



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	山东大学（威海）
--------	----------

参赛队号	19194220011
------	-------------

队员姓名	1.	钟倩文
	2.	李壮壮
	3.	毕研庆

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 汽车行驶工况的构建与研究

摘 要：

汽车行驶工况是描述汽车行驶的速度-时间曲线，体现汽车道路行驶的运动学特征，是汽车行业的一项重要的、共性基础技术，也是汽车各项性能指标表达优化时的主要标准。本文研究了通过现实中汽车道路行驶所采集的数据，来对汽车行驶工况曲线进行构建的方法。利用 SPSS 软件对原始数据进行分析，使用 Python 语言对数据处理、模型求解以及可视化作图。

对于问题一：设计了三个数据处理步骤：数据插值与预分割、数据扩充、异常处理和数据修正。首先，将时间片段缺失长度大于 180 秒的节点进行**预分割**，在分割出来的 212 段数据片段内部对 GPS 车速进行指数插值，对其它信息进行线性插值，一共插入 45624 行数据。然后补充了加速度、位移、新的瞬时油耗、停车判别 4 条新信息。通过加速度信息，剔除了汽车加、减速度异常的数据 476 行；通过停车判别和 GPS 车速，剔除了长期停车所采集的异常数据 187 段；通过持续的低速判别，将连续的低速片段中 GPS 车速置为 0m/s，一共处理了 148 段；通过遍历数据，寻找 180 秒以上的片段，删去了怠速超过 180 秒的数据 300 段。通过三步处理，对原始数据进行全面修正。

针对问题二：依据统计分析，使用连续的 5 个怠速运行时间点作为运动学片段的开始和结尾标志。遍历数据，判断出怠速区间，然后对两个怠速区间内的片段进行分离，得到独立的运动学片段并进行保存。剔除无效的片段后一共得到 2950 条运动学片段。综合考虑后，又删除了 550 条长度小于 20 秒和长度大于 1300 秒的运动学片段，最终得到 2400 条比较好的运动学片段。

针对问题三：论述了离散小波压缩模型可能并不适合本题。因此，提出了**基于主成分分析与 K-means 聚类分析的运动学片段分类模型**，并通过遗传算法对 4 类聚类的运动学片段样本进行抽取，构建汽车行驶工况曲线。首先对原始数据中 9 个特征进行扩充，得到一共 14 个特征。通过主成分分析，选取 7 个累积贡献率带到 95.154% 的特征进行聚类。经过论述，产生了 4 个聚类。接着使用遗传算法模型，对目标函数：归一化加权累积均方误差进行迭代优化。提取出了具有代表性的运动学片段构建汽车行驶工况曲线。最终得到的归一化加权累积均方误差值为 0.073。对于汽车运动特征评估体系，使用题目中提到的 9 种特征误差以及误差均值作为评估标准。将构建的工况曲线特征值与原始数据中特征的统计平均值进行比较，得到以下数据：平均速度（1.52%）、平均行驶速度（1.48%）、平均加速度（15.24%）、平均减速度（12.91%）、怠速时间比（0.43%）、加速时间比（0%）、减速时间比（5.29%）、速度标准差（0.71%）、加速度标准差等（3.5%）。计算**误差均值**

为 4.53%。

最后对模型进行了综合评价，论证了模型构建的汽车行驶工况曲线具有代表性。能较好的还原原始数据的特征。同时也在最后提出了模型中 6 点还能进一步改进的地方。本文对真实数据的严谨处理、对模型建立的严格论证，最后得出了较好结果，能为我国各地的实际道路交通状况的汽车行驶工况的建立提供一定的参考。

关键词：汽车行驶工况 插值法 主成分分析 K-means 聚类 遗传算法

目录

一、问题重述	5
1.1 问题背景.....	5
1.2 需要解决的问题.....	5
二、模型假设说明.....	8
三、符号说明	9
四、问题一：数据预处理.....	10
4.1 问题分析.....	10
4.2 数据预处理方法以及实现.....	10
4.2.1 数据插值与预分割.....	10
4.2.2 数据扩充.....	11
4.2.3 异常数据处理和数据修正.....	13
4.3 数据预处理结果分析.....	14
4.3.1 数据预处理小结.....	14
4.3.2 数据预处理评价.....	15
五、问题二：运动学片段的提取.....	16
5.1 问题分析.....	16
5.2 运动学片段提取与处理.....	16
5.2.1 运动学片段分割.....	16
5.2.2 分割片段处理.....	17
5.3 运动学片段提取结果分析.....	17
5.3.1 运动学片段提取小结.....	17
5.3.2 运动学片段提取评价.....	18
六、问题三：汽车行驶工况的构建.....	18
6.1 问题分析.....	18
6.2 工况曲线的构建.....	19
6.2.1 特征建立与扩展.....	19
6.2.2 小波变换模型.....	21
6.2.3 主成分分析模型.....	23
6.2.4 K-means 聚类模型	26
6.2.5 长片段抽样.....	29
6.2.6 基于遗传算法的工况曲线合成.....	30
6.3 汽车运动特征的评估体系.....	33
6.3.1 归一化加权累积均值误差.....	34
6.3.2 特征误差与误差均值.....	34
6.4 汽车行驶工况的构建结果分析.....	35
6.4.1 工况曲线的建立小结.....	35
6.4.2 评价模型小结.....	36
6.4.3 评价模型与工况曲线评估.....	36
七、总结与展望	37
7.1 实验结论.....	37
7.2 模型评价.....	37

7.3 模型改进.....	38
参考文献	39
附录	40
最终构建工况曲线数据（截取部分）	40
主要实现代码（输出各个特征误差）	41
主要实现代码（数据拼接）	42
主要实现代码（数据处理）	43

一、问题重述

1.1 问题背景

汽车行驶工况(Driving Cycle)^[1]又称车辆测试循环,是描述典型汽车行驶的速度-时间曲线。主要应用于车辆发动机的燃油经济性 & 污染物排放性能的评价。它是由一系列的“怠速-运行-怠速”的运动学片段组成的。对特定交通条件下的真实工况提取^[2~4],可以对车辆的运行状态和道路行驶的运动学特征进行综合模拟。可以说汽车行驶工况是汽车行业的一项重要、共性基础技术,是车辆能耗/排放测试方法和限值标准的基础,也是汽车各项性能指标标定优化时的主要基准。

我国早期使用的是欧洲的 NEDC 行驶工况标准。由于近年来我国汽车保有量快速增长,道路交通状况发生了很大的改变。政府、企业和民众日渐发现以 NEDC 工况为基准所优化标定的汽车,实际油耗与法规认证结果偏差越来越大。至于国际上的其它工况标准如 WLTC,在怠速时间比和平均速度这两个主要工况特征上,与我国实际汽车行驶工况法差异更大。再加上我国幅员辽阔,各个城市发展程度、气候条件以及交通状况的不同,决定了一套旧标准不能很好的适应于我国各地的实际道路交通状况。因此,希望能针对我国各个城市自身的汽车行驶数据,来构建与该城市最吻合的汽车行驶工况标准。

