部分城市和地区采用的 NEDC 工况就是 1997 年由欧洲提出的[4,5]。而汽车行驶工况是具有地域性和车辆类别性的，地形地貌、交通环境、人口密度、汽车性能、路面等都会对其产生影响。同时，随着新能源汽车的兴起和发展，传统汽车行驶工况已不能合理评估新能源汽车的节能效果、制动能量回收效率和怠速起停等新技术的效果。使用其他国家和地区多年前制定的汽车行驶工况并不能很好的贴合我国的汽车行驶工况，地理环境、自然气候、城市规划乃至驾驶习惯等都会对汽车的运行特征产生影响，制定符合本国或者本区域的自然因素和社会因素的汽车行驶工况有利于精准描述汽车实际运行情况、正确指导汽车研发工作的走向，合理制定新的汽车行业指标和规划、提高政府公信力。但由于我国对于汽车行驶工况的研究起步较晚，现阶段只有部分一线城市（如北京、上海等）构建出了符合当地情况的汽车行驶工况。中国汽车研发中心牵头编制了 CLTC 中国工况，目前该工况处于征求意见阶段，还未施行[3]。

1.2 问题重述

基于上述背景，本文查阅相关资料对构建汽车行驶工况的方法、模型和算法进行研究，在此基础上依据原始数据文件夹中给出的不同时段，某一轻型汽车的道路行驶采集数据，解决以下三个问题：

（1）对文件 1、文件 2 和文件 3 给出的同一辆汽车不同时段的行驶数据进行数据预处理；

（2）结合运动学片段的定义，从进行问题（1）预处理后的数据中提取合理的运动学片段，并给出最终得到的运动学片段数量；

（3）科学构建汽车行驶工况曲线，建立全面合理的汽车运动特征评估体系，然后对比本文所构建汽车行驶工况与实际汽车行驶 v-s 曲线之间的误差，进行误差分析以说明本文所构建汽车行驶工况的准确性和合理性。

经过国内外的研究和实践，汽车工况构建主要有三种方法：数据采集法、计算机仿真法和人工经验法。显然，本题要求使用数据采集法构建汽车行驶工况。该方法的具体流程如图 2.1 所示：

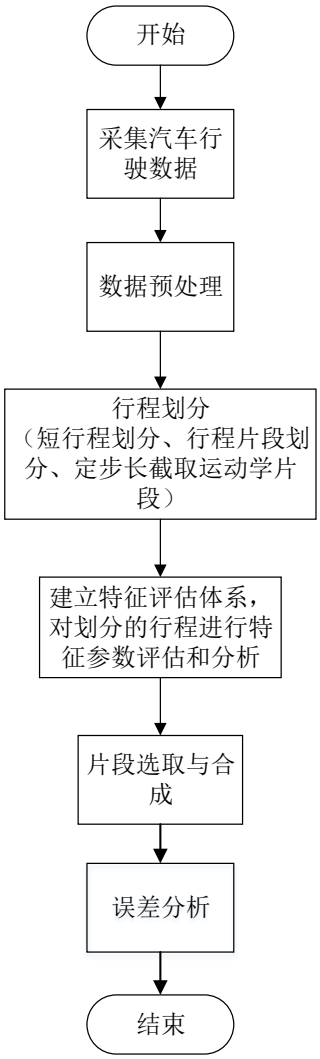


图 1.1 汽车行驶工况构建流程图

二、问题分析

本题要求使用原始数据文件夹中提供的同一汽车在三个不同时段的实际道路行驶数据，构建出能描述该汽车行驶工况的速度-时间曲线，并将所构建曲线的特征与采集数据的相应特征进行对比和误差分析，以验证本文所构建汽车行驶工况的准确性和合理性，即构建汽车行驶工况并分析验证其代表性优劣，对于题目中提出的问题具体分析如下：

针对问题一，为了对文件 1、文件 2 和文件 3 采集到的数据进行预处理，首先需要对数据缺失、变速异常、停车数据、持续低速、异常怠速五个类别的数据情况进行统计和分

析，然后制定相应的数据删减和添补的方案，最后依照处理方案，调整数据，得到预处理后的数据，其中异常怠速和停车状态存在交叉环节，如何对两者进行区分，并制定相应的数据删减方案是问题一的重点。

针对问题二，运动学片段是从一个怠速状态开始，经历加速、匀速和减速运行后，在下一个怠速状态开始前一个时间点结束的片段。通过考虑运动学片段的定义特征、时间连续性、过程完整性、结构正确性、长度合理性等原则制定运动片段筛选规则，根据规则从问题一处理后的数据中提取有效运动学片段。

针对问题三，通过运动片段构建运行工况，是一个从众多运动片段中选取最具有代表性的几个运动片段的过程，如何评判这个运动片段具有代表性以及如何评判最终构建工况曲线具有代表性是首先要解决的问题。因此，本题首先需要找到合理的指标来描述运动学片段以及工况曲线，在选取好汽车运动特征指标后，我们需要把众多的运动学片段通过聚类方法分为几个大的类别，然后，从每一大类中提取最能代表该类运动片段特征的运动学片段，最后根据各类运动片段在处理后的总运动曲线上的时间占比，得到各类运动片段在总运行工况上所占时间，结合排序后的备选片段和所占时间拼接得到最终的汽车运行工况曲线，得到运行工况曲线后，按照题目要求还需要进行误差分析，根据误差值判断所构建汽车运行工况的准确性。

三、模型假设

1. 假设汽车在采样时间 1s 内的任何变速过程皆为均匀的；
2. 假设汽车无故障行驶，行驶过程不考虑天气，汽车自身故障等情况的影响；
3. 假设汽车启动之后，汽车进入怠速状态，车主 5s 后起步；
4. 假设车主从停车到熄火存在 5s 间隔，该阶段仍属于怠速状态；
5. 假设所有参考资料都是真实无误的；

四、符号说明及名词定义

符号	含义
T	历经总时间
v_i	时刻i的速度
v_e	平均速度
v_{er}	平均行驶速度
v_{std}	速度标准差
$v_{\max}$	最大速度
$v_{\min}$	最小速度
$A_{a,\max}$	最大加速度
$A_{b,\max}$	最大减速度
$A_{a,e}$	平均加速度
$A_{d,e}$	平均减速度
A_{std}	加速度标准差
S	运行总路程
R_i	怠速时间比

R_a	加速时间比
R_b	减速时间比
R_e	匀速时间比
n_i	怠速运动时刻点个数
n_a	加速度运动时刻点个数
n_d	减速度运动时刻点个数
n_e	匀速运动时刻点个数
n_k	第 k 个运动片段的时刻点个数
$n_{i\sim j}$	速度从 i 到 j 的时刻点个数
$n_{a,i\sim j}$	加速度从 i 到 j 的时刻点个数
p	运动特征评估体系评估指标个数
m_k	第 k 类运动片段集中运动片段个数
L	主成分分析法中的系数矩阵
Y	主成分分析法中的主成分向量
R	主成分分析法中的相关系数矩阵
λ	主成分分析法中的相关系数矩阵的特征值
φ	主成分分析法中主成分贡献率
f	筛选出主成分个数
G	主成分分析后的主成分指标矩阵
b	运动特征评估体系中的指标矩阵
b_j	指标矩阵 b 中所有运动片段第 j 个指标的向量
$x_{i,j}$	运动特征评估体系中的无量纲化后的指标矩阵
Z	运动特征评估体系中的经标准化后的指标矩阵
u_j	运动特征评估体系中的第 j 个指标的数学期望
σ_j	运动特征评估体系中的第 j 个指标的方差
$\overline{z_i}$	标准化指标矩阵 Z 第 i 个运动片段的标准化指标平均值
O	数据集集
D	聚类集
d_i	聚类集第 i 个元素
$d_{OD,ki}$	第 k 个数据集点到 i 个质心的欧氏距离
E_{k-m}	K-Means 聚类精度
μ_i	第 i 个聚类集的重心/均值向量
x_i	第 i 个运动片段无量纲化综合指标
$\overline{x}_{i,\mathrm{k}}$	第 k 个类中，第 i 个运动片段无量纲化综合指标平均值
$x_{j,k}$	第 k 个类中，所有运动片段的第 j 个指标无量纲化集合
$y_{\mathrm{j},k}$	第 k 个类中，所有运动片段的第 j 个指标无量纲化平均值
Y_k	第 k 个类中，所有运动片段所有指标无量纲化平均值
N_k	第 k 个类中，运动片段个数
T_k	第 k 个类中，所有运动片段历经时间之和

$t_{i,k}$	第 k 个类中，第 i 个运动片段历经总时间
T_{total}	汽车运行工况设定时间
N	运动学片段总个数
$E_{k,i}$	第 k 个类中，第 i 个运行片段综合指标值与该类片段综合指标值的差值

五、问题一（数据预处理）

5.1 原始数据分析

附件提供了某城市某一轻型汽车在不同时间段内实际道路行驶采集的 3 个数据文件，其中采样频率为 1Hz，即 1s 采集一次汽车的 GPS 车速、XYZ 轴加速度、经纬度、发动机转速、扭矩百分比、瞬时油耗、油门踏板开度、空燃比、发动机负荷百分比、进气流量数据，共计 13 个数据量，3 个参考文件原始数据总量和采集时间总长如表 5.1 所示。

表 5.1 三个文件原始数据情况

文件名	数据总量（行）	采集总时长(h)
文件一	185725	51.59
文件二	145825	40.51
文件三	164914	45.81

- **十三个变量简要分析**

1）GPS 车速即为汽车行驶速度，为汽车行驶工况研究的重要参数。

2）X 轴加速度、Y 轴加速度、Z 轴加速度这 3 个变量一般是由三轴加速度传感器测取的，表征运动物体的空间加速度，能更为准确的反应物体的运动性质，它的优势在于在不知道运动方向的情况下，计算加速度。由于我们能够获取汽车行驶的 GPS 速度。为了简化计算，本文所有的加速度计算均使用速度的变化率来进行，不使用 X 轴加速度、Y 轴加速度和 Z 轴加速度平方和求解。

3）通过经纬度信息进行定位，发现该汽车主要在福建莆田地区活动。

4）后面 7 个变量皆为发动机相关参数，其中发动机转速和扭矩百分比为发动机的重要参数，发动机转速的高低与汽车车速密切相关，扭矩为曲轴端输出的力矩，力矩的大小与汽车的加速度相关。

附件所提供的汽车行驶的原始数据为数据采集装置直接记录的原始采集数据，由于环境影响和采集误差等因素的存在，导致原始数据中包含部分不良数据和存在一些数据缺失情况，基于此，本文对原始数据进行预处理，对缺失值和逻辑错误数据按照一定原则进行清洗。

在对问题一的五个异常数据类型进行数据预处理之前，根据 13 个数据量的上下限约束，对原始数据进行初步分析，把 3 个文件中 13 个数据量的异常率进行统计，结果如下表所示：

表 5.2 原始数据变量值异常情况统计表

文件名		文件 1			文件 2			文件 3		
变量名	正常范围	异常率	最小值	最大值	异常率	最小值	最大值	异常率	最小值	最大值

GPS 车速	0~350	0%	0	111.5	0%	0	116.6	0%	0	261.4
经度	±180	0%	119.3677	119.6619	0%	0	118.9219	0%	118.968	119.4306
纬度	±90	0%	25.9131	26.0069	0%	0	25.4631	0%	25.3525	26.2384
发动机转速	0rpm~12000rpm	0%	125	2725	0%	175	4450	0%	162	2762
扭矩百分比	0%~200%	0%	0	76	0%	0	94	0%	0	79
瞬时油耗	0L/h~80L/h	0%	0	69.28	0%	0	69.28	0%	0	69.28
油门踏板开度	0%~120%	0%	0%	20%	0%	0%	47%	0%	0%	27.50%
发动机负荷百分比	0~100	0%	9	100	0%	10	92	0%	10	91

我们对 3 个原始数据文件中存在上下限范围约束的 8 个变量的值进行统计分析，3 个文件变量值异常率皆为 0%，但是我们观察到文件 2 中经纬度数据出现了全 0 的现象，而其他时刻的经纬度变化都大致处于一个范围区间内，这意味着出现了错误的定位信息，经纬度为 0 的数据行特征如表 5.3 所示，出现异常的三个时间区间里， GPS 车速都是存在采样值的，且其他变量，如发动机转速和扭矩百分比等数据都是时刻变化的，这意味着经纬度采集值为 0 可能只是经纬度采集模块出现了问题，汽车的行驶是正常的，且经纬度数值不会对后续运动学片段选取和汽车工况构造造成影响，我们不需要把该行数据当做异常数据处理。

表 5.3 三个经纬度为 0 类型情况

类型	开始时间	结束时间	持续时间（s）	GPS 车速(码）	其他变量值
类型一	2017/11/2 11:14:03	2017/11/2 11:14:04	1	48.8	变化
类型二	2017/11/4 10:37:59	2017/11/4 10:38:37	38	0	变化
类型三	2017/11/4 14:45:20	2017/11/4 14:49:59	279	51.5	变化

5.2 不良数据类型分析

按照问题一提供的不良数据主要类型，本文把其归为数据缺失、变速异常、停车数据、持续低速、异常怠速五个类别，其具体表述和处理方式如表 5.4 所示：

表 5.4 不良数据类型表

类型	具体表述	处理方式
数据缺失	由于高层建筑覆盖或过隧道等，GPS 信号丢失，造成所提供数据中的时间不连续	酌情填充
变速异常	汽车加、减速度异常的数据（0 至 100km/h 的加速时间大于 7 秒，紧急刹车最大减速度大于 8 m/s^2）	剔除数据
停车数据	长期停车（如停车不熄火等候人、停车熄火了但采集设备仍在运行等）所采集的异常数据	剔除数据

持续低速	长时间堵车、断断续续低速行驶情况（该时间区域内最高车速小于 10km/h）	按怠速情况处理
异常怠速	怠速时间超过 180 秒（怠速最长按 180s 处理）	超过 180s 部分进行剔除

数据预处理阶段的质量对后续运动学片段提取的准确性和汽车行驶工况构建的代表性影响重大，本文对于各个类型不良数据的具体处理在 5.3 小节中进行了科学的阐述和简要推证，数据预处理步骤设计如下：