在此背景下,本题提供了 3 个数据文件。分别为同一辆轻型汽车,在同一座城市内的不同时间段的实际道路行驶采集的数据。依据该数据,最终构建一条汽车行驶工况曲线(1200-1300 秒)。同时要求建立汽车运动特征模型,模型能代表所采集数据源的对应特征,并且与原数据对应特征间的误差越小越好。

1.2 需要解决的问题

实际道路运行产生的汽车行驶数据不同于实验室模拟的产生一些“怠速-运行-怠速”、“加速”、“减速”的运动学片段。实际数据中由于路况的复杂性和一些突发的偶然事件,会导致数据相对复杂。分析数据的过程中不可避免的需要对数据进行预处理。根据处理完的数据以及怠速时间提取出大量的运动学片段。针对这些片段再选取出具有代表性运动学片段进行汽车行驶工况曲线的合成,最后建立模型对曲线进行评估和分析。

对于附件中的三个文件,分别记录了 2017 年 11 月份和 12 月份的三个时间段的道路行驶数据。其中采集的数据分别为:时间、GPS 车速、X 轴加速度、Y 轴加速度、Z 轴加速度、经度、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。这些数据中有经纬度数据丢失、瞬时油耗与实际情况不符、时间信息不连续且空缺部分大小的分布非常离散等问题。

因此在处理题目给出的问题之前,还需要解决以下问题:

1) 附件的文件 1 有三个时间点的纬度数据丢失。这三个点都是处在时间不连续处,因此先直接删除,之后依据不连续时间的长短决定是否进行插值。文件 2 中有 300 个时间点的经纬度数据丢失。这 300 个时间点的数据也是处在时间不连续处,且与前后的有效数据点的时间均相差了 2 个小时左右。因此,这 300 个数据也全部删除。

2) 附件数据中许多速度为 0km/s 或者低速行驶的状态下,瞬时油耗异常大。汽车的瞬时油耗计算公式为:单位时间(1s)的喷油量/单位时间(1s)的里程。因此,汽车在低速行驶或者发动机空转时的瞬时油耗理论计算将非常大,但这却并不符合实际情况。对瞬时

油耗异常的部分，使用空燃比和进气量进行修正。空燃比就是空气与燃料的比值，根据采集的进气量数据，可得出：瞬时油耗=空燃比*进气量。

3) 附件数据中许多数据都是时间不连续的。空缺的时间片段从 8 个多小时到几秒钟不等。在此，我们依据时间片段缺失的大小进行分别处理。**缺失小于或等于 180s 的，对 GPS 速度进行指数插值^[5]，对其它参数进行线性插值。缺失大于 180s，对数据进行预分割。**此处的预分割即有利于新特征（如加速度、移动距离等的）的建立，也有利于后续依据怠速时间对运动学片段的提取。同时也能通过预分割出来的数据长短，对运动学片段进行初步筛选。

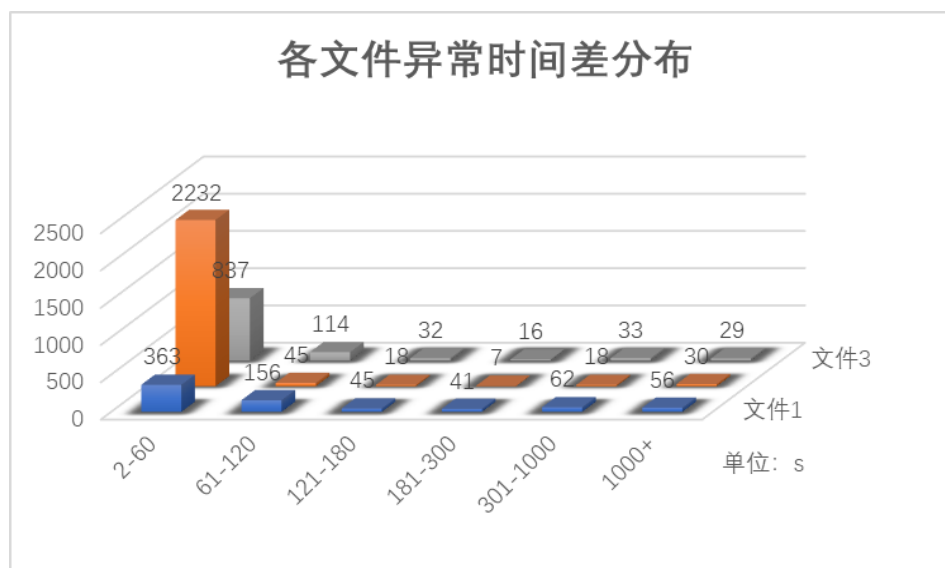


图 1 文件 1-文件 3 时间异常片段统计

问题一：数据预处理

分析附件中的三个数据文件就可以知道，汽车行驶数据的采集设备直接记录的原始采集数包含许多不良数据值。在解决问题一之前，就已经对某些非常明显错误数据进行删除、替换。最重要的是对数据按照空缺的时间片段进行了预分割。

问题一的数据预处理，事实上是为第二问、第三问服务的。题中提出了不良数据主要包括以下几种类型：

- (1) 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；
- (2) 汽车加、减速度异常的数据（普通轿车一般情况下：0 至 100km/h 的加速时间大与 7 秒，紧急刹车最大减速度在 $7.5 \sim 8 \text{ m/s}^2$ ）；
- (3) 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。
- (4) 长时间堵车、断断续续低速行驶情况（最高车速小于 10km/h），通常可按怠速情况处理。
- (5) 一般认为怠速时间超过 180 秒为异常情况，怠速最长时间可按 180 秒处理。

因此，问题一需要针对上述情况，对数据进行相应的插值、修正、补充与剔除。

问题二：运动学片段的提取

运动学片段是指汽车从怠速状态开始到下一个怠速状态开始之间的车速区间，如图 2 所示。依据问题一预处理好的数据，划分多个运动学片段，最后给出各数据文件最终得到的运动学片段。同样，问题二中提取的运动学片段也是用于构建问题三中汽车行驶工况曲线。曲线构建准确与否与运动学片段的提取有很大关系。

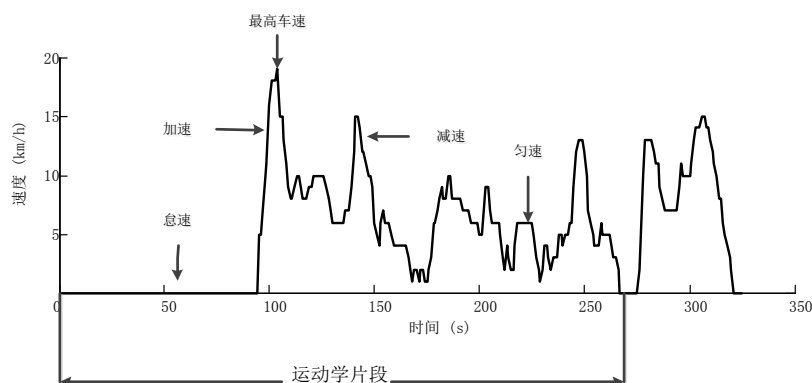


图 2 运动学片段的定义

问题三：汽车行驶工况的构建

上述两问中对数据进行了预处理同时提取出了多个运动学片段。需要依据这些运动学片段构建出一条最符合所采集数据源（经处理后的数据）相应特征的汽车行驶工况曲线（1200-1300 秒）。同时需要建立该曲线的评估模型，比对工况曲线与数据源之间的相应特征。两者间的误差越小，说明所构建的汽车行驶工况的代表性越好。对于汽车行驶工况的构建，题中给出了以下几点要求：

（1）科学、有效的构建方法（数学模型或算法，特别鼓励创新方法，如果采用已有的方法，必须注明来源）；

（2）合理的汽车运动特征评估体系（至少包含但不限于以下指标：平均速度（km/h）、平均行驶速度（km/h）、平均加速度（ m/s^2 ）、平均减速度（ m/s^2 ）、怠速时间比（%）、加速时间比（%）、减速时间比（%）、速度标准差（km/h）、加速度标准差等（ m/s^2 ）；

（3）按照你们所构建的汽车行驶工况及汽车运动特征评估体系，分别计算出汽车行驶工况与该城市所采集数据源（经处理后的数据）的各指标（运动特征）值，并说明你们所构建的汽车行驶工况的合理性。

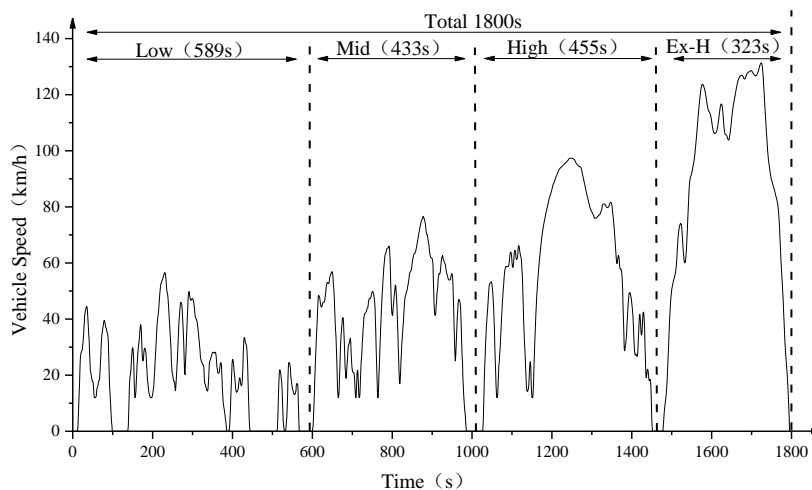


图 3 汽车行驶工况曲线

二、模型假设说明

编号	假设
1	0 至 100km/h 的加速时间大于 7 秒，即最大加速度为 3.97m/s^2
2	紧急刹车最大减速度在 $7.5\sim 8\text{m/s}^2$ 之间
3	长期低速（小于 10km/s）行驶可按怠速处理
4	一般认为怠速时间超过 180 秒为异常情况，怠速最长时间可按 180 秒处理
5	停车的依据可以是 GPS 车速和经纬度坐标
6	GPS 对经纬度坐标的平均定位误差 10m 以内
7	空燃比与进气流量相对准确

数据文件预分割说明：

将数据按照时间片段缺失的大小进行了预分割。对少量（180s 内）的缺失数据进行了插值。而大于该缺失时间片段的部分，对缺失部分的前后数据进行预分割。可以看到，分割后的长度大部分都在 500 秒以上。已知附件中的三个文件也都是同一辆车在不同时间段的真实行驶数据，可以认为此处的分割与原始数据不同时间段采集的三个数据文件并无区别。综合考虑了 GPS 卫星的轨道速度以需要三颗卫星才能精确定位的特点、高层建筑覆盖或过隧道时间长度、将怠速时间超过 180s 视为异常情况等特点。将分割依据，也就是时间缺失片段大小设定为 180s。

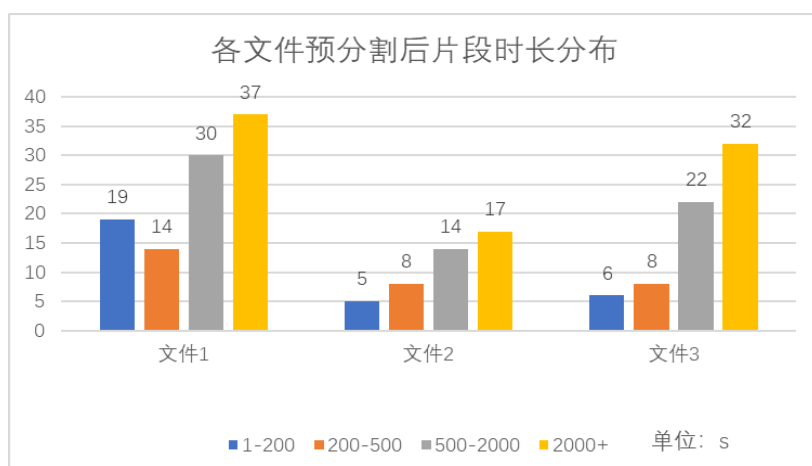


图 4 各文件分割统计

以 180s 时间片段缺失为依据，分别将文件 1 分割为 100 段，文件 2 分割为 44 段，文件 3 分割为 68 段。

三、符号说明

符号	含义
T_{lost}	空缺的时间片段长度
T_{stop}	停车时间
T_{idle}	怠速时间
a_{pri}	原始数据中加速度（减速度为负值）
v_{pri}	原始中汽车速度
v_{last}	间断点前速度
v_{next}	间断点后速度
T_{lsp}	低速行驶时间
T_{seq}	运动学片段长度
v_{seq}	运动学片段平均速度
a_{useq}	运动学片段平均加速度
a_{dseq}	运动学片段平均减速度
v_i	运动学片段速度
a_i	运动学片段加速度
v_{seqz}	平均行驶速度
P_{idle}	怠速时间比
P_{au}	加速度时间比
v_{csd}	速度标准差
P_{vmax}	大于峰值 60%速度占比

四、问题一：数据预处理

4.1 问题分析

从问题描述中可知，原始数据中包括且不限于题目描述中的 5 中不良数据。本题的要求就是依据这些不良数据的类型，对原始数据中缺失的值进行插值补充；错误的值进行合理修正；异常的数据进行删除。具体做法如下：

1) GPS 信号丢失，造成提供数据中的时间不连续，对于 $T_{lose} \leq 180s$ 的数据，对速度信息进行指数插值，对其他信息进行线性插值。

2) 对于汽车加速度异常的数据，先依据 GPS 车速和时间信息进行加速度信息的扩充，然后对其进行修正，同时对速度也进行迭代修正。

3) 长期停车，也就是汽车速度和经纬度坐标在一定的时间内基本不变的情况下，所采集的数据，这一部分数据应该剔除。

4) 长时间堵车、断断续续的低速行驶可以视为怠速情况，对于这一部分同样也是依据汽车速度以及经纬度坐标在一定时间内变化较小的情况下，将速度置 0，经纬度坐标与连续低速数据段开始时的经纬度一致。