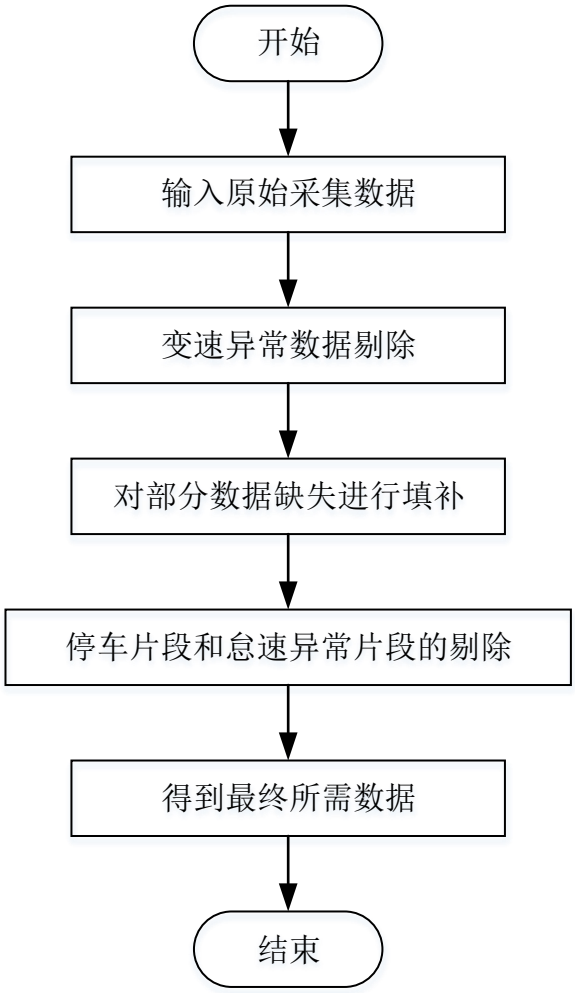


图 5.1 数据预处理步骤设计

数据预处理编程整体的流程步骤如下：

数据预处理具体编程思路：

Step1： 对文档中原始数据中变速异常数据，即对减速度大于 8 m/s^2，加速度小于 3.968 m/s^2，的数据进行剔除，得到新数据 1。

Step2： 对新数据 1 进行扫描，判断上下行数据的时间列是否以 1s 为间隔连续，如果是，继续向下扫描，如果否，认定该处存在数据缺失，定位缺失数据位置并统计连续缺失数据个数，如此重复，直至扫描完整个文档，汇总新数据 1 中缺失数据情况；

Step3： 根据缺失数据情况，通过缺失前后的采样数据，酌情对缺失数据个数小于某一范围的片段进行填补，得到新数据 2；

Step4： 对新数据 2 进行搜索，定位长时间速度持续为 0 片段和怠速运行片段，根据一定

规则判别其中怠速异常片段和停车熄火片段，然后把新数据 2 中怠速异常片段和停车熄火片段分别按一定规定进行剔除，得到最终数据；

5.3 不良数据预处理

5.3.1 变速异常数据处理

1）加速度

加速度为描述物体速度变化快慢的物理量，是汽车行驶状态的一个重要的评价参数，它不能直接通过数据采集设备进行采集，只能利用采集的 GPS 速度，通过以下公式计算求得：

假设汽车在时间 t 到时间 t+T 内做匀变速运动，那么该变速过程的加速度可以用下述式子进行表示：

$$a_t = \frac{v_{t+T} - v_t}{T} \tag{5.1}$$

其中：v_t 和 v_{t+T} 分别为 t 时刻和 t+T 时刻采样的 GPS 车速。

附件提供的原始数据的采样时间为 1s，本文假设采样时间 1s 内的所有运动过程皆为均匀的，那么在 T=1s 内，汽车运动过程的加速度为：

$$a_t = v_{t+1} - v_t \tag{5.2}$$

其中，加速过程，a_t 大于 0m/s^2,减速过程，a_t 小于 0m/s^2。

数据并非都是以采样时间为 1s 连续的，对于出现时间缺失的区间，我们假定其运动过程是均匀变化的，变速过程加速度恒定，那么对于该段时间的加速度利用式子（5.1）进行计算，即缺失时间前后速度差值除以缺失的时间。

2）极限加速度[6]

汽车行径过程中，地面对后轮的摩擦力是驱动汽车的外力，地面对前轮的摩擦力是汽车运动的阻力，如果忽略前轮所受到的摩擦阻力，由于后轮和地面之间的摩擦系数一定，并且后轮对地面的正压力有限，致使汽车存在极限加速度，超出极限加速度的运行状态是不存在的，因而对于原始采集数据中变速异常数据需要进行剔除。

- 普通轿车一般情况下，题目给出的参考变速限制为：：

①最大加速度限制：百公里加速的时间必须大于 7 秒，即最大加速度不超过 3.968 m/s^2（或 14.286 km/h/s。）

②最大减速度限制：紧急刹车最大减速度必须控制在在 7.5~8 m/s^2，即最大减速度不超过 8 m/s^2（或 28.8 km/h/s）。

综上，汽车加速度的上下限为：-8 m/s^2≤a_t≤+3.968 m/s^2（或-28.8 km/h/s≤a_t≤+14.286 km/h*s）。基于该极限加速度的限制，对原始采集数据中变速异常的片段进行搜寻并予以剔除。

3）变速异常数据统计

首先，我们分别对原始数据文件夹中的文件 1、文件 2、文件 3 进行扫描，根据式子（5.1）和（5.2）计算每个采样时间的加速度的大小，

然后根据汽车加速度的上下限限制，对 3 个文件中加速度异常的数据进行定位，统计每个文件中变速异常数据个数，然后对该类错误数据予以剔除，并更新 excel 中变速异常

处理后数据个数，整体情况如下表 5.5 所示：

表 5.5　变速异常数据处理情况总表

文件名	加速异常个数（行）	减速异常个数（行）	变速异常数据个数（行）	剩余数据（行）
文件 1	26	2	28	185697
文件 2	329	81	410	145415
文件 3	115	38	153	165761

从表 5.5，明显可以看出：在最大加速度+14.286 km/h/s 的限制下，文件 1 中总共有 26 个加速异常的片段，文件 2 中总共有 329 个加速异常的片段，文件 3 中总共有 115 个加速异常的片段。在最大减速度-28.8 km/h/s 的限制下，文件 1 中总共有 2 个减速异常的片段，文件 2 中总共有 81 个加速异常的片段，文件 3 中总共有 38 个加速异常的片段，对文件中存在的变速异常数据均剔除。

综上，文件 1 总计搜寻到 28 个变速异常数据，剩余数据行数为 185697 行，文件 2 总计搜寻到 410 个变速异常数据，提出该类异常数据后，剩余数据行数为 145415 行，文件 3 总计搜寻到 153 个变速异常数据，提出该类异常数据后，剩余数据行数为 165761 行，经过变速异常处理的 3 个原始文件，分别更名为文件 1’，文件 2’，文件 3’。

5.3.2 缺失数据处理

1）数据缺失产生原因

汽车的速度信息，主要通过 GPS 测速仪进行采集，GPS 测速的原理主要是以高速运动为卫星的瞬时位置作为已知的起算数据，卫星通过不断的发送自身的时间信息和星历参数，GPS 接收器通过卫星数据采用空间距离后方交会方式计算汽车每一采样前后的经纬度坐标，除以采样时间，即为汽车的运行速度。由于经纬度定位计算都需要包含该位置三维地心坐标和卫星接收器的时间差，故 GPS 测速至少需要 4 颗卫星的观测来进行最终的计算，而 GPS 卫星本身是属于微波传输，卫星信号从几万公里以外的太空传来，其信号强度通常只有普通无线电广播信号的二十分之一不到，这就意味着它很容易被阻挡，如图 5.2 和图 5.3 所示，汽车通过有高层建筑或或者隧道等遮蔽物时，卫星信号都可能会被遮挡住，从而造成采集数据的缺失。

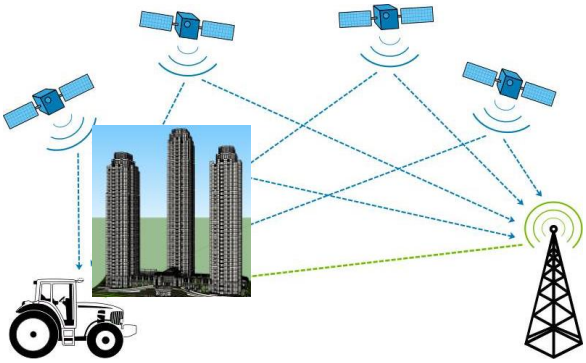


图 5.2 高层建筑遮挡导致 GPS 信号弱

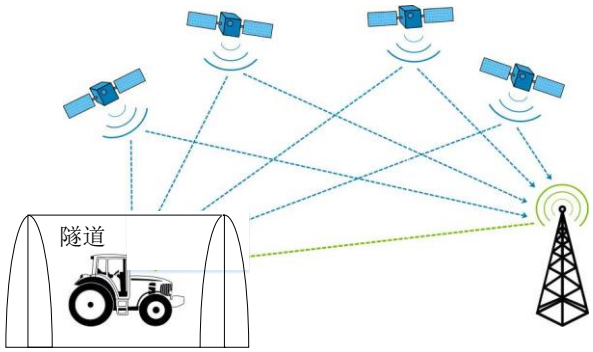


图 5.3 过隧道导致 GPS 信号弱

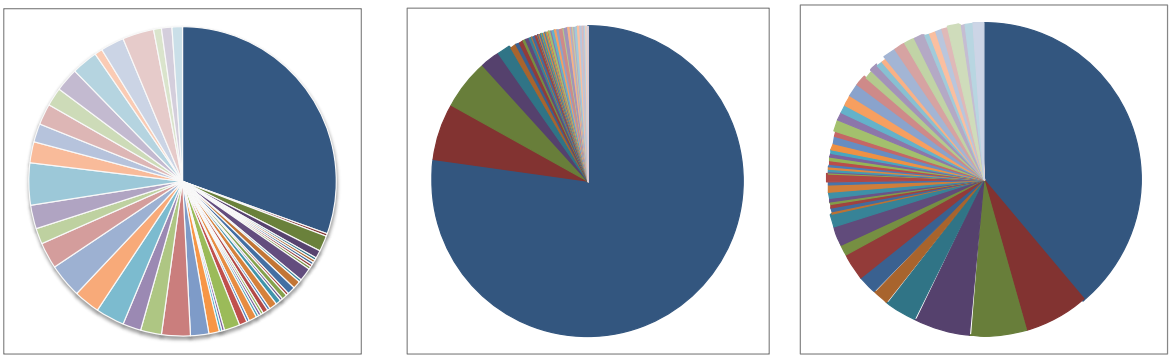
2）数据缺失情况分析

为了获得数据缺失的情况，我们对经过变速异常处理过后的文件 1’到文件 3’的数据进行扫描，时间列上下不以 1s 的采样时间连续的数据行进行定位，记做数据缺失，并

从该位置向下计数，统计连续缺失的数据个数，具体情况如下所示：

表 5.6 数据缺失总体情况表

文件名	缺失数据总数	连续缺失个数最小值	连续缺失个数最大值	缺失时间小于 60 秒		缺失时间为 1 秒	
				行数	百分比	行数	百分比
文件 1’	723	1	38607	364	50.35%	111	15.35%
文件 2’	2374	1	51303	2254	94.95%	1743	73.42%
文件 3’	1096	1	42942	872	79.56%	341	31.11%



a) 文件 1’　　b) 文件 2’　　c) 文件 3’

图 5.4 数据连续缺失 60s 内分布饼状图

（占比最多深蓝色部分为缺失时间 1s 的情况）

文件 1’，文件 2’和 文件 3’缺失的数据总数分别为 723，2374 和 1096，其中 3 个文件均存在超过 600 分钟（10 小时）的长时间数据连续缺失现象， 其中文件 2’的连续时间缺失甚至高达 855 分钟（14.25 小时），我们对文件 1’中连续缺失为 38607s 的数据进行查看，发现缺失的时间段为 2017 年 12 月 19 日 19 点 21 分 57 秒到 2017 年 12 月 20 日 6 点 5 分 22 秒，文件 2’中连续缺失的 51303s 数据缺失时段为 2017 年 11 月 1 日 20 点 5 分 2 秒到 2017 年 11 月 2 日 10 点 20 分 3 秒，文件 3’中连续缺失的 42942s 数据缺失时段为 2017 年 12 月 5 日 20 点 31 分 4 秒到 2017 年 12 月 6 日 8 点 48 分 37 秒，这些皆属于夜晚时间，那么我们由此可以推断长时间连续数据缺失极大可能是由于驾驶员夜间停工休息，导致汽车处于待机状态，无行驶信息。

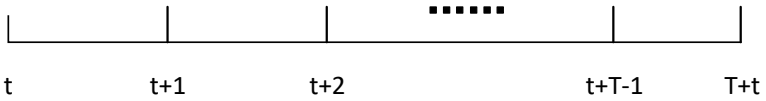
结合图 5.4 数据连续缺失 60s 内分布饼状图和表 5.6 中相关数据，我们可以看到：3 个文件夹数据缺失最多的情况为间隔时间 1s 的数据缺失，其中文件夹 2’中连续 1s 缺失出现情况更是达到了总缺失情况的 73.42%，定位 1s 缺失的时间区段，发现有些时间区间内大量出现短时间(1s~2s）数据缺失现象，这可能是由于车辆行经城市繁华地带，高层建筑密集，连续高楼对 GPS 信号出现短暂遮挡，造成数据连续间断，对于这类现象，我们需要对于缺失数据进行填补，来保证采集数据的连续性，为后续运动学片段的选取服务。

对于数据缺失较长片段，汽车运行情况具有多样性，比如，对文件 1’中缺失时间为 21s 的某一个运动过程进行简要分析，缺失前后汽车速度均为 87.6 km/h*s，在这缺失的 21s 过程里，可能汽车一直保持 87.6 km/h*s 的速度做匀速运动，可能先进行了一段加速

过程，然后减速到了原始速度，也可能先减速了一段时间，然后又加速到达 87.6 km/h*s，亦可能是一些更复杂的运动过程，在这段较长缺失时间内，汽车运行情况复杂多样，因而我们无法对数据连续缺失太长的运动过程进行科学推断和还原，故我们不对较长时间缺失的数据进行填补。

3）缺失数据填补

短时间内汽车的运动状态与其缺失数据前后具有一定的关联性，我们可以利用缺失前后运动状态对缺失时刻数据进行还原，补全缺失数据信息。对于数据缺失部分，我们认为该过程中的运动状态是均匀变化的，故可以利用缺失前后的运动数据信息，对缺失的每一采样周期做均匀变化处理，数据填补方式如下所示：



$$s_{t+i}=\frac{i}{T}(s_{T+t}-s_t)+s_t \qquad (5.3)$$

其中，s_t 为缺失前 1s 汽车的行驶信息，s_{T+t} 为缺失后 1s 汽车的行驶信息，s_{t+i} 为缺失的第 i 秒汽车的行驶信息，T 为连续缺失的总时间，i 取值从 1,2,3，…T 。

我们以均值代替缺失数据值进行数据填补，随着缺失时间的增加，上述数据替代将变得越来越不科学，那么，为了控制填补数据对数据样本的误差干扰，我们只对缺失时间为 1s（单体占比最大）的数据进行数据填补，1s 的时间缺值小，通过 1s 前后的数据信息的均值当做该时段的运行数据，填补数据与实际情况的偏差极低，由此填补的数据，准确度很高。

对 3 个文件分别填补，最终填补情况如下：

表 5.7 数据填补情况

文件名	填补数据行数	填补后数据总量
文件 1’	111	185808
文件 2’	1743	147158
文件 3’	341	166102

从表 5.7 中可知，文件 1’、 文件 2’和 文件 3’填补的数据行数分别为 111、1743 和 341，填补后，数据总量变为 185808、147158 和 166102。数据填补处理过后三个文件夹分别更名为文件 1*, 文 2 件*, 文件 3*。

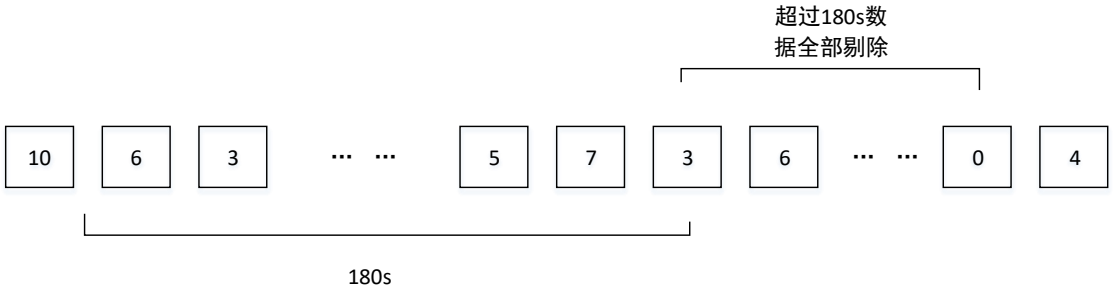
在数据处理中，本文采用首先对变速异常片段清理再进行数据缺失的填补，这是为了避免缺失数据填补后又被变速异常给剔除而导致操作多余的情况的产生。

5.3.3 异常怠速和停车状态处理

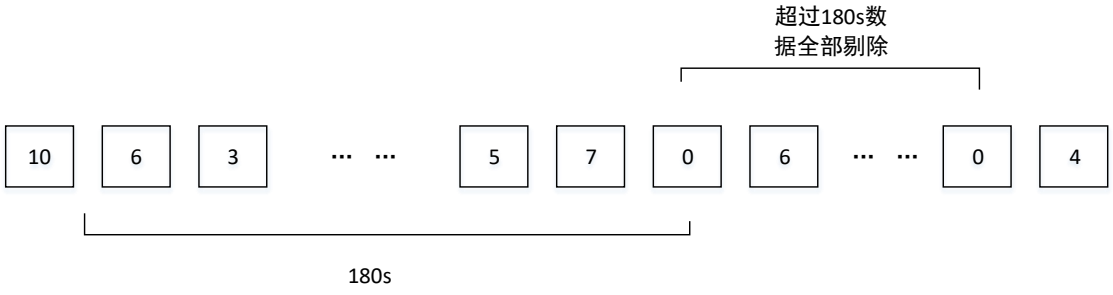
长期停车意味着在很长的一段时间区间内，汽车的 GPS 车速一直保持 0 不变，而怠速状态是指在不大于 180s 的一段时间区间内，汽车的最高车速小于 10km/h 的运行情况，停车和怠速存在相互交叉的定义范围，即 180s 内速度一直保持为 0 不变，其既可以看做停车，又可以当做怠速，为了分别对异常怠速和长期停车两种情况得数据进行删减，我们需要对怠速异常和停车状态进行区分判别，其判别思路和两种情况下的数据处理如图 5.5 和图 5.6 所示，判别主要以怠速过程不能超过 180s 这个条件进行：

1）异常怠速

异常怠速主要分为两种情况，第一种：从汽车进入怠速（车速开始小于 10km/h）开始，持续到 180s，如果此时 GPS 车速不等于 0，而是大于 0 小于 10km/h 的某一车速，那么认为此时出现怠速异常现象，对 180s 之后的怠速数据清除。第二种：从汽车进入怠速（车速开始小于 10km/h）开始，持续 180s，如果此时 GPS 车速等于 0，但是在其附近的时间内，存在大于 0 而小于 10km/h 的任一车速，那么也当做怠速异常处理，对怠速 180s 之后的怠速数据清除。



（a）



（b）

图 5.5 怠速异常数据处理

2）停车状态

停车状态我们的判定方式是从汽车进入怠速（车速开始小于 10km/h）开始，持续到 180s，如果此时 GPS 车速等于 0，且该 0 值处于很长一段全 0 区间里，那么我们认定此时为停车状态，对于停车状态，汽车首先存在一个由停车到熄火的过程，之后汽车又存在一个再次启动到起步过程，根据前文的模型假设中汽车启动之后，车主 5s 后起步和汽车停车到熄火存在 5s 间隔这两个条件，我们前后保留 5s 为 0 的数据行，然后把中间数据全部删除。

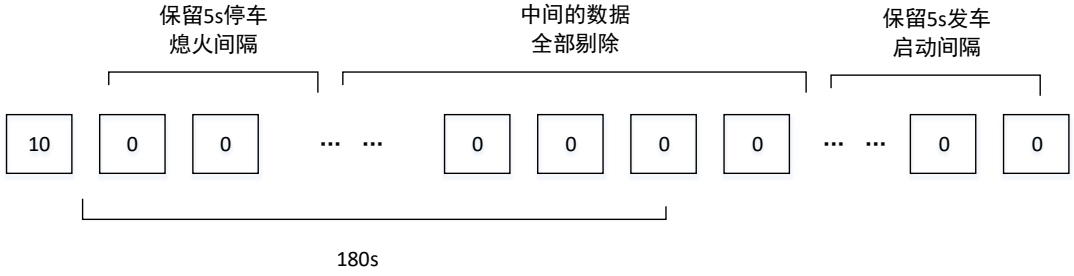


图 5.6 停车状态数据处理

对数据填充处理后得到的文件 1*，文 2 件*，文件 3*数据按上述规则进行异常怠速和停车状态的处理，处理结果如表 5.8 所示。

表 5.8 异常怠速和停车状态处理情况

文件名	删除数据行数	删除后数据总量
文件 1*	8048	177760
文件 2*	5780	141378
文件 3*	11800	154302

通过对异常怠速和长时间停车数据的处理，对于文件 1*，文 2 件*，文件 3*，我们分别删除了 8048 行，5780 行和 11800 行数据，处理过后，三个文件夹剩余的数据总量分别为 177760 行，141378 行和 154302 行。

5.4 数据预处理总结果

考虑问题一中所提出的五种不良数据类型，我们对原始数据进行了变速异常数据处理、数据填充处理及异常怠速和停车状态处理三个阶段，最终得到的用于后续运动学片段提取和汽车工况构造的有效数据，数据预处理结果如下表所示：

表 5.9 数据预处理过程结果表

行数	文件 1	文件 2	文件 3
原始数据	185725	145825	164914
变速异常数据处理	-28	-410	-153
缺失数据填补处理	+111	+1743	+341
异常怠速和停车状态处理	-8048	-5780	-11800
数据总体变化	-7965	-4447	-11612
数据变化百分比	4.29%	3.05%	7.04%
数据预处理后有效数据	**177760**	**141378**	**154302**

经过三个阶段的数据填补和剔除，对文件 1，文件 2，文件 3 数据进行更新，新数据仍以原名命名。根据表 4.9 可以看到，文件 1 数据总体减少了 7965 行，减少比例为 4.29%，数据预处理后留下的有效数据为 177760 行，文件 2 数据减少行数相对最少，仅 4447 行，减少比例为 3.05%，数据预处理后留下 141378 行有效数据，文件 3 数据的减少行数最多，删减数据高达 11612 行，减少比例达到 7.04%，剩余的有效数据行数降至 154302 行。

六、问题二（运动学片段提取）

6.1 运动学片段定义

为了模拟汽车在实际行驶过程中的频繁启动，加速，减速，匀速等行驶状态，把汽车从一个怠速状态开始到下一个怠速状态开始间的行驶过程定义为一个运动学片段，其通常是由一个怠速部分和一个运动部分构成，其中运动部分需要包含至少一个加速过程，一个减速过程及一个匀速过程，运动学片段的长短不定，可以短至几秒，也可以长达几个小时，它的划分只与下一个怠速状态出现与否相关[7]。运动学片段的示意图 6.1 所示：

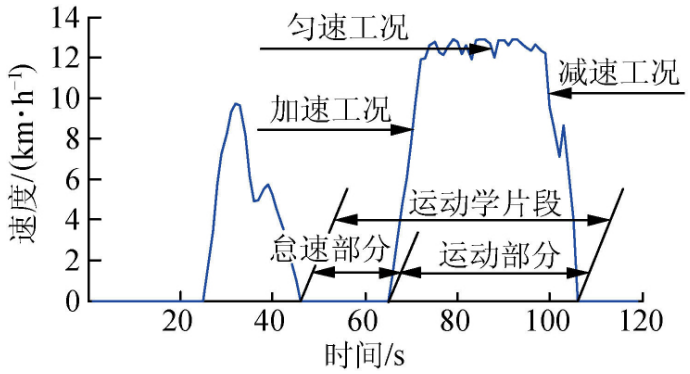


图 6.1 运动学片段示意图

汽车行驶过程的四种运动状态定义区分：

（1）怠速过程：汽车停止运动，但发动机保持最低转速运转的连续过程，汽车车速在该运动过程内最高不超过 10km/h。

（2）加速过程：汽车不属于怠速过程（长时间维持最高车速不超过 10km/h），且汽车加速度大于 0.1m/s^2，小于最大加速度限制的连续过程。

（3）减速过程：汽车不属于怠速过程（长时间维持最高车速不超过 10km/h），且汽车加速度小于-0.1m/s^2，大于最大减速度限制的连续过程。

（4）匀速过程：汽车加速度的绝对值小于 0.1m/s^{2}非怠速的连续过程。

6.2 运动学片段筛选规则

结合车辆行驶工况的实际情况，考虑到运动学片段选择的合理性和有效性，我们参考相关资料，对运动学片段的筛选定下了以下规则：

（1）满足运动学片段的定义；

运动学片段必须包含怠速过程，加速过程，减速过程，匀速过程这四个过程，一般来说只要汽车从怠速过程过渡到 GPS 车速超过 10km/h 以上，那么就会存在加速情况，此时只需要判别其加速度有无大于加速过程的最低加速度限制 0.36km/h/s（0.1 m/s^2）即可，如果加速度大于最低限制，那么该段时间内汽车存在加速过程，同时存在加速过程也就意味着在进入下一怠速运动前，汽车的 GPS 车速必将减到 10km/s 以下，减速情况必然会出现，此时只需要判别其减速的加速度是否小于-0.1m/s^2，如果小于，那么该段时间内，汽车存在减速过程。对于匀速过程的判定，只需要在怠速过程外，判别相邻时间内的汽车运动加速度有无出现-0.36km/h/s 到 0.36km/h/s 的过程，如果有，那么该段时间内，存在匀速过程，满足这些条件的相邻怠速区间，即为初始的运动学片段。

（2）运动学片段的时长大于 20s；

运动学片段定义的划分是从一个怠速状态开始到下一个怠速状态开始的一段行驶过程，其片段长度具有随机性，几秒到几小时不等。我们后续需要使用运动学片段构造汽车的行驶工况，那么我们要求提取的运动学片段必须具有一定代表意义，能够较为准确的描述汽车行驶过程中的四个运动状态的分布及相关特征，然而，如果运动学片段太短的话，它的四个运动状态占比的时间都很短，运动状态变化很快，匀速过程短，如果对片段时长不加以限制，可能存在，一些由 GPS 车速采集的灵敏度缺陷（表现为短时间内速度实时采集出现滞后等现象），生成一些因为采样误差而产生的符合运动学片段定义的误差区间被记入运动学片段，其对于后续汽车工况的构建是无法采用的，故需要对运动学片段的时长加以限制，根据参考文献[8]，我们采用 20s 作为运动片段时长最小限制。

（3）运动学片段内数据缺失率（GPS 数据缺失率小于 10%）；

运动学片段选取对后续的汽车工况模拟的准确性将产生直接的影响，后续处理中我们会对运动学片段进行更进一步分析，提取重要的特征值。汽车行驶的相关特征主要为：时间、速度、路程，结合运动片段具有的 4 种运行状态，特征会进行更加细化的区分，然而，运动学片段缺失数据占比太多的话，我们将无法得到该片段内汽车准确的运动过程，进而也无法提炼出能表征该运动过程的特征参数，所以，我们需要对运动学片段内数据缺失率加以限制，根据参考文献[8],我们规定，当初始运动学片段的数据缺失率高于 10%，那我们就认为该运动学片段缺乏完整性，将其剔除。

我们开始还考虑对于连续数据缺失的时长加以限制，主要是考虑可能出现运动学片段时长很长，数据缺失率小于 10%，满足规则（3），但是数据缺失都是连续存在的现象，比如有一个时长为 10000s（166min）的运动学片段，其中存在连续 900s（15min）的数据缺失，除此之外其他间断的数据缺失总和小于 100s，那么，这个片段是满足规则（3）的缺失率限制的，但是我们对于车辆行驶情况有连续 15min 的空白，这段时间内，运动情况复杂多样，甚至可能存在怠速状态，会对运动学片段的划分造成重大影响，但是后来我们通过对初始运动学片段中的连续时间缺失时长进行统计，发现三个文件满足规则（3）限制的运动学片段，皆没有连续超过 300s 的数据空缺，这个缺失时间长度的运动状态对于长时间运动而言，影响甚微，故我们最终不把连续数据缺失的时长限制放在运动学片段筛选的规则中。