5) 对怠速时间 $T_{idle} \geq 180s$ 的情况，将 180 秒后的数据全部删除。

4.2 数据预处理方法以及实现

对于上述分析，将问题的求解过程分为：**数据插值与预分割、数据扩充、异常处理和数据修正**。这三个步骤经过设计，在数据插值和分割以及扩充后，进行异常处理和数据修正，在修正原始数据的同时防止引入新的不良数据。

4.2.1 数据插值与预分割

以空缺的时间片段长度 $T_{lose} \geq 180s$ 为界限（缺失片段长度在第二章进行了论述），文件 1 分割为 100 段，文件 2 分割为 44 段，文件 3 分割为 68 段。对于其中的每个数据段内部时间片段确实的部分，分别进行插值。对速度进行指数插值，插值模型为：

$$v = \begin{cases} e^{at} + b & v_{last} \leq v_{next} \\ -1/e^{at} + b & v_{last} > v_{next} \end{cases} \quad (1)$$

指数插值中系数 a 和系数 b 的求解：

$$\begin{cases} e^{at_1} + b = v_{last} \\ e^{at_2} + b = v_{next} \end{cases} \quad \text{其中 } t_1 = 0, t_2 = t_1 + T_{lost} \quad (2)$$

⇓

$$a = \ln(v_{next} - v_{last} + 1) / T_{lost}, \quad b = v_{last} - 1 \quad (3)$$

$$\begin{cases} -\frac{1}{e^{at_1}} + b = v_{last} \\ -\frac{1}{e^{at_2}} + b = v_{next} \end{cases} \quad \text{其中 } t_1 = 0, t_2 = t_1 + T_{lsot} \quad (4)$$

↓

$$a = \frac{\ln\left(\frac{1}{-v_{next} + v_{last} + 1}\right)}{T_{lsot}}, \quad b = v_{last} + 1 \quad (5)$$

指数插值算法求解步骤：

- Step1: 判断原始数据断点前速度是否断点后数据。
Step2: 调用相应的模型求解算法。
Step3: 求解算法根据计算公式 (3、4) 计算得到系数 a 和系数 b 的值。
Step4: 调用公式 (1) 中的模型，计算 t 从 0 到 T_{lsot} ，以 1 为间隔的值。
Step5: 将计算出的结果按照时间顺序填充到原始数据的 excel 文件中。

对于 X 轴加速度、Y 轴加速度、Z 轴加速度、经度、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、进气流量等数据，使用线性插值，插值模型为：

$$y = at + b \quad (6)$$

线性插值中常数 a 和常数 b 的求解：

$$\begin{cases} at_1 + b = v_{last} \\ at_2 + b = v_{next} \end{cases} \quad \text{其中 } t_1 = 0, t_2 = t_1 + T_{lsot} \quad (7)$$

↓

$$a = \frac{v_{next} - v_{last}}{T_{lsot}}, \quad b = v_{last} \quad (8)$$

线性插值算法求解步骤：

- Step1: 依据公式 (8) 计算得到系数 a 和系数 b。
Step2: 调用公式 (7) 中的模型，计算 t 从 0 到 T_{lsot} ，以 1 为间隔的值。
Step3: 将计算出的结果按照时间顺序填充到原始数据的 excel 文件中。

通过对附件中文件 1、文件 2 和文件 3 分割得到的 212 段数据分别进行插值，使得分割出来的数据中不再有时间不连续的片段。一共对原始数据插入 45624 行数据。插入数据长度与原始数据长度百分比约为 9.3%。

4.2.2 数据扩充

原始数据中包含了许多有价值的信息，根据这些信息的组合与计算，可以得出新的有效信息。根据时间-GPS 车速信息，可以扩充得到加速度信息；根据经纬度-时间信息，可以扩充得到位移信息；根据进气流量-空燃比信息，可以扩充得到新的瞬时油耗信息；根据 GPS 车速-位移信息，可以得到更加精确的停车判别信息。

表 1 数据扩充信息组合

信息组合对	时间	进气流量	位移
GPS 车速	加速度	—	停车判别
经纬度	位移	—	—
空燃比	—	瞬时油耗	—

加速度的计算公式为： $v_2 - v_1 / t$ 。数据插值与分割后，每段内部时间都是连续的，时间间隔为 1s，因此加速度的计算公式为：

$$v_2 - v_1, \text{ 其中 } v_2 \text{ 为下一个时间点的速度, } v_1 \text{ 为该时间点的速度} \quad (9)$$

依据此公式，迭代计算出每个时间点的加速度值，并在 excel 文件中的 GPS 速度信息后新加一列加速度信息。

位移的计算公式为： $x = x_2 - x_1$ 。同样的，根据时间信息，我们能知道下一个连续的时间点的坐标和本时间点的坐标。考虑到地球是一个曲面，计算曲面距离是需要考虑曲率半径。使用半正矢公式^[6~7]迭代计算出每个点相对上一个点的位移。公式如下：

$$haversin(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2} \quad (10)$$

$$haversin\left(\frac{d}{r}\right) = haversin(\phi_2 - \phi_1) + \cos(\phi_1)\cos(\phi_2)haversin(\lambda_2 - \lambda_1) \quad (11)$$

$$d = r \times haversin^{-1}(h) = 2r \times \arcsin(\sqrt{h}) \quad (12)$$

$$\begin{aligned} d &= 2r \times \arcsin\left(\sqrt{haversin(\phi_2 - \phi_1) + \cos(\phi_1)\cos(\phi_2)haversin(\lambda_2 - \lambda_1)}\right) \\ &= 2r \times \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \end{aligned} \quad (13)$$

在此使用了 excel 中的快速计算功能对位移数据进行填充，公式如下：

=6371004*SQRT(POWER(COS(G2*PI()/180)*(F3*PI()/180-F2*PI()/180),2)+POWER((G3*PI()/180-G2*PI()/180),2))

瞬时油耗信息虽然原数据中已经给出，但是经过第一章中的简要分析可知，原数据中的瞬时油耗与实际情况相比有非常大的出入。因此需要对瞬时油耗进行修正，修正的依据就是利用进气量和空燃比来计算出新的瞬时油耗，然后对比原始数据。对相差较大的进行修正，对相差不大的取均值。瞬时油耗的计算公式为：

$$\text{瞬时油耗} = \text{空燃比} \times \text{进气量} \quad (14)$$

停车判别的依据可以是速度或者位移，但是考虑到 GPS 信号的误差，应该对这两种信息进行综合考虑。判断前后各连续的 15 个时间点数据，如果速度为 0m/s 且位移也同为 0m 时，则可以认为这一点为停车。按时间顺序遍历，对所有判断为停车的时间点添加字段 1，