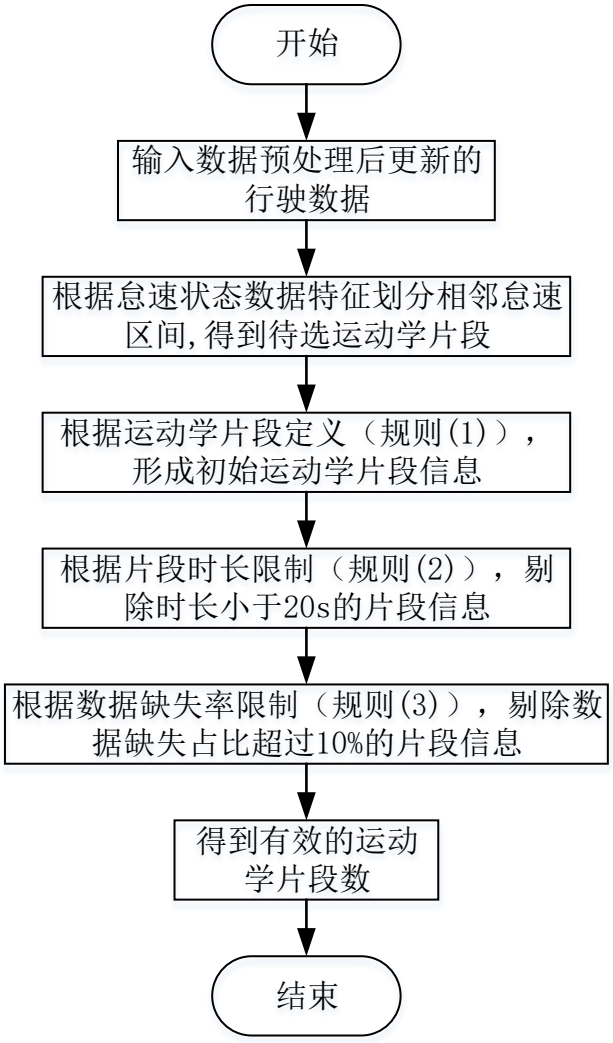


图 6.2 运动学片段筛选步骤

运动学片段筛选步骤：

Step1：扫描文件夹中各行 GPS 车速，判断此时是否处于怠速运行，如果是，记做开始，向下扫描，直至出现下一个怠速状态时停止，把这个时间区间记录下来，重复操作，直至找到所有相邻怠速区间；

Step2:对每一个相邻怠速区间,判断其是否存在加减速过程,即加速度绝对值大于 0.1m/s^2，把不满足要求的区间剔除；

Step3：再对区间内怠速过程外，是否存在匀速过程进行判断，即加速度绝对值小于 0.1m/s^2），把不满足要求的区间剔除，得到初始运动学片段；

Step4：对初始的运动学片段进行处理，判断片段总时长是否大于 20s，把不满足最低时长限制的片段剔除；

Step5： 计算剩余片段内的 GPS 车速的数据缺失率，如果存在 10%以上的数据缺失，那么把该运动学片段剔除，遍历处理，得到最终的有效运动学片段；

6.3 运动学片段提取结果分析

通过前文提出的运动学片段筛选步骤，我们首先确立相邻怠速区间，得到待选运动学片段，然后，根据运动学片段的定义，获取符合运动学片段定义的初始片段数，然后进入

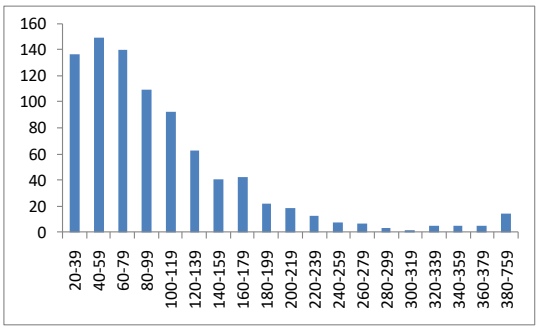
有效片段筛选阶段，我们进行了两步处理，一是对片段时长小于 20s 的片段进行剔除，二是对数据缺失率大于 10%的片段进行筛除，经过这样筛选处理之后得到的运动学片段为有效的运动学片段， 用于第三问汽车行驶工况构建。

表 6.1 运动学片段提取结果

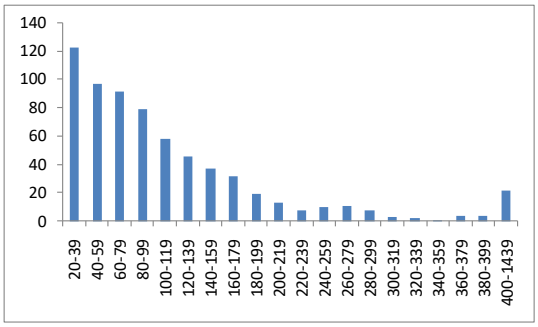
个数	文件 1	文件 2	文件 3
待选运动学片段	1847	1522	1440
初始运动学片段	1253	921	913
片段时长限制筛选后	1196	843	829
数据缺失率限制筛选后	876	668	560
最终得到的有效运动学片段	**876**	**668**	**560**

运动学片段提取结果如表 6.1 所示，从表中，我们可以看到，通过对相邻怠速区间的确定，文件 1、文件 2 和文件 3 分别得到 1847,1522，1440 个待选运动学片段，然后，根据运动学片段定义，把待选片段中不包含加速，减速，匀速过程的片段筛选掉，3 个文件分别筛选掉了 594,601,527 个片段，该步处理后，我们得到 1253,921,913 个初始运动学片段，之后，再进行片段时长限制筛选和数据缺失率限制筛选，分别剔除 377,253,353 个片段后得到最终的有效运动学片段，3 个文件最终有效的运动学片段为 876,668 和 560 个。

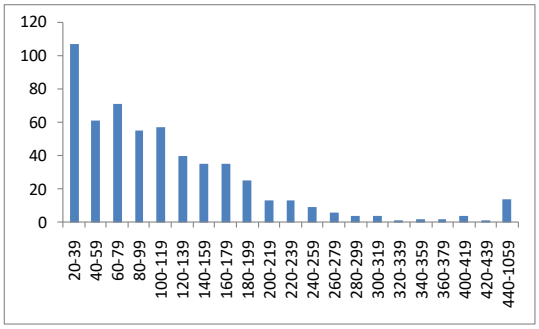
相较初始运动学片段个数，我们通过新增添两条限制规则，3 个文件分别减少了 30%，27.5%和 38.66%的不良运动学片段，剩余的运动学片段皆具有一定的代表性，能够较为准确的反映汽车的行驶情况，能有效参与问题三汽车行驶工况的构建，其中文件 1 中最终有效运动学片段长度由 20s 到 759s 之间不等，文件 2 中最终有效运动学片段长度由 20s 到 1439s 之间不等，文件 3 中最终有效运动学片段长度由 20s 到 1059s 不等，各个文件片段的具体长度分布情况于图 6.2 中可见。



(a) 文件一



(b) 文件二



（c）文件三

图 6.2 有效运动学片段长度分布直方图

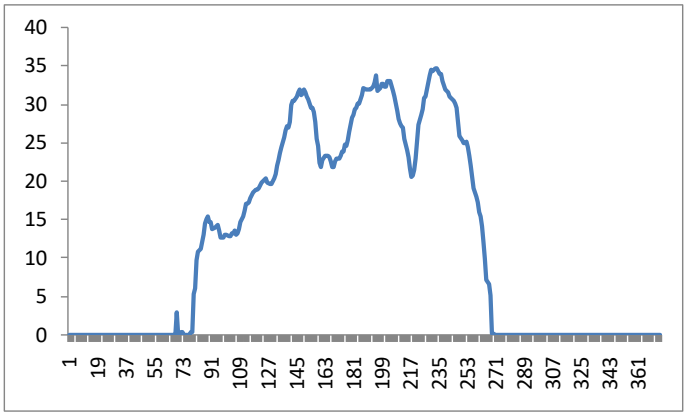


图 6.3 文件 2 中某一时长 323s 的运动学片段

通过查看 3 个文件中有效的运动学片段，我们发现不同运动学片段反映的交通状况可能是一致的。不同的时间、不同的地理位置和公路类型会出现相同的片段，有时候繁忙的高速公路上的片段特征可能和拥挤的城市中的片段完全一致。将这些片段类型和交通状况联系起来，针对性地分析符合低速、中速和高速运动形态，并在此基础上构建工况，这种思路是完全合理的。

七、问题三（汽车行驶工况构建）

通过问题二的求解，运动片段已经从预处理后的采集数据中划分了出来，接下来就要开始对运动片段包含的汽车运动信息进行提取和分析，用更为具体的特征描述运动片段所包含的隐含内容。因此在构建汽车运行工况模型前，本文首先建立汽车运动特征评估体系，然后再构造有关模型和算法得到汽车运行工况。

7.1 汽车运动特征评估模型建立

汽车运动片段是描述汽车从一个怠速阶段开始到下一个怠速阶段开始，且中间包含了加速、减速和匀速过程，这一段时间内汽车运动的速度-时间曲线。在问题二的求解中，本文从经预处理的采集数据附件 1～3 中分别划分出……个运动片段，但是直接观察这些速度-时间曲线或者数据并不能够全面地提取出这些运动片段的全部信息。为了能够更加直观地读出这些运动片段所包含的大部分信息，需要给汽车运动特征设定一系列指标，并且通过这些指标的值能反映出汽车的运行信息。

为了较为科学和全面地选取指标，本文首先结合物理中运动学的描述方法，将指标分为时间、速度和路程这三大类别；接着根据汽车运动片段特点的定义，提取加速、减速、匀速、怠速这四个运行状态。结合时间、速度、路程和加速、减速、匀速、怠速，可以得到以下 11 个描述汽车运动特征的指标，列表如表 7.1：

表 7.1 汽车运动特征指标表

序号	符号	名称	单位
1	T	历经总时间	s
2	v_e	平均速度	Km/h
3	v_{er}	平均行驶速度	Km/h

4	v_{std}	速度标准差	Km/h
5	$v_{\max}$	最大速度	Km/h
6	$A_{a,\max}$	最大加速度	m/s^2
7	$A_{b,\max}$	最大减速度	m/s^2
8	$A_{a,e}$	平均加速度	m/s^2
9	$A_{d,e}$	平均减速度	m/s^2
10	A_{std}	加速度标准差	m/s^2
11	S	运行总路程	Km

使用上述数据可以较全面描述一个运动片段的运动信息，但当需要描述多个运动片段的综合运动特征时，需要考虑数据分布的问题，所以本文再设置 4 个具有统计学意义的指标，统计学指标如表 7.2：

表 7.2 汽车运行统计学指标表

序号	符号	名称
1	R_i	怠速时间比
2	R_a	加速时间比
3	R_b	减速时间比
4	R_b	匀速时间比

其中总路程用每秒路程累计叠加表示，因为采样时间间隔仅为 1s，且汽车速度不能突变，所以本文将汽车每秒的运动简化为一个匀变速运动，于是行驶路程这一指标的计算公式如下：

$$S=\sum_{i=1}^{n-1}\frac{\left(v_{i+1}+v_i\right)}{2} \tag{7.1}$$

式中，n表示该运动片段总点数。

时间类指标的计算公式如下：

$$t_i=n_i \tag{7.2}$$

$$t_a=n_a \tag{7.3}$$

$$t_d=n_d \tag{7.4}$$

$$t_e=n_e \tag{7.5}$$

其中，n_i、n_a、n_d、n_e分别表示怠速点个数、加速度点个数、减速度点个数、匀速点个数。

速度类指标的计算式如下：

$$v_e=\frac{S}{T} \tag{7.6}$$

$$v_{er}=\frac{S}{T-t_i} \tag{7.7}$$

$$v_{std}=\sqrt{\frac{\sum_{i=1}^{n}\left(v_i-v_e\right)^2}{n-1}} \tag{7.8}$$

$$v_{\max}=\max\left\{v_1,v_2,\cdots,v_n\right\} \tag{7.9}$$

$$v_{\min}=\min\left\{v_1,v_2,\cdots,v_n\right\} \tag{7.10}$$

加速度指标的计算式如下：

$$A_{\mathrm{a,max}}=\max\left\{a_1,a_2,\cdots,a_n\right\} \tag{7.11}$$

$$A_{\mathrm{b,max}}=\min\left\{a_1,a_2,\cdots,a_n\right\} \tag{7.12}$$

$$A_{\mathrm{d},e}=\frac{sum\{a_i\mid a_i>0.1,i=1,2\cdots,n\}}{T_d} \tag{7.13}$$

$$A_{std}=\sqrt{\frac{\sum_{i=1}^{n}a_i^2}{n-1}} \tag{7.14}$$

统计学指标的计算式为：

$$R_i=\frac{T_i}{T} \tag{7.15}$$

$$R_a=\frac{T_a}{T} \tag{7.16}$$

$$R_d=\frac{T_d}{T} \tag{7.17}$$

$$R_e=\frac{T_e}{T} \tag{7.18}$$

式中 R_i、R_a、R_d 和 R_e 分别表示怠速时间比、加速时间比、减速时间比和匀速时间比。

7.2 汽车运行工况的构建方法

7.2.1 基于主成分分析的运动特征指标重构

本文一共选取了 11 个运行特征指标和 4 个统计学指标，总共 34 个指标。那么每一个运动片段都将由 15 个指标的值共同表示和描述。显然这种表示方法维度较高，而且某些指标之间存在一定的线性关系，可以相互表示，因而不需要保留所有指标。所以，本文拟对 15 个指标进行降维处理，转化成能够高度涵盖这 15 个指标包含的信息但个数却远远降低的综合指标。

主成分分析法是一种常用的降维方法，其主要思想是用少数综合指标取代原指标，进而简化数据结构和降维。假设共有 p 个指标，则每个运动片段的 p 个指标可以用一个 p 维向量表示，假设这个向量为 $b_i=\left[b_1,b_2,\cdots,b_p\right]$，那么 m 个运动片段的指标向量组合在一起可以构成一个指标矩阵 b：

$$b=\begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,\mathrm{p}} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,\mathrm{p}} \\ \vdots & \vdots & \ddots & \vdots \\ b_{\mathrm{m},1} & b_{m,2} & \cdots & b_{m,\mathrm{p}} \end{bmatrix} \tag{7.19}$$

主成分分析法的主要思路是找到主成分分量Y，Y与指标矩阵b之间存在如下关系[9]：

$$Y=b\cdot L \tag{7.20}$$

$$L=\begin{bmatrix} l_{1,1} & l_{2,1} & \cdots & l_{\mathrm{m},1} \\ l_{1,2} & l_{2,2} & \cdots & l_{\mathrm{m},2} \\ \vdots & \vdots & \ddots & \vdots \\ l_{1,\mathrm{p}} & l_{2,p} & \cdots & l_{\mathrm{m},\mathrm{p}} \end{bmatrix} \tag{7.21}$$

其中，L为系数矩阵，$Y_i=b_{i,1}\cdot l_{1,i}+b_{i,2}\cdot l_{2,i}+\cdots+b_{i,\mathrm{p}}\cdot l_{\mathrm{p},i}$。

且满足如下条件：

$$\begin{cases} L_i\cdot L_i^T=1; \\ Cov(Y_i,Y_J)=0,\mathrm{i}\neq \mathrm{j},\mathrm{i}、\mathrm{j}=1,2,\cdots m \\ Var(Y_1)\geq Var(Y_2)\geq\cdots\geq Var(Y_m) \end{cases} \tag{7.22}$$