而判断为非停车的时间点添加 0。

4.2.3 异常数据处理和数据修正

经过上述两个步骤的处理，数据通过分割、插值和扩充操作，引入了不少不良数据。因此在这一步中，对引入的不良数据和原始的不良数据进行集中处理。需要处理的部分包括：汽车加速度 a_{pri} 异常、长期停车数据异常、瞬时油耗数据异常、长期低速行驶按照怠速处理、怠速时长超过 180 秒的片段的异常处理。

对于汽车加速度 a_{pri} 大于 3.97m/s^2 或者小于 $-8\sim 7.5\text{m/s}^2$ 的情况，对速度进行限定，高于上限加速度时，加速度就取 3.97m/s^2 。小于下限加速度时，下限就取 -8m/s^2 。然后依次对加速度变化造成的速度变化进行迭代更新。考虑到汽车行驶速度的上限基本为 60m/s ，从 0m/s 加速到最大速度也不过 16 秒，也就是 16 个数据点。减速度就更小了。因此，可以断言，加速度和速度的迭代更新只限于少量的数据，对大部分的数据影响甚微。事实上，实际修改的值也说明了修改范围不大。

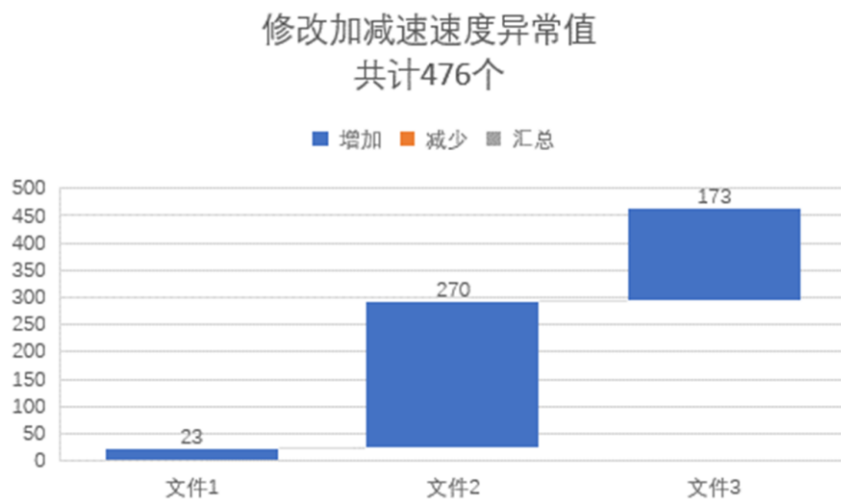


图 5 加速度修改数量

根据 4.2.2 小节中对停车判断的扩充，可以轻易地将停车等待的数据和其它如低速行驶、怠速运行等数据区分开。因为对所有判断为停车的时间点添加了字段 1，因此直接删除相应的时间节点即可。

同样，在 4.2.2 小节中对计算得到瞬时油耗进行了数据补充，根据与原始数据中的瞬时油耗进行对比。对相差较大的进行直接替换，对相差不大的取均值后替换。

根据要求，认为一段时间内堵车、断断续续低速行驶情况（速度小于 2.78m/s ）的情况可认为是怠速情况。对数据中的速度信息进行汇总统计，去除速度为 0m/s 的时刻。对于每个文件中判断为连续低速行驶的时间总长度统计如下：

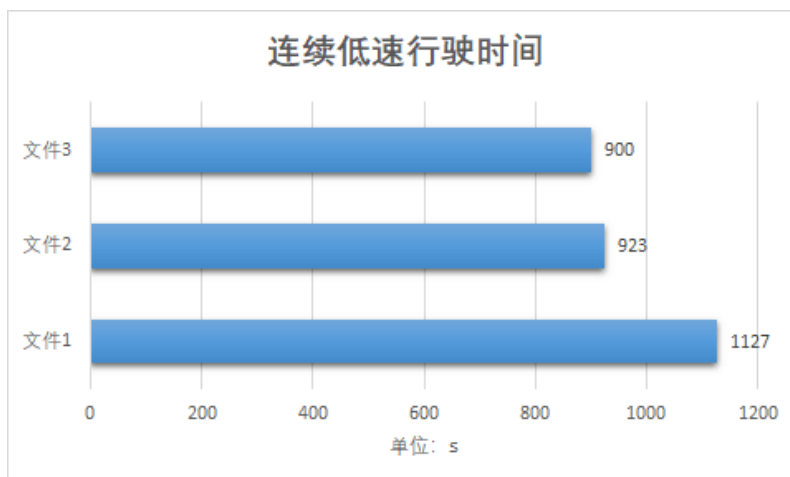


图 6 连续低速行驶时间统计

事实上，对文件进行统计发现，11-20 秒内的连续低速行驶时间占了绝大多数。但是 20 秒内的低速行驶符合城市交通实际情况，而且 20 秒内也算不上长时间的低速行驶。因此，将判断是否怠速的时间长度设置为 20 秒。超过这个时间段的低速行驶将一整段速度数据全置 0m/s，低于这一时间段低速行驶则不作处理。

经过上一步处理后，产生了许多怠速行驶片段。根据要求，需要将大于 180 秒的怠速片段截去。由于之前的处理，所以所有的怠速片段速度均为 0m/s。因此，只需要判断数据中是否存在连续的 180 个时间点以上的数据速度为 0，如果存在，就将 180 个点以后的为 0 的数据去除。对于每个文件中判断为怠速时间过长并去除的时间总长度统计如下：

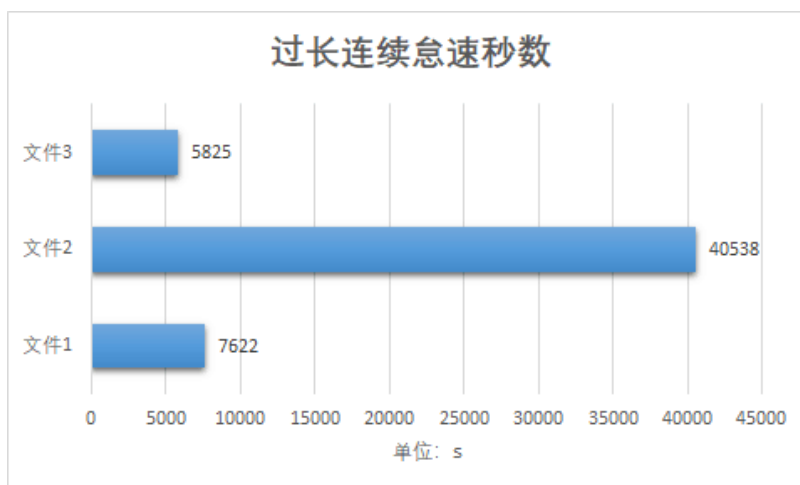


图 7 怠速时间长度统计

4.3 数据预处理结果分析

4.3.1 数据预处理小结

对于第一题的要求，也即数据预处理。首先对数据进行了总体分析，发现了三个比较主要的问题：1) 附件文件中有经纬度数据异常。2) 瞬时油耗的记录并不符合实际情况。3) 附件文件有许多时间片段不连续，且缺失片段长度 T_{lost} 从 2 秒到数小时不等。对于以上三种情况，分别采取了不同的措施。对于经纬度数据异常的点，丢失数据量小的进行了