Y是经过主成分分析后的主成分向量，将Y_i从大到小排列，数值越大表明主成分相关性越强。

主成分分析的具体步骤如下：

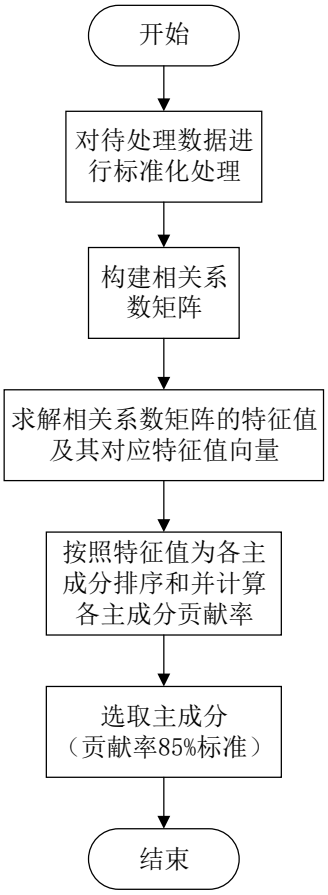


图 7.1 主成分分析流程图

Step1：标准化处理。需要进行数据标准化处理是因为，指标矩阵内包含有多种数据，比如时间数据、路程数据和速度数据，且有些数据的数量级差距较大，因此需要对数据进行标准化处理，处理后得到标准化指标矩阵 Z ：

$$Z=\begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,\mathrm{p}} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,\mathrm{p}} \\ \vdots & \vdots & \ddots & \vdots \\ z_{\mathrm{m},1} & z_{\mathrm{m},2} & \cdots & z_{\mathrm{m,p}} \end{bmatrix} \tag{7.23}$$

其中，矩阵内元素 $z_{\mathrm{i,j}}$ 计算式为：

$$z_{\mathrm{i,j}}=\frac{b_{\mathrm{i,j}}-u_j}{\sqrt{\sigma_j}} \tag{7.24}$$

$$b_j=\left\{b_{i,j}\mid i=1,2,\cdots,m\right\} \tag{7.25}$$

$$u_j=E(b_j) \tag{7.26}$$

$$\sigma_j=D(b_j) \tag{7.27}$$

Step2：相关系数矩阵的构建。构建方法是利用标准化指标矩阵 Z 的元素计算出相关系数矩阵 R 内各元素的值，相关系数矩阵又称为样本矩阵的协方差矩阵，其计算式如下：

$$R=\begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,\mathrm{l}} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,\mathrm{l}} \\ \vdots & \vdots & \ddots & \vdots \\ r_{\mathrm{q},1} & r_{\mathrm{q},2} & \cdots & r_{\mathrm{q,l}} \end{bmatrix} \tag{7.28}$$

相关系数矩阵内元素 $r_{\mathrm{i,j}}$ 计算式为：

$$r_{\mathrm{i,j}}=\frac{\sum_{k=1}^{m}\left(z_{k,i}-\overline{z_i}\right)\left(z_{k,\mathrm{j}}-\overline{z_j}\right)}{\sqrt{\sum_{k=1}^{m}\left(z_{k,i}-\overline{z_i}\right)^2\sum_{k=1}^{m}\left(z_{k,\mathrm{j}}-\overline{z_j}\right)^2}} \tag{7.29}$$

Step3：相关系数矩阵特征值和特征向量的求解。求解 $\det(\lambda E-R)=0$，得到相关系数矩阵 R 的 p 个特征值 $\lambda_1,\lambda_2,\cdots,\lambda_p$ 和特征向量，将这 m 个特征值按照从大到小的顺序排列。特征值越大表示该成分相关系数越强，根据特征值大小关系，可以求出各主成分的贡献率 φ，贡献率计算公式如下：

$$\varphi_i=\frac{\lambda_i}{\sum_{i=1}^{m}\lambda_i} \tag{7.30}$$

假设选取了 f 个主成分，这样标准化指标矩阵 Z 的列数将缩减为 f 个，所得到的主成分指标矩阵 G 为：

$$G=\begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,f} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,f} \\ \vdots & \vdots & \ddots & \vdots \\ g_{\mathrm{q},1} & g_{\mathrm{q},2} & \cdots & g_{\mathrm{q},f} \end{bmatrix} \tag{7.31}$$

7.2.2 基于 K-Means 聚类的运动片段分类

K-Means 聚类即 K 均值聚类，是一种无监督学习、迭代求解的聚类算法。该算法具有实现简单、参数设置简单的优点，尤其在待分类点分类特征明显的时候效果较好。一般 K-Means 聚类判断指标有两类，一类是相似度，另一类是距离。对于不同的聚类对象，有不同的指标计算方法。当聚类对象为向量时，其聚类指标为：

$$s(x_i, x_j) = \frac{x_i \cdot x_j^T}{\|x_i\| \|x_j\|} \tag{7.32}$$

当聚类对象为点时，聚类指标为点距：

$$d(x_i, x_j) = \sqrt{\sum \left(x_i - x_j \right)^2} \tag{7.33}$$

K-Means 聚类的基本流程如下：

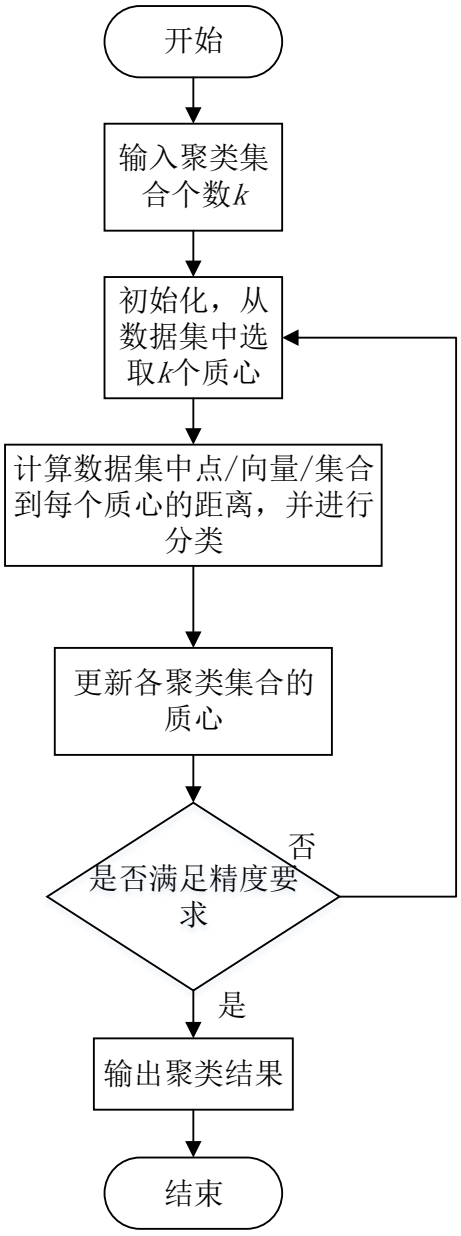


图 7.2 K-Means 聚类流程图

Step1：设定聚类集合数目、选取初始聚类中心。将输入数据分为聚类集合 D 和数据集 O 两部分，初始选定聚类集合数目后，从输入数据中随机或者按照一定规则选取 k 个聚类中心。
Step2：对数据集点进行分类。计算数据集中每个剩余运动片段到各个聚类中心的距离，并进入到距离最近的聚类集合。结合本题的有关参数设定，第 k 个数据集点到第 i 个质心的距离表达式为：

$$d_{OD,ki}=\sqrt{\sum_{j=1,y_i\in D}^{f}\left(g_{k,j}-d_i\right)^2} \tag{7.34}$$

Step3：更新质心。当每个类别有新的运动片段加入时，都要重新计算重心，更新质心。然后继续按照 step2 分配数据集中剩余的运动片段。
Step4：判断迭代是否可以结束。当所有运动片段都不再移动或者精度满足要求后则判断分类完成。假设分成 k 类后，形成运动片段集合 $\left(D_1,D_2,\cdots,D_k\right)$，其中聚类精度计算公式为：

$$E_{k-m}=\sum_{\mathrm{i}=1}^{n}\sum_{x\in D_i}\left(x-\mu_i\right)^2 \tag{7.35}$$

$$\mu_i=\frac{\sum_{x\in D_i}x}{\left|D_i\right|} \tag{7.36}$$

式中 μ_i 为第 i 个聚类集合的质心。

7.2.3 基于综合指标的运动片段选取方法

在完成运动片段分类后，需要从每个类中选取足够时间长度的一个或多个运动片段以完成最后的汽车运行工况曲线。从一类运动片段中选取能最好代表这一类运动片段运行特征的片段，可以提高最终所构建汽车运行工况的准确率。因此，区别于传统随机运动片段选取的策略，本文采用基于综合指标的运动片段选取方法。

首先，由于指标矩阵单位不同，需要对指标矩阵内的元素进行归一化处理，处理后的归一化指标矩阵元素计算式为：

$$x_{i,j}=\frac{b_{i,j}-\min(\mathrm{b}_j)}{\max(\mathrm{b}_j)-\min(\mathrm{b}_j)} \tag{7.37}$$

第 i 个运动片段的综合指标 x_i 定义为该片段所有经归一化处理的指标之和，其表达式为：

$$x_i=\sum_{j=1}^{f}x_{i,j} \tag{7.38}$$

按照上述思路，一个聚类集合内所有运动片段的综合指标 Y_k 也可以定义为经归一化处理后的指标之和[4]：

$$Y_k=\sum_{i=1}^{N_k}y_{\mathrm{j},k} \tag{7.39}$$

$$y_{\mathrm{j},k}=\frac{\bar{x}_{\mathrm{j,k}}-\min(x_{j,k})}{\max(x_{j,k})-\min(x_{j,k})} \tag{7.40}$$

$$\bar{x}_{j,k}=\frac{\sum_{j=1}^{f}x_{i,j,k}}{m} \tag{7.41}$$

式中，$y_{j,k}$ 表示第 k 个聚类集合中，所有运动片段的第 j 个指标归一化后的均值，$\bar{x}_{j,k}$ 则表示第 k 个聚类集合中，所有运动片段的第 j 个指标的均值。

将每个运动片段的综合指标 x_i 与该类运动片段的综合指标 Y_k 做差，按照差值绝对值从小到大的顺序对备选片段进行排序，优先选择偏差绝对值小的片段。

计算出各类片段在总数据中（预处理后的）所占比例，结合题目所给的运行工况总时间范围，得到各类中需要提取的运动片段总长度，作为运动片段提取数目的依据。需要从各个聚类集合中提取运动片段的时间长度和 T_k 为：

$$T_k = \frac{\sum_{i=1}^{m} t_{i,k}}{\sum_{k=1}^{f}\sum_{i=1}^{m} t_{i,k}} \times T_{total} \tag{7.42}$$

7.3 汽车运行工况求解

7.3.1 综合指标的求解

根据 7.1 节提取的 11 个汽车运动特征指标和 4 个统计学指标，本文可以用 15 个指标来描述一个运动片段。这样得到的运动特征指标矩阵指标维度为 15，在后续运动片段分类时需要计算和对比的数据比较多，而且观察这 15 个指标，部分指标间有较明显的线性关系，如加速度时间和加速时间比例，这样的具有明显线性关系的指标可以通过线性组合重组成一个新的指标，这个新指标涵盖了原来几个指标的信息，具有概括性而又不改变信息量。本文使用主成分分析的方法进行指标降维。通过计算相关系数矩阵 R 的特征值和特征向量，得到新指标，即第一主成分、第二主成分等的数值大小及各主成分与原指标之间的线性关系。整理数据，得到特征值大小排在前几位的主成分，并计算其贡献率大小以及累计贡献率大小。

计算后知，前 3 位主成分就可以达到原 15 个指标所涵盖信息量的 99%，同时指标维度缩短为原来的 20%。本文取前 3 位主成分，此时可以得到新的运动特征指标矩阵 G，每个运动片段的信息用一个 3 维向量 $g_i(i=1,2,\cdots,N)$ 表示。

7.3.2 运动片段的分类及相应运动特征分析

接着需要对所有运动片段进行分类，本文使用 K-Means 聚类方法，以运动片段指标向量之间的欧几里得距离为标准判断数据集中向量到聚类中心向量的距离，按照距离最小原则进行聚类。由于 K-Means 聚类方法需要提前设定聚类个数，本文结合 NEDC 工况和 WLTC 工况[10]，认为将聚类个数 k 设置在 3～5 的范围比较合适。最终通过计算发现将所有运行片段分为 3 类时精确度最高，于是设定聚类个数 k=3，此时得到分类结果如图 7.3 所示，图中红色表示第一类、蓝色表示第二类、绿色表示第三类，各类中的总运动片段数量以及运动片段累积历经时间如下表 7.4 所示。

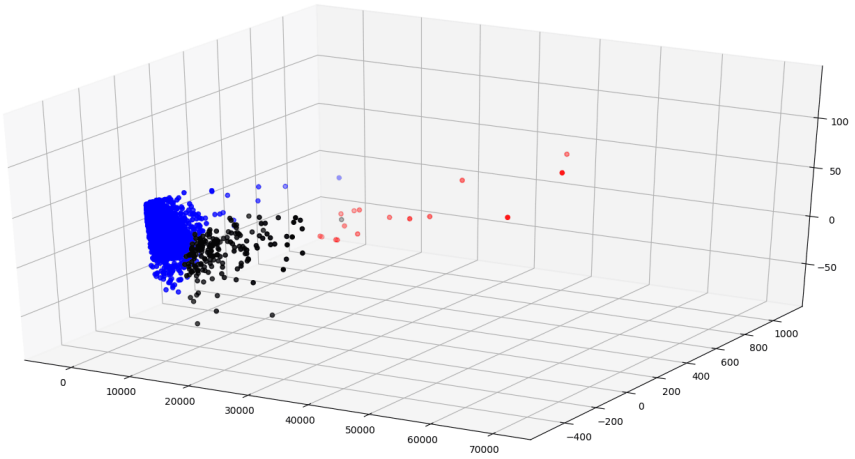


图 7.3 运动片段聚类结果图

表 7.4 聚类结果统计表

	第一类	第二类	第三类
运动片段数量	1904	184	16
T	0.001428922	0.006252045	0.063150263
R_i	0.347441009	0.128265189	0.152760727
R_a	0.468710184	0.631498249	0.741780035
R_d	0.308476351	0.62801484	0.413676618
R_e	0.011664066	0.017622064	0.064238718
v_e	0.018528507	0.020337383	0.066017447
v_{er}	0.013489404	0.017308724	0.065723224
v_{std}	0.023969441	0.028945643	0.078668867
$v_{\max}$	0.009175964	0.013142728	0.066980165
$A_{\mathrm{a,max}}$	0.117616181	0.130677399	0.086904053
$A_{\mathrm{b,max}}$	0.032345357	0.022535382	0.090032642
$A_{a,e}$	0.127929838	0.392031126	0.613501836
$A_{\mathrm{d},e}$	0.08945335	0.164266414	0.644716575
A_{std}	0.182749885	0.314552646	0.644888136
S	0.000639579	0.005468001	0.062506383
综合指标值	1.5736	2.5209	3.8555