插值，丢失数据量大的判断出与前后时间间隔较大，因此直接删除。对于瞬时油耗，根据空燃比和进气量重新计算出一组新的瞬时油耗，与原瞬时油耗相比，差异大的直接替换，差异小的取平均值替换。对于附件文件中的时间不连续现象，依据时间间隔的长短，对间隔短的数据，速度信息进行指数插值，其它信息进行线性插值。时间间隔长的进行分割。

其次，根据题目提出的不良数据的五种类型，设计了三个步骤：数据插值与分割、数据扩充、异常处理和数据修正。根据这三个步骤依次将五种不良数据进行修改，同时对插值和数据扩充带来的不良数据也一并修正。其中数据分割之前就已经完成，插值也依据速度或者其他信息分别进行了指数插值和线性插值。对于数据扩充，根据 4 对信息对，分别扩充得到 4 条新的信息。分别为：加速度、位移、新瞬时油耗和停车判别。数据异常处理和数据修正则是分别针对五种类型进行异常处理和修正。

对 GPS 信号丢失的数据进行时间间隔判断，大于 180 秒的进行分割，小于 180 秒的进行插值。对于汽车加速度 a_{pri} 超出要求范围内的数，按照边界值替换，同时对速度进行迭代修正，在此论证了加速度和速度的修正整体数据而言影响甚微。对于长期停车，依据可速度和位移的变化量，综合考虑前后各 15 个时间点，也就是 30 秒的时间数据。如果限定时间内速度均为 0m/s，位移均为 0，则判断这一段数据为停车，对数据进行删除。对长时间堵车、低速行驶的情况，设定时间阈值为 20 秒，大于 20 秒的片段，对着一整段速度数据全置为 0m/s。对于怠速时间 $T_{idle} \geq 180s$ 的片段，则直接删除。

上述内容有些部分执行顺序有交叉，有些部分操作步骤有重合。但对结果影响不大，下表是各数据删除、修改等操作的数量。有些是单个值，有些则是连续的片段算一个统计量。而新增数据则是一种类型算一个统计量。

表 2 数据处理数量统计

处理类型	处理数量	处理类型	处理数量
分割段	212	新增数据	4
经纬度异常	303	加速度异常	476
瞬时油耗异常	489573	长时间停车	187
GPS 信号丢失（插值）	45624	长时间低速行驶	148
怠速时间异常	300	—	—

4.3.2 数据预处理评价

在上述实验中，不仅完成了对题目指出的五种不良数据的预处理的操作，对每一种数据存在的问题都设计方法将其解决（4.2 节）。还对经纬度异常和瞬时油耗异常等非题目指出的数据异常进行了修正，对新数据进行建立。同时考虑到后续新特征的建立，以及后续依据怠速时间对运动学片段的提取，根据时间间隔对数据进行了预分割（预分割说明见第二章）。数据预处理部分考虑到了各种数据不良情况并进行解决，对后续操作也进行了数据扩充和分割。总的来说，还是比较成功的。

限于时间原因，有些部分没有办法实践更好的策略。在此列举出来：

1) GPS 信号丢失片段小于 180 秒时进行插值。对于速度，采用的是指数插值模型，对于其它信息采用的是线性插值模型。速度用指数插值模型事实上也基本满足实际情况，但是如果采用 5 次多项式插值，得到的插值曲线可能会更加平滑。至于其它信息，可能也有更好的插值办法。

2) 对于长时间低速行驶视为怠速情况的判别时间 T_{lsp} ，阈值设置为 20 秒。对于长时间停车视为异常数据的判别时间 T_{stop} ，阈值设置为 30 秒。这两个参数的设置都是基于统计分析得出的。因此可能存在更好的参数。

五、问题二：运动学片段的提取

5.1 问题分析

运动学片段是指汽车从怠速状态开始至下一个怠速状态开始之间的车速区间。在实际道路行驶中，车辆从起步出发至目的地停车，由于受道路交通状况影响，其间可能经过多次起步和停车等操作。车辆的行程即可视为各种片段的组合。其中某些片段反映的交通状况可能相同。不同的时间、地理位置和公路类型会出现相同的片段，有时繁忙的高速公路上的片段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，针对性地分析不同速度的运动形态，并在此基础上构建工况。

对于附件中的三个文件，就是实际的汽车行驶记录。考虑到实际情况中与一些偶然的停车导致速度为 0m/s，以及低速行驶不超过 30s 时一些速度值为 0m/s 的断点。因此将连续的 5 个怠速运行时间点作为运动学片段的开始和结尾标志。本题的运动学片段的提取，也是为了第三问服务。由于前一章对数据进行较好的预处理，因此本题运动学片段的提取相对轻松。

5.2 运动学片段提取与处理

5.2.1 运动学片段分割

对于分割出来的 212 个数据片段，逐一读取然后寻找连续的 5 个怠速运行时间片段，依次为开始，直到下一个连续的 5 个怠速运行时间片段出现未知，这就是一个运动学片段。

- Step1: 对独立的驾驶时间段的数据进行读取
- Step2: 检测五秒内速度是否持续为 0，若为 0 则该区间为怠速区间
- Step3: 循环检测数据文件中的怠速区间，分割为不同的片段
- Step4: 对每个独立的运动学片段进行保存
- Step5: 剔除持续时间过短的运动学片段
- Step6: 对结果进行保存，便于后续处理

5.2.2 分割片段处理

分割出来的运动学片段，对其中无效的片段进行处理后，一共有 2950 条，长度分布非常离散，从几十秒到两千多秒不等。在分割出的运动学片段中，存在一些持续时长过短的片段，这些片段没有足够的时长，提供的信息量过少，无法进行进一步的利用，对这些过短的片段进行删除。