分析表 7.4 不同类别运动片段的指标值，可以发现：

第一类运动片段的总时长、平均速度、平均行驶速度、加速时间比例、匀速时间比例、减速时间比例、平均行驶速度、最大速度、平均加速度、平均减速度、加速度标准差和总路程这 12 这指标值都小于第二类运动片段和第三类运动片段的相应指标，而怠速比例这一指标约为另外两类运动片段怠速比例的 2 倍。结合这些指标数据，可以分析出第一类运动片段所描述的汽车运动特征为：汽车运行速度较低、提速后运行较短时间就要减速、提速后行驶较短距离就要减速、较长时间处于等候状态。结合这些运动特征，可以判断第一类运动片段描述的是该汽车在交通情况拥堵路段的行驶工况，运动时段有可能是节假日或

者早晚高峰时期。

第二类运动片段的加速时间比例、减速时间比例这两个指标与第一类和第三类运动片段相比较大；怠速时间比例、匀速时间比例、平均速度、平均行驶速度、速度标注差、最大速度、最大加速度、最大减速度、加速度标准差这 9 个指标值偏小；其他指标值适中。结合上述特点，可以分析出第二类运动片段所描述的汽车运动特征为：汽车加减速运行频繁、运行速度不高、运行较为平稳、速度起伏不大。根据这些运动特征，可以判断第二类运动片段描述的是该汽车在交通较为通畅的城区的行驶工况，该城区网络较为交错，汽车提速和减速频繁可能是行驶过程中遇到较多红灯，也有可能是行人穿行造成的；此外，汽车平均速度不高也对应了城市内汽车行驶限速这一实际情况。

第三类运动片段的总时长、加速时间比例、匀速时间比例、平均速度、平均行驶速度、速度标准差、速度最大值、平均加速度、平均减速度、加速度标准差和总路程这 11 个指标值都大于第一类和第二类运动片段的相应指标值。由此可以看出，第三类运动片段所描述汽车运动特征为：汽车运行顺畅、运行速度偏高、启停车次数少。显然，第三类运动片段描述的是该汽车在城郊或者高速路段的行驶工况，汽车较长时间都处于高速运行状态，提速后可以运行较长距离；此外，匀速行驶时间长可以说明道路通畅，这也与高速或者城郊地区，红绿灯或者收费站点间隔远的特征相匹配。

7.3.3 运动片段的提取与拼接

根据表 7.4 内各个聚类内所有运动片段累积历经时间，利用公式（7.42）可以计算出在汽车运行工况曲线中，需要从每一个聚类中提取的运动片段累积时间长度。本文题目要求汽车运行工况曲线总长度范围为 1200s～1300s，可以得到各类应提取运动片段的时间范围。

在对运动片段进行分类后需要从每一类运动片段中提取出最有代表性的一个或一个运动片段，因此接下来需要进行每一类运动片段综合指标与其中每一个运动片段综合指标的比较。

由于指标单位不一，数量级不等，因此不能使用运动特征指标矩阵 b 内数据直接计算综合特征。在此之前需要对同一指标的数据进行无量纲/归一化处理，处理后的数据只反应原数据在该类数据中的大致程度，而且数据范围集中在 0～1，每个指标都可以得到同等程度的比较。使用式（7.37）进行无量纲处理后，得到无量纲运动指标矩阵，将每个运动片段的指标归一化结果相加就可以得到每个运动片段的综合指标。

对于一类运动片段综合指标的计算，首先进行一指标的无量纲处理。用同一指标的平均值表示该类运动片段该指标的值，参考计算式（7.40），得到 $y_{j,k}$ 为第 k 类运动片段中第 j 个指标平均值的无量纲处理结果。

仿照一个运动片段综合指标的计算，将所有指标的 $y_{j,k}$ 相加，就可以得到这一类运动片段的综合指标值 Y_k。选取标准为单个运动片段综合指标到本类运动片段综合指标的距离：

$$E_{k,i} = \left| Y_k - x_i \right| \tag{7.43}$$

计算出每一类片段中各个片段综合特征参数与各类的综合特征参数差值，列出差值排序表，对备选片段进行排序。对各类片段进行从小到大排序后，得到每类片段排名前九的距离指标，如表 7.5 所示：

表 7.5 各类运动片段距离指标排序表

	第一类	第二类	第三类
No.1	0.0606	2.0429	5.0801
No.2	0.0509	2.0233	5.0373

No.3	0.0435	1.9824	4.8931
No.4	0.0433	1.9722	4.8421
No.5	0.0428	1.9556	4.5794
No.6	0.0423	1.9071	4.3739
No.7	0.0249	1.8409	4.3148
No.8	0.015	1.793	3.9597
No.9	0.0046	1.1836	0.4494
…	…	…	…

结合表 7.5 排序结果和计算出的运动片段时间提取范围，可以从各类运动片段中提取合适数量的运动片段，最终得到汽车运行工况 v-s 图如图 7.4，从各类聚类集合中提取的运动学片段数量和累计运动时间见表 7.6。

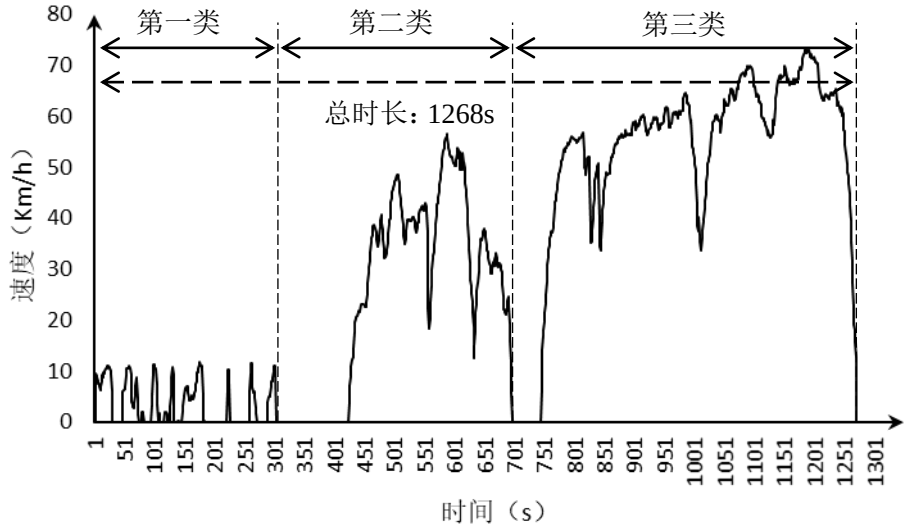


图 7.4 汽车运行工况图

表 7.6 提取运动片段信息表

	第一类	第二类	第三类
提取片段个数	8	1	1
累计时间（s）	301	392	575

图 7.4 所示的汽车运行工况 v-s 曲线基本上与本文 7.3.2 中对三类运动片段的分析和预测内容相匹配。

从第一类运动片段中提取的代表性运动片段曲线十分短促，且曲线高度在 10Km/h 附近起伏，速度跨度小，曲线上升下落迅速，基本上没有速度平台区，可以初步判断第一类运动片段描述的是汽车在交通拥堵的城市地区的运行工况，频发处于提速和加速状态，且汽车运行速度低与堵车时汽车缓慢前行的运动状态相对应。

第二类运动片段所提取出的代表性运动片段曲线连续时间长，曲线与坐标轴所包围面积较大（行驶的路程较远），速度平台区（匀速运动）持续时间较短、汽车速度跌落和上升的幅度较大，速度基本处于较高水平。由此可以推测第二类运动片段描述的是汽车在交通畅通的城市地区的运行工况，较高速运动持续时间长可以反映出道路较为通畅，而加速和减速较为频繁、加速和减速幅度大可以说明该路段存在一些需要急刹车的特殊情况，比

如行人穿行等。

第三类运动片段所提取出的代表性运动片段曲线很长，曲线与时间轴所包围面积大，速度平台区持续时间长，且速度维持在 60Km/h 附近，汽车加减速波动不大。这些特征可以在一定程度上反映汽车在高速公路的行驶工况，高速公路上没有红绿灯和行人且收费站间隔一般较远，所以汽车可以较长连续时间行驶，且高速公路汽车速度偏高。曲线中出行的两次加大幅度速度上升和跌落可能是汽车转弯、换道或者超车等因素造成，基本符合实际高速路段汽车的运动特征。

7.3.4 误差分析

在构建好汽车运行工况后还需进行误差分析，用以验证和说明本文所构建工况曲线的准确性。因此，本文将所得汽车运行工况与经处理后的采集数据的综合指标值进行对比，然后计算误差率。

经过计算得到本文所构建汽车运行工况和经处理后的采集数据的综合指标值以及两者之间的综合指标误差率如表 7.7 所示：

表 7.7 各指标及误差对比表

	构建运行工况	经处理后的采集数据	误差率
R_i	0.386950771	0.328356777	0.1784461
R_a	0.372968308	0.49213361	0.242140
R_d	0.281574592	0.342862796	0.178754
v_e	0.192443012	0.231766486	0.169669
v_{er}	0.185116066	0.227221115	0.273324
v_{std}	0.287994291	0.294008304	0.0204553
$v_{\max}$	0.184152139	0.174302024	0.0565118
$A_{a,e}$	0.056755539	0.045719598	0.2413831
$A_{\mathrm{d},e}$	0.127456306	0.092953974	0.3711765
A_{std}	0.152237395	0.118253449	0.2873822
平均误差		0.2019242	

由表 7.7 可以看出，这 10 个指标大多集中在 2%~30%的误差范围，各个指标平均误差约为 20.19%，其中只有平均减速度误差较大为 37.11765%。由此说明本文构建汽车工况与题目所给数据反映的汽车运行特征基本相符。

为了更好地说明本文所构建汽车运行工况效果，本文再分别采用相似系数和欧式距离这两个指标进行误差分析。

相似系数[11]即待检验向量间的夹角，当夹角为0°时表示相似性好，于是可以用余弦函数表示，其计算式为：

$$\cos\theta = \frac{\langle A, B\rangle}{\|A\| \cdot \|B\|} \tag{7.44}$$

欧氏距离验证[11]即计算两向量之间的欧式距离，其计算式为：

$$\mathrm{d} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \tag{7.45}$$

欧氏距离偏差率计算式为：

$$e=\frac{d}{n} \tag{7.46}$$

式中n表示向量维数。

带入表 7.7 中数据，得到构建运行工况指标向量与经处理后的采集数据指标向量间的相似系数为 0.983，非常接近于 1，由此可以说明本文构建汽车运行工况代表性较强。

再次使用表 7.7 种数据，得到两指标向量间的欧氏距离为 0.1199，欧氏距离偏差率为 1.199%。这也再次说明本文所构建汽车运行工况的准确性，误差分析各个指标数值汇总如表 7.8 所示。

表 7.8 误差分析指标表

平均误差	相似系数	欧氏距离偏差率
20.19%	0.983	1.199%

八、模型的评价

8.1 模型的优点

1. 问题一中我们对短时间内缺失的数据，通过缺失前后的数据信息，采用均值的思想，把数据进行了填补，还原了缺失数据，一定程度上保证了运动过程的完整性。

2. 问题二中运动学片段的提取，除考虑符合运动学片段的基本定义外，新增了片段时长限制和片段数据整体缺失率限制两条新的筛选规则，使得最终提取的运动学片段更科学，更具代表性。

3. 问题三在进行运动片段提取时，本文采用以指标距离偏差为判据的提取原则，既避免了传统随机提取法的随机性，又避免了最佳增量法和 V-A 矩阵法计算的复杂性，起到了平衡计算精度和计算复杂度的作用。

8.2 模型的缺点

1．我们在整个建模过程中，由于时间限制，对于原始数据中各个运动时刻汽车发电机的相关参数的运用比较少，如发电机转速、扭矩百分比和瞬时油耗等参数对于汽车运动状态的关联可以做进一步考虑分析。

2．在使用 K-Means 聚类方法对运动片段进行分类时，由于该算法需要人为设定聚类集合的个数，所以有可能因为人的主观性而使得分类个数设置不合理，从而造成类别缺失或者类与类分界不明显的情况。同时，聚类算法是一种边界性很绝对的算法，当一个数据/向量/集合已经被划分到其中一类时，就一定不会同时存在于其他的聚类集合中。但是，某些处于聚类集合边界的数据可能会对其相邻的几个聚类集合都产生影响。所以，本文使用的聚类方法还有待改进，比如，可以尝试使用神经网络和聚类结合的方法，提高分类的准确性。

3．本文在进行误差分析时，只进行了构建运行工况与经处理后的采集数据之间的偏差程度对比、欧式距离对比和相似系数对比，没有从总体分布情况、相关性强弱等方面进行分析，无法说明所构建汽车行驶工况各指标分布效果的好坏。

九、参考文献

[1] 中国产业信息，2018 年中国汽车行业整体市场需求、细分市场需求及零售量预测，http://www.chyxx.com/industry/201806/649121.html，2019/9/21。
[2] 李连，2016-2025 年国内外汽车行业发展趋势预测，https://wenku.baidu.com/view/d1e8300b85868762caaedd3383c4bb4cf6ecb744.html，2019/9/21。
[3] 翔宇观车，一分钟读懂 NEDC 工况：如何达到工况百公里油耗/电耗水平，https://baijiahao.baidu.com/s?id=1613753349785670755&wfr=spider&for=pc，2019/9/21。
[4] 李洋，基于聚类算法的汽车行驶工况研究[D]，北京，北京理工大学，2016 年。
[5] e 号研究院，还原真实的汽车行驶工况很有必要，切莫再让厂家数据欺骗消费者，http://www.sohu.com/a/247502266_100098859，2019/9/21。
[6] 百度文库，汽车的极限加速度和极限速度，https://wenku.baidu.com/view/ba91f4b881eb6294dd88d0d233d4b14e85243ebe.htm，2019/9/22。
[7] 高建平, 任德轩, 郗建国. 基于全局 K-means 聚类算法的汽车行驶工况构建[J]. 河南理工大学学报(自然科学版), 2019, 38(01):117-123。
[8] 姚延钢，城市道路轻型汽车运行工况构建[D]，2017 年。
[9] 仇多洋，汽车行驶工况的构建及波动特性研究[D]，2012 年。
[10] 王岐东，贺克斌，姚志良, 霍红. 中国城市机动车行驶工况研究[J]. 环境污染与防治, 2007，29(10):745-748,784。
[11] 邵攀登，任田园，李忠玉, 苟琦智等. 基于短行程 V-A 矩阵法的西安市某公交线路工况构建[J].中国汽车工程学会年会论文集, 2017:1362-1365。
[12] 张建伟，李孟良，艾国和，张富兴等. 车辆行驶工况与特征研究[J].汽车工程, 2005,27(2):220-224,245。

十、附录

- **附录一：问题一（数据预处理）的求解程序（Python 语言）**

```
importnumpy as np
import pandas as pd
importdatetime
import copy
time_inc=[]                    #时间不连续节点
speedU=[]                  #加速不合要求 Vup>100/7
speedD=[]                  #减速不合要求 Vdn>28.8
data1 = pd.read_excel('C:/Users/19946/Desktop/redata/文件 3.xlsx')  #原始数据
data1 = pd.DataFrame(data1)
data1 = np.array(data1)
data1_out=copy.deepcopy(data1)          #原始数据副本（记录处理后最终数据）

#寻找断续时间节点
fori in range(data1.shape[0]-1):
    date1 = datetime.datetime.strptime(data1[i][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
```

```
    date2 =
datetime.datetime.strptime(data1[i+1][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
ss=(date2-date1).seconds
ifss!=1:
time_inc+=[[i,ss]]
time_inc=np.array(time_inc)
time_inc=time_inc+np.array([1,0]*time_inc.shape[0]).reshape([time_inc.shape[0],2])
time_inc = pd.DataFrame(time_inc)
time_inc.to_excel('C:/Users/19946/Desktop/time_inc.xlsx')     #输出断续时间节点
#缺失数据补充（仅补充间隔 1 秒）
ff1=[]    #过渡列表，给原始数据补充 1 秒缺失数据
i=0
j=data1_out.shape[0]-1                #跟踪待补充数据表维度
data1_out=data1_out.tolist()        #向量转列表
while(i<j):
    date3 =
datetime.datetime.strptime(data1_out[i][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
    date4 =
datetime.datetime.strptime(data1_out[i+1][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
ss=(date4-date3).seconds
ifss==2:
        ff1=[(date3+datetime.timedelta(seconds=1)).strftime('%Y/%m/%d %H:%M:%S')]
        ff1=ff1+(1/2*(np.array(data1_out[i+1][1:])+np.array(data1_out[i][1:]))).tolist()
        data1_out.insert(i+1,ff1)
        j=j+1
i=i+1
#新数据中寻找加减速异常数据
fori in range(len(data1_out)-1):
    date5 =
datetime.datetime.strptime(data1_out[i][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
    date6 =
datetime.datetime.strptime(data1_out[i+1][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
ss=(date6-date5).seconds
if (data1_out[i+1][1]-data1_out[i][1])/ss>100/7:
speedU.append(i+1)
if (data1_out[i][1]-data1_out[i+1][1])/ss>28.8:
speedD.append(i+1)
data1_out=np.array(data1_out)
#剔除异常数据
data1_out=np.delete(data1_out,speedU,axis=0)
data1_out=np.delete(data1_out,speedD,axis=0)
#数据导出，加减速异常数据节点
speedU=np.array(speedU)
speedD=np.array(speedD)
```

```
speedU=speedU+np.ones([1,speedU.shape[0]])
speedD=speedD+np.ones([1,speedD.shape[0]])
speedU=pd.DataFrame(speedU.T)
speedD=pd.DataFrame(speedD.T)
speedU.to_excel('C:/Users/19946/Desktop/speedU.xlsx')        #输出加速异常节点
speedD.to_excel('C:/Users/19946/Desktop/speedD.xlsx')        #输出减速异常节点

'''
#全 0 片段提取
n=0
count=[]
count1=[]
i=0
while(i<data1.shape[0]-1):
if data1[i][1]==0:
while(data1[i][1]==0):
                        n=n+1
i=i+1
ifi==data1.shape[0]:
break
count.append([i-n,n])
i=i+1
      n=0
fori in range(len(count)-1):
if count[i][1]>=180:
count1.append([count[i][0],count[i][1]])
count=np.array(count)
count1=np.array(count1)
count=pd.DataFrame(count)
count1=pd.DataFrame(count1)
count.to_excel('C:/Users/19946/Desktop/count.xlsx')
count1.to_excel('C:/Users/19946/Desktop/count1.xlsx')
'''

#怠速异常片段提取
#怠速片段提取
n=0
i=0
cnt=[]      #记录怠速点
cnt1=[]     #记录怠速异常点
data1_out=data1_out.tolist()        #向量转列表
while(i<len(data1_out)-1):
if float(data1_out[i][1])<10:
while(float(data1_out[i][1])<10):
```

```
                    n=n+1
i=i+1
ifi==len(data1_out):
break
cnt.append([i-n,n])
i=i+1
    n=0
#怠速异常片段提取
fori in range(len(cnt)-1):
ifcnt[i][1]>180:
cnt1.append([cnt[i][0],cnt[i][1]])
cnt2=copy.deepcopy(cnt1)
cnt1=np.array(cnt1)+np.ones([len(cnt1),1])
cnt1=pd.DataFrame(cnt1)
cnt1.to_excel('C:/Users/19946/Desktop/cnt1.xlsx')      #输出怠速异常片段
#确定怠速异常数据和停车异常数据片段
cut=[]  #异常片段删除区间记录
fori in range(len(cnt2)-1):
if float(data1_out[cnt2[i][0]+179][1])!=0:
cut.append([cnt2[i][0]+180,cnt2[i][0]+cnt2[i][1]-1])
else:
        cut1=[]
        n=0
        j=cnt2[i][0]
while(j<cnt2[i][0]+cnt2[i][1]-1):
if float(data1_out[j][1])==0:
while(float(data1_out[j][1])==0):
                        n=n+1
                        j=j+1
if j==cnt2[i][0]+cnt2[i][1]:
break
cut1.append([j-n,j-1])
            j=j+1
            n=0
ind=[]
for m in range(len(cut1)):
ind.append(cut1[m][1]-cut1[m][0])
if m==len(cut1):
break
index=ind.index(max(ind))
cut.append([cut1[index][0]+5,cut1[index][1]-5])
#删除异常数据片段
data1_out=np.array(data1_out)
fori in range(len(cut)):
```

```
        data1_out=np.delete(data1_out,range(cut[i][0],cut[i][1]+1),axis=0)
#数据输出
cut=np.array(cut)
cut=pd.DataFrame(cut)
cut.to_excel('C:/Users/19946/Desktop/cut.xlsx')       #输出待删除异常数据片段
data1_out=pd.DataFrame(data1_out)
data1_out.to_excel('C:/Users/19946/Desktop/data1_out.xlsx')       #最终数据输出
```

- **附录二：问题二（运动片段划分）的求解程序（Python 语言）**

```
importnumpy as np
import pandas as pd
importdatetime
data1 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP1/data1/data1_out.xlsx')  #原始数据
data1 = pd.DataFrame(data1)
data1 = np.array(data1)
#怠速片段提取
n=0
i=0
rec_idle=[]       #记录怠速片段，包括怠速起始点，结束点，时长
while(i<data1.shape[0]-1):
if data1[i][1]<10:
while(data1[i][1])<10:
                        n=n+1
i=i+1
ifi==data1.shape[0]:
break
rec_idle.append([i-n,i-1,n])
i=i+1
      n=0
#rec_idle=np.array(rec_idle)+np.array([[1,1,0]]*len(rec_idle))
#rec_idle=pd.DataFrame(rec_idle)
#rec_idle.to_excel('C:/Users/19946/Desktop/rec_idle.xlsx')       #输出待候选片段

##挑选运动学片段
#初步挑选候选运动学片段
sport=[]     #储存候选运动学片段
fori in range(len(rec_idle)-1):
      sport.append([rec_idle[i][0],rec_idle[i+1][0],rec_idle[i+1][0]-rec_idle[i][0]+1])
#sport=np.array(sport)+np.array([[1,1,0]]*len(sport))
#sport=pd.DataFrame(sport)
#sport.to_excel('C:/Users/19946/Desktop/sport.xlsx')       #输出怠速异常片段
#根据规则剔除非运动学片段，必须包括怠速，加速，减速和匀速 4 个过程
count=[]        #记录非运动学片段索引
```

```
fori in range(len(sport)):
upsp=0        #记录存在加速阶段
dnsp=0        #记录存在减速阶段
consp=0       #记录存在匀速阶段
      j=sport[i][0]
while(j<=sport[i][1]):
            date1                                                                    =
datetime.datetime.strptime(data1[i][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
            date2                                                                    =
datetime.datetime.strptime(data1[i+1][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
ss=(date2-date1).seconds
if (data1[j+1][1]-data1[j][1])/ss>0.1/3.6 and data1[j+1][1]>=10:
upsp=1
            if abs(data1[j][1]-data1[j+1][1])/ss<0.1/3.6 and min([data1[j][1],data1[j+1][1]])>=10:
consp=1
if (data1[j][1]-data1[j+1][1])/ss>0.1/3.6 and data1[j][1]>=10:
dnsp=1
            j=j+1
if j==data1.shape[0]-1:
break
ifupsp==1 and consp==1 and dnsp==1:
count.append(i)
sport1=[]     #储存运动学片段
fori in range(len(count)):
sport1.append(sport[count[i]])
#sport1=np.array(sport1)+np.array([[1,1,0]]*len(sport1))
#sport1=pd.DataFrame(sport1)
#sport1.to_excel('C:/Users/19946/Desktop/sport1.xlsx')       #输出运动学片段

##筛选有效运动学片段
#剔除片段时间长度少于 20 秒的片段
i=0
cnt_20=[]
while(i<len(sport1)):
if sport1[i][2]>=20:
            cnt_20.append(i)
i=i+1
sport2=[]     #记录剔除时间短的片段
fori in range(len(cnt_20)):
sport2.append(sport1[cnt_20[i]])
#sport2=np.array(sport2)+np.array([[1,1,0]]*len(sport2))
#sport2=pd.DataFrame(sport2)
#sport2.to_excel('C:/Users/19946/Desktop/sport2.xlsx')       #输出运动学片段
#剔除缺失时间不低于 10%的片段
```

```
cnt_10=[]
fori in range(len(sport2)):
misst=0
    date3 =
datetime.datetime.strptime(data1[sport2[i][0]][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
    date4 =
datetime.datetime.strptime(data1[sport2[i][1]][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
len_time=(date4-date3).seconds
    j=sport2[i][0]
while(j<sport2[i][1]):
        date5 =
datetime.datetime.strptime(data1[j][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
        date6 =
datetime.datetime.strptime(data1[j+1][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
ss=(date6-date5).seconds
misst=misst+ss-1
        j=j+1
ifmisst/len_time<0.1:
        cnt_10.append(i)
sport3=[]       #剔除时间缺失率不小于 10%的片段
fori in range(len(cnt_10)):
sport3.append(sport2[cnt_10[i]])
sport3=np.array(sport3)+np.array([[1,1,0]]*len(sport3))
#sport3=pd.DataFrame(sport3)
#sport3.to_excel('C:/Users/19946/Desktop/sport.xlsx')     #输出运动学片段

'''
#寻找处理后新数据断续时间节点
tt=[]           #记录时间断续节点以及时间
timein=[]
fori in range(data1.shape[0]-1):
    date7 = datetime.datetime.strptime(data1[i][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
    date8 =
datetime.datetime.strptime(data1[i+1][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
ss=(date8-date7).seconds
ifss!=1:
timein+=[[i,ss]]
fori in range(len(sport3)):
for j in range(len(timein)):
        if timein[j][0] in range(sport3[i][0],sport3[i][1]+1):
tt.append([sport3[i][0],sport3[i][1],timein[j][0],timein[j][1]])
'''
```

- **附录三：问题三（构建汽车运行工况）的求解程序（Python 语言）**

——求特征参数值——

```
importnumpy as np
import pandas as pd
importdatetime
data1 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP1/data3/data1_out.xlsx')  #汽车运行数据
data1 = pd.DataFrame(data1)
data1 = np.array(data1)
data2 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP2/sport3.xlsx')  #运动学片段
data2 = pd.DataFrame(data2)
data2 = np.array(data2)
data2=data2-np.array([[1,1,0]]*data2.shape[0])
feature=[]                    #记录每个运动学片段特征
#特征参数
T=0                 #运行时间(s)
S=0                 #运行距离(m)
Ti=0                #怠速时间(s)
Ta=0                #加速时间(s)
Td=0                #减速时间(s)
Te=0                #匀速时间(s)
Vmax=0              #最大速度(Km/h)
Vm=0                #平均速度(Km/h)
Vmr=0               #行驶平均速度(Km/h)
Vs=0                #速度标准偏差(Km/h)
Amax=0              #最大加速度(m/s^2)
Am=0                #平均加速度(m/s^2)
Dmax=0              #最大减速度(绝对值,m/s^2)
Dm=0                #平均减速度(绝对值,m/s^2)
As=0                #加速度标准偏差(m/s^2)
fori in range(data2.shape[0]):

time1=datetime.datetime.strptime(data1[data2[i][0]][0].replace('.000.',''),'%Y/%m/%d  %H:%M:%S')

time2=datetime.datetime.strptime(data1[data2[i][1]][0].replace('.000.',''),'%Y/%m/%d  %H:%M:%S')
    T=(time2-time1).seconds
    S=0
Ti=0
    Ta=0
    Td=0
Te=0
    Vmax=0
Vm=0
Vmr=0
```

```
Vss=[]
count_Vmr=0
    Amax=0
    Am=0
count_Am=0
Dmax=0
Dm=0
count_Dm=0
    Ass=[]
for j in range(data2[i][0],data2[i][1]):
        S=S+1/2*(data1[j][1]+data1[j+1][1])

time3=datetime.datetime.strptime(data1[j][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')

time4=datetime.datetime.strptime(data1[j+1][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
ss=(time4-time3).seconds
if data1[j][1]<10:
Ti=Ti+ss
if (data1[j+1][1]-data1[j][1])/ss>0.1/3.6 and data1[j+1][1]>=10:
                Ta=Ta+ss
if Amax<(data1[j+1][1]-data1[j][1])/ss:
                    Amax=(data1[j+1][1]-data1[j][1])/ss
                Am=Am+(data1[j+1][1]-data1[j][1])/ss
count_Am=count_Am+1
Ass.append((data1[j+1][1]-data1[j][1])/ss)
if (data1[j][1]-data1[j+1][1])/ss>0.1/3.6 and data1[j][1]>=10:
                Td=Td+ss
ifDmax<(data1[j][1]-data1[j+1][1])/ss:
Dmax=(data1[j][1]-data1[j+1][1])/ss
Dm=Dm+(data1[j][1]-data1[j+1][1])/ss
count_Dm=count_Dm+1
        if abs((data1[j][1]-data1[j+1][1])/ss)<0.1/3.6 and min([data1[j][1],data1[j+1][1]])>=10:
Te=Te+ss
if Vmax<data1[j][1]:
                Vmax=data1[j][1]
Vm=Vm+data1[j][1]
if data1[j][1]>=10:
Vmr=Vmr+data1[j][1]
count_Vmr=count_Vmr+1
Vss.append(data1[j][1])
Vm=(Vm+data1[data2[i][1]][1])/data2[i][2]
Vmr=Vmr/count_Vmr
    Am=Am/count_Am
Dm=Dm/count_Dm
```

```
Vss.append(data1[data2[i][1]][1])
        Vs=np.std(np.array(Vss))
        As=np.std(np.array(Ass))
        feat=[T,S,Ti/T,Ta/T,Td/T,Te,Vmax,Vm,Vmr,Vs,Amax,Am,Dmax,Dm,As]
feature.append(feat)
feature=np.array(feature)
feature=pd.DataFrame(feature)
feature.to_excel('C:/Users/19946/Desktop/feature.xlsx')       #输出特征参数数据
```