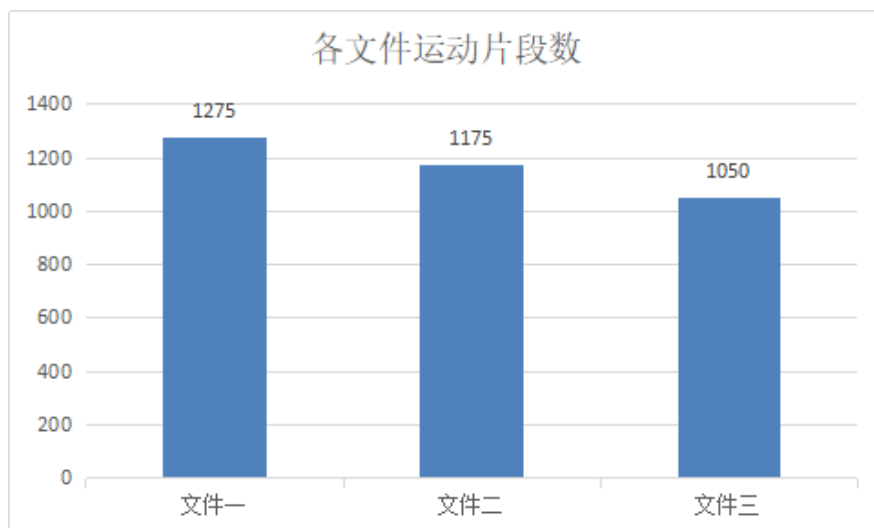


图 8 各文件运动片段数

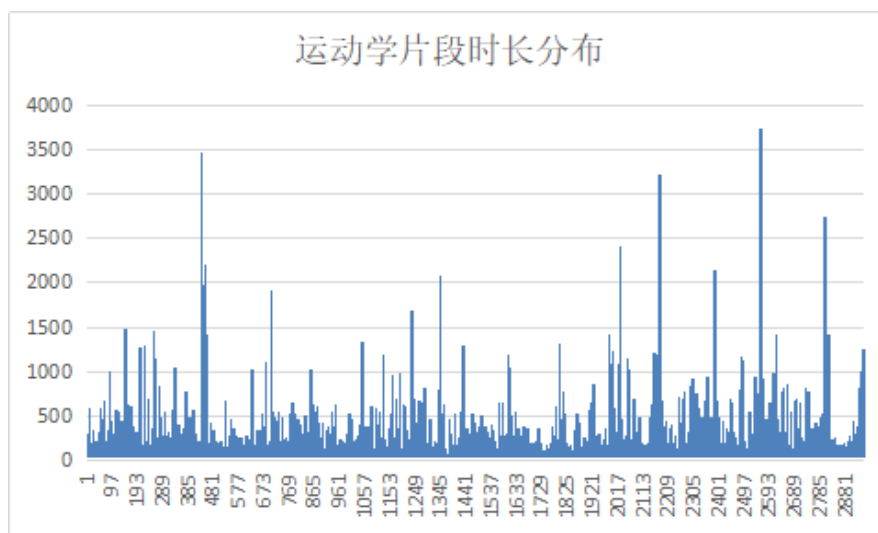


图 9 运动学片段时长分布

从上图可以看出，分割出的运动学片段时长分布范围很广，在各个区间内数量都比较多，说明运动学片段的分割方法比较合理，没有产生聚集现象。利用分割出的运动学片段，进行后续处理，能充分体现该数据的运动学特征。

5.3 运动学片段提取结果分析

5.3.1 运动学片段提取小结

运动学片段的提取依赖于第一题中的预处理，加之根据空缺时间片段的长度进行的预

分割，所以本题的运动学片段提取还是相对简单的。通过对独立的驾驶时间段的数据进行读取。检测五秒内速度是否持续为 0，若为 0 则该区间为怠速区间。循环检测数据文件中的怠速区间，分割为不同的片段。对每个独立的运动学片段进行保存。剔除持续时间过短的运动学片段。最后对结果进行保存。至此，运动学片段的提取就完成了。

5.3.2 运动学片段提取评价

在上述实验中，针对第二题的运动学片段的提取，设计了比较合理分段的模型。成功得从附录中的文件一、文件二、文件三中，提取出 2950 个完整运动学片段。考虑到用于构建问题三中所提出来的汽车行驶工况。在此对运动学片段进行了初步筛选，对于长度小于 20 秒的片段和长度大于 1300 秒的片段进行了删除。共删去 550 条运动学片段。总的来说，运动学片段的提取还是比较成功的。

限于时间原因，有些部分的处理可能需要更加谨慎一些。比如将 5 个连续的怠速片段作为运动学片段的开始和结束标志。这个参数的得出也是基于统计分析得来的。还有对于运动学片段的删除，小于 20 秒以下的片段删除基本上影响不大。但是大于 1300 秒的 22 条数据直接删除未免有点可惜了，若时间宽裕可以使用离散小波变换模型来对这部分的数据进行压缩。

六、问题三：汽车行驶工况的构建

6.1 问题分析

本题需要根据上述处理好的数据，构建一条能体现参与数据采集汽车行驶特征的汽车行驶工况曲线^[8]（1200-1300 秒），该曲线的汽车运动特征能代表所采集数据源（经处理后的数据）的相应特征，两者间的误差越小，说明所构建的汽车行驶工况的代表性越好。除了构建工况曲线外，还需要建立合理的评估体系对曲线进行评价。因此，题目中提出了三点要求：

1) 科学有效的构建方法。即对曲线的构建需要合情合理、有依有据。

2) 合理的汽车运动特征评估体系（至少包括但不限于以下指标：平均速度（km/h）、平均行驶速度（km/h）、平均加速度（m/s²）、平均减速度（m/s²）、怠速时间比（%）、加速时间比（%）、减速时间比（%）、速度标准差（km/h）、加速度标准差等（m/s²））。即建立的特征评估体系中需要对以上特征进行合理分析，除此之外还可以根据实际需求进行特征添加扩展。

3) 依据构建的汽车行驶工况及汽车运动特征评估体系，分别计算出汽车行驶工况与该城市所采集数据源（经处理后的数据）的各指标（运动特征）值，并说明所构建的汽车行驶工况的合理性。

对于上述三点要求，首先需要对得到的运动学片段进行处理，对特征进行扩充、对片段进行抽样。其次通过建立模型，提取特定的运动学片段来构建曲线。然后计算运动学片段的相应特征。最后与处理后的采集数据的统计特征进行多方面的比较。

6.2 工况曲线的构建

得益于第一题中较为严谨的数据预处理以及第二题对运动学片段的分离和选取。现在要做的工作主要集中在模型建立上。对于题目中给出的 9 种特征，依据特征间的关系以及统计分析，建立新的特征。对于工况曲线构建常用的两种模型，小波变换模型和聚类模型分别进行分析，然后比较选择。考虑到长片段的数量以及长片段可能代表的一类运动学片段，因此构建工况曲线上时，长片段也需要考虑其中。长片段参与公款曲线的合成时，由于长度过长，需要对其进行压缩。最后合成的工况曲线应该能代表多类运动学片段。

6.2.1 特征建立与扩展

根据第二问中分离出来的 2400 个运动学片段，依次对其进行特征建立可扩展。题目中给出的 9 种特征中，基本覆盖车辆行驶过程中的绝大部分信息。但是有些特征相对于 9 种基本特征来说应该补充，比如减速度标准差 (m/s^2)、速度峰值 (m/s)、速度中位数 (m/s)。而有些特征则是从统计分析的角度进行补充，比如大于峰值 60% 速度占比 (%)，运动学片段的时长。