——选运动学片段组成工况曲线————

```
fromsklearn.decomposition import PCA
fromsklearn.cluster import KMeans
importmatplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
importnumpy as np
import pandas as pd
import copy
importdatetime
data1 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP3/feature1.xlsx')  #文件 1 特征参数数据
data1 = pd.DataFrame(data1)
data1 = np.array(data1)
feature2 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP3/feature2.xlsx')  #文件 2 特征参数数据
feature2 = pd.DataFrame(feature2)
feature2 = np.array(feature2)
feature3 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP3/feature3.xlsx')  #文件 3 特征参数数据
feature3 = pd.DataFrame(feature3)
feature3 = np.array(feature3)
data0 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP1/data1/data1_out.xlsx')  #文件 1 汽车运行数据
data0 = pd.DataFrame(data0)
data0 = np.array(data0)
data02 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP1/data2/data1_out.xlsx')  #文件 2 汽车运行数据
data02 = pd.DataFrame(data02)
data02 = np.array(data02)
data03 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP1/data3/data1_out.xlsx')  #文件 3 汽车运行数据
data03 = pd.DataFrame(data03)
data03 = np.array(data03)
sport = pd.read_excel('C:/Users/19946/Desktop/test4_resultP2/sport1.xlsx')  #文件 1 运动学片段
```

```
sport = pd.DataFrame(sport)
sport = np.array(sport)
sport=sport-np.array([[1,1,0]]*sport.shape[0])
sport2 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP2/sport2.xlsx')  #文件2运动学片段
sport2 = pd.DataFrame(sport2)
sport2 = np.array(sport2)
sport2=sport2-np.array([[1,1,0]]*sport2.shape[0])
sport3 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP2/sport3.xlsx')  #文件3运动学片段
sport3 = pd.DataFrame(sport3)
sport3 = np.array(sport3)
sport3=sport3-np.array([[1,1,0]]*sport3.shape[0])
#合并 3 个文件的运动学片段，索引 0:875 为文件 1，876：1543 为文件 2,1544：2103
data=data1.tolist()+feature2.tolist()+feature3.tolist()
data=np.array(data)
#PCA 3 个主成分
pca_d = PCA(n_components=3)      #选择 3 个主成分
data_new=pca_d.fit_transform(data)
clust=KMeans(n_clusters=3)                          #构造聚类,分 3 类
clust.fit(data_new)                                 #聚类
label_p=clust.labels_                               #获取聚类标签
#聚类画图
la1 = data_new[label_p == 0]
la2 = data_new[label_p == 1]
la3 = data_new[label_p == 2]
x1=[]
y1=[]
z1=[]
x2=[]
y2=[]
z2=[]
x3=[]
y3=[]
z3=[]
fori in range(la1.shape[0]):
x1.append(la1[i][0])
y1.append(la1[i][1])
z1.append(la1[i][2])
fori in range(la2.shape[0]):
x2.append(la2[i][0])
y2.append(la2[i][1])
z2.append(la2[i][2])
fori in range(la3.shape[0]):
```

```
x3.append(la3[i][0])
y3.append(la3[i][1])
z3.append(la3[i][2])
fig=plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x1,y1,z1,c='b')
ax.scatter(x2,y2,z2,c='r')
ax.scatter(x3,y3,z3,c='k')
plt.show()

'''
plt.scatter(x0[:, 0], x0[:, 1], c = "red", marker='o', label='label0')
plt.scatter(x1[:, 0], x1[:, 1], c = "green", marker='*', label='label1')
plt.scatter(x2[:, 0], x2[:, 1], c = "blue", marker='+', label='label2')
plt.xlabel('petal length')
plt.ylabel('petal width')
plt.legend(loc=2)
plt.show()
'''
'''
data_c1=[]              #类别 1 的片段
data_c2=[]
data_c3=[]
index1=[]               #记录属于类别 1 的片段索引
index2=[]
index3=[]
data2=copy.deepcopy(data)
data2=data2.tolist()
fori in range(len(data2)):
data2[i].insert(0,i)
#运动学片段分类
fori in range(len(data2)):
iflabel_p[i]==0:
            data_c1.append(data2[i])
index1.append(i)
iflabel_p[i]==1:
            data_c2.append(data2[i])
index2.append(i)
iflabel_p[i]==2:
            data_c3.append(data2[i])
index3.append(i)
fori in range(len(data_c1)):
      dc1=np.array(data_c1).T
for j in range(1,dc1.shape[0]):
```

```
            max_c1=max(dc1[j])
            min_c1=min(dc1[j])
            data_c1[i][j]=(data_c1[i][j]-min_c1)/(max_c1-min_c1)
fori in range(len(data_c2)):
    dc2=np.array(data_c2).T
for j in range(1,dc2.shape[0]):
            max_c2=max(dc2[j])
            min_c2=min(dc2[j])
            data_c2[i][j]=(data_c2[i][j]-min_c2)/(max_c2-min_c2)
fori in range(len(data_c3)):
    dc3=np.array(data_c3).T
for j in range(1,dc3.shape[0]):
            max_c3=max(dc3[j])
            min_c3=min(dc3[j])
            data_c3[i][j]=(data_c3[i][j]-min_c3)/(max_c3-min_c3)
data_rec1=[]
data_rec2=[]
data_rec3=[]
fori in range(len(data_c1)):
    data_rec1.append([data_c1[i][0],sum(data_c1[i][1:])])
fori in range(len(data_c2)):
    data_rec2.append([data_c2[i][0],sum(data_c2[i][1:])])
fori in range(len(data_c3)):
    data_rec3.append([data_c3[i][0],sum(data_c3[i][1:])])
#类无量纲处理
mean_c1=[]
mean_c2=[]
mean_c3=[]
fori in range(1,dc1.shape[0]):
    meanc1=np.mean(dc1[i])
    maxc1=max(dc1[i])
    minc1=min(dc1[i])
    mean_c1.append((meanc1-minc1)/(maxc1-minc1))
total_clu1=sum(mean_c1)
fori in range(1,dc2.shape[0]):
    meanc2=np.mean(dc2[i])
    maxc2=max(dc2[i])
    minc2=min(dc2[i])
    mean_c2.append((meanc2-minc2)/(maxc2-minc2))
total_clu2=sum(mean_c2)
fori in range(1,dc3.shape[0]):
    meanc3=np.mean(dc3[i])
    maxc3=max(dc3[i])
    minc3=min(dc3[i])
```

```
        mean_c3.append((meanc3-minc3)/(maxc3-minc3))
total_clu3=sum(mean_c3)
fori in range(len(data_rec1)):
        data_rec1[i][1]=abs(data_rec1[i][1]-total_clu1)
fori in range(len(data_rec2)):
        data_rec2[i][1]=abs(data_rec2[i][1]-total_clu2)
fori in range(len(data_rec3)):
        data_rec3[i][1]=abs(data_rec3[i][1]-total_clu3)
zc1={}
zc2={}
zc3={}
fori in range(len(data_rec1)):
zc1[data_rec1[i][0]]=data_rec1[i][1]
zc1=sorted(zc1.items(),key=lambda item:item[1])         #升序排序
fori in range(len(data_rec2)):
zc2[data_rec2[i][0]]=data_rec2[i][1]
zc2=sorted(zc2.items(),key=lambda item:item[1])
fori in range(len(data_rec3)):
zc3[data_rec3[i][0]]=data_rec3[i][1]
zc3=sorted(zc3.items(),key=lambda item:item[1])
#每类时间占比
sumT1=0
sumT2=0
sumT3=0
fori in range(sport.shape[0]):
    date1                                                                                    =
datetime.datetime.strptime(data0[sport[i][0]][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
    date2                                                                                    =
datetime.datetime.strptime(data0[sport[i][1]][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
ss=(date2-date1).seconds
ifi in index1:
            sumT1=sumT1+ss
ifi in index2:
            sumT2=sumT2+ss
ifi in index3:
            sumT3=sumT3+ss
fori in range(sport2.shape[0]):
    date3                                                                                    =
datetime.datetime.strptime(data02[sport2[i][0]][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
    date4                                                                                    =
datetime.datetime.strptime(data02[sport2[i][1]][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
ss=(date4-date3).seconds
if i+876 in index1:
            sumT1=sumT1+ss
```

```
if i+876 in index2:
        sumT2=sumT2+ss
if i+876 in index3:
        sumT3=sumT3+ss
fori in range(sport3.shape[0]):
    date5                                                                    =
datetime.datetime.strptime(data03[sport3[i][0]][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
    date6                                                                    =
datetime.datetime.strptime(data03[sport3[i][1]][0].replace('.000.',''),'%Y/%m/%d %H:%M:%S')
ss=(date6-date5).seconds
if i+1544 in index1:
        sumT1=sumT1+ss
if i+1544 in index2:
        sumT2=sumT2+ss
if i+1544 in index3:
        sumT3=sumT3+ss
sum_c1=sumT1/(sumT1+sumT2+sumT3)*1200
sum_c2=sumT2/(sumT1+sumT2+sumT3)*1200
sum_c3=sumT3/(sumT1+sumT2+sumT3)*1200
'''

'''
#挑选片段
select1=[]
select2=[]
select3=[]
sum1=0
fori in range(len(zc1)):
if sum1<=sum_c1:
        date3                                                                =
datetime.datetime.strptime(data0[sport[zc1[i][0]][0]][0].replace('.000.',''),'%Y/%m/%d  %H:%M:
%S')
        date4                                                                =
datetime.datetime.strptime(data0[sport[zc1[i][0]][1]][0].replace('.000.',''),'%Y/%m/%d  %H:%M:
%S')
        sum1=sum1+(date4-date3).seconds
        select1.append([sport[zc1[i][0]][0],sport[zc1[i][0]][1]])
if sum1>sum_c1:
break
sum2=0
fori in range(len(zc2)):
if sum2<=sum_c2:
        date3                                                                =
datetime.datetime.strptime(data0[sport[zc2[i][0]][0]][0].replace('.000.',''),'%Y/%m/%d  %H:%M:
```

```
%S')
        date4                                                                    =
datetime.datetime.strptime(data0[sport[zc2[i][0]][1]][0].replace('.000.',''),'%Y/%m/%d  %H:%M:
%S')
        sum2=sum2+(date4-date3).seconds
        select2.append([sport[zc2[i][0]][0],sport[zc2[i][0]][1]])
if sum2>sum_c2:
break
sum3=0
fori in range(len(zc3)):
if sum3<=sum_c3:
        date3                                                                    =
datetime.datetime.strptime(data0[sport[zc3[i][0]][0]][0].replace('.000.',''),'%Y/%m/%d  %H:%M:
%S')
        date4                                                                    =
datetime.datetime.strptime(data0[sport[zc3[i][0]][1]][0].replace('.000.',''),'%Y/%m/%d  %H:%M:
%S')
        sum3=sum3+(date4-date3).seconds
        select3.append([sport[zc3[i][0]][0],sport[zc3[i][0]][1]])
if sum3>sum_c3:
break
'''
```

——计算总体误差———

```
importnumpy as np
import pandas as pd
data1 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP3/feature1.xlsx')  #文件 1 特征参
数数据
data1 = pd.DataFrame(data1)
data1 = np.array(data1)
feature2 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP3/feature2.xlsx')  #文件 2 特征
参数数据
feature2 = pd.DataFrame(feature2)
feature2 = np.array(feature2)
feature3 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP3/feature3.xlsx')  #文件 3 特征
参数数据
feature3 = pd.DataFrame(feature3)
feature3 = np.array(feature3)
data0 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP1/data1/data1_out.xlsx')  #文件 1
汽车运行数据
data0 = pd.DataFrame(data0)
data0 = np.array(data0)
data02 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP1/data2/data1_out.xlsx')  #文件 2
汽车运行数据
```

```
data02 = pd.DataFrame(data02)
data02 = np.array(data02)
data03 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP1/data3/data1_out.xlsx')  #文件 3 汽车运行数据
data03 = pd.DataFrame(data03)
data03 = np.array(data03)
sport = pd.read_excel('C:/Users/19946/Desktop/test4_resultP2/sport1.xlsx')  #文件 1 运动学片段
sport = pd.DataFrame(sport)
sport = np.array(sport)
sport=sport-np.array([[1,1,0]]*sport.shape[0])
sport2 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP2/sport2.xlsx')  #文件 2 运动学片段
sport2 = pd.DataFrame(sport2)
sport2 = np.array(sport2)
sport2=sport2-np.array([[1,1,0]]*sport2.shape[0])
sport3 = pd.read_excel('C:/Users/19946/Desktop/test4_resultP2/sport3.xlsx')  #文件 3 运动学片段
sport3 = pd.DataFrame(sport3)
sport3 = np.array(sport3)
sport3=sport3-np.array([[1,1,0]]*sport3.shape[0])
#合并 3 个文件的运动学片段，索引 0:875 为文件 1，876：1543 为文件 2,1544：2103
data=data1.tolist()+feature2.tolist()+feature3.tolist()
data=np.array(data)
#无量纲处理获得总体特征
data_T=data.T
mean_total=[]
fori in range(data_T.shape[0]):
meancto=np.mean(data_T[i])
maxcto=max(data_T[i])
mincto=min(data_T[i])
mean_total.append((meancto-mincto)/(maxcto-mincto))
total_clu=sum(mean_total)
zc1=[1470,29,219,565,354,247,164,283]          #第 1 类挑选的片段
zc2=1                                          #第 2 类挑选的片段
zc3=79                                         #第 3 类挑选的片段
select=[]                                      #合并片段
select=select+[data1[zc2].tolist()]+[data1[zc3].tolist()]
fori in range(len(zc1)):
if zc1[i]<876:
select=select+[data1[zc1[i]].tolist()]
else:
select=select+[feature2[zc1[i]-876].tolist()]
select=np.array(select).T
```

```
mean_select=[]
fori in range(select.shape[0]):
mean_sel=np.mean(select[i])
max_sel=max(select[i])
min_sel=min(select[i])
mean_select.append((mean_sel-min_sel)/(max_sel-min_sel))
total_sel=sum(mean_select)
error_total=abs(total_sel-total_clu)/total_clu
print('Total error is %f' % error_total)
```