通过特征扩充，得到一共 14 组特征。特征建立计算过程各不相同。特征的计算公式为：

表 3 各特征计算公式

需要计算的量	公式
平均速度	$v_{seq} = \frac{\sum_{i=1}^{T_{seq}} v_i}{T_{seq}}$
平均行驶速度	$v_{seqz} = \frac{\sum_{i=1}^{T_{seq}} v_i}{T_{seq}}, v_i \text{ 不等于 } 0$
平均加速度	$a_{useq} = \frac{\sum_{i=1}^{T_{seq}} a_i}{T_{seq}}, a_i = v_{seq} - v_i, \text{ 且 } a_i \geq 0$
平均减速度	$a_{dseq} = \frac{\sum_{i=1}^{T_{seq}} a_i}{T_{seq}}, a_i = v_{seq} - v_i, \text{ 且 } a_i < 0$

怠速时间	$P_{idle} = \frac{\sum_{i=1}^{T_{seq}} T_{idle}}{T_{seq}}$
加速度时间比	$P_{au} = \frac{\sum_{i=1}^{T_{seq}} T_{a_i}}{T_{seq}}, \quad a_i > 0$
减速度时间比	$P_{ad} = \frac{\sum_{i=1}^{T_{seq}} T_{a_i}}{T_{seq}}, \quad a_i \leq 0$
速度标准差	$v_{csd} = \frac{\sum_{i=1}^{T_{seq}} \sqrt{(v_i - v_{seq})^2}}{T_{seq}}$
加速度标准差	$a_{ucsd} = \frac{\sum_{i=1}^{T_{seq}} \sqrt{(a_i - a_{useq})^2}}{T_{seq}}$
减速度标准差	$a_{dcsd} = \frac{\sum_{i=1}^{T_{seq}} \sqrt{(a_i - a_{dseq})^2}}{T_{seq}}$
速度峰值	$v_{\max} = \max(v_i)$
大于速度峰值 60%占比	$P_{v\max} = \frac{\sum_{i=1}^{T_{seq}} (T_{v_i})}{T_{seq}}, \quad v_i > 60\% \times v_{\max}$
速度中位数	$v_{mid} = rank(v_i), \quad i = T_{seq}$
片段长度	$T_{seq}, \text{ 统计量}$

上表中给出了 14 个特征的计算公式，其中基础特征 9 个，扩充特征 5 个。依据公式可

以轻松计算出每个运动学片段的特征具体数值。截取部分数据展示如下：

id	1平均速度	2平均行驶速	3平均加速度	4平均减速度	5怠速时间比	6加速时间比	7减速时间	8速度标准	9加速度标	10减速度标准差	11速度峰值	12大于峰值	13速度中位	14片段时
1	8.611627907	10.5796977	0.473587432	-0.947127509	0.18604651	0.4651163	0.232558	5.831359	0.245395	1.008206403	17.4	0.465116	9.6	44
2	0.096875	1.54922539	0.471750472	-0.235993115	0.9375	0.03125	0.0625	0.377065	0.000472	0.152739643	1.7	0.0625	0	33
3	16.49626168	24.1791208	0.248786862	-0.497562908	0.31775701	0.4299065	0.214953	14.60777	0.276865	0.481875096	39.4	0.383178	17.7	108
4	18.95677966	23.5460679	0.504717394	-0.539129161	0.19491525	0.3983051	0.372881	13.3888	0.385379	0.442870916	46.3	0.237288	22.6	119
5	10.03134328	22.7826647	0.317194069	-0.409705151	0.55970149	0.2313433	0.179104	12.36366	0.395532	0.454820712	30.1	0.358209	0	135
6	0.637931034	2.31221097	0.458104281	-0.152752319	0.72413793	0.0689655	0.206897	1.056152	0.319365	0.154439237	2.8	0.275862	0	30
7	6.076829643	15.0651596	0.666629632	-0.428556123	0.59663866	0.1512605	0.235294	8.193108	0.739857	0.689950543	28.5	0.142857	0	120
8	10.39783784	19.045356	0.3349933	-0.398800029	0.45405405	0.2702703	0.227027	11.38313	0.309088	0.387266279	30.1	0.372973	5.2	186
9	12.68913043	14.0648908	0.366153294	-0.473842273	0.09782609	0.4782609	0.369565	9.43326	0.473266	0.415402732	25.5	0.48913	13.5	93
10	4.880952381	8.91265597	0.455093198	-0.591607506	0.45238095	0.3095238	0.238095	5.0791	0.326764	0.786130698	13.8	0.309524	5	43
11	1.284615385	4.45318489	0.248524931	-0.363219923	0.71153846	0.1826923	0.125	2.286111	0.235605	0.253048848	8	0.105769	0	105
12	6.362222222	8.17976629	0.329150209	-0.470204509	0.22222222	0.4444444	0.311111	4.539373	0.261794	0.362235622	14.5	0.377778	7	46
13	8.542307692	9.45086275	0.359938706	-0.375587535	0.09615385	0.4615385	0.442308	5.995701	0.490543	0.374543797	16.3	0.403846	8.5	53
14	27.5245098	27.52424	0.212296533	-0.295345446	0	0.4117647	0.480392	8.685749	0.131665	0.357698464	37.3	0.705882	30.2	103
15	15.56271186	19.9604356	0.52544107	-0.600500505	0.22033898	0.4067797	0.355932	12.21254	0.399638	0.435285373	32.4	0.440678	16	60
16	1.336666667	5.72775318	0.652614624	-0.652614624	0.76666667	0.1333333	0.133333	2.797318	0.494702	0.273897521	9.4	0.133333	0	31
17	1.716666667	3.168987	0.22681405	-0.194416671	0.45833333	0.25	0.291667	1.677465	0.21718	0.192436347	4.4	0.375	2.3	25
18	0.170833333	2.04897551	0.638250638	-0.319284802	0.91666667	0.0416667	0.083333	0.571168	0.000638	0.180510504	2.3	0.083333	0	25
19	3.6	5.27982401	0.39019306	-0.910392672	0.31818182	0.4772727	0.204545	3.692068	0.34326	0.703028131	12.4	0.204545	2.5	45
20	0.045454545	1.4985015	0.416250416	-0.416250416	0.96969697	0.030303	0.030303	0.25713	0.000416	0.000416042	1.5	0.030303	0	34
21	0.670833333	2.29967149	0.208281263	-0.416458437	0.70833333	0.1666667	0.083333	1.107354	0.181598	0.305479266	3	0.166667	0	25
22	4.508546793	7.76428813	0.47563499	-0.422792529	0.41935484	0.2580645	0.290323	4.643648	0.268242	0.235971894	13	0.354839	2.8	32
23	6.162711864	11.7286539	0.518483953	-0.518483953	0.47457627	0.2542373	0.254237	8.295637	0.681499	0.627590225	26.4	0.220339	2.4	60
24	6.881081081	14.1436587	0.486070605	-0.833214303	0.51351351	0.3243243	0.189189	8.100078	0.262657	0.664797923	21	0.297297	0	38
25	33.86734007	34.3295757	0.200311578	-0.318284385	0.01346801	0.4781145	0.40404	11.74681	0.192021	0.334735155	47.3	0.787879	37.4	298
26	9.563809524	16.7363877	0.504324518	-0.645529734	0.42857143	0.3047619	0.238095	11.3678	0.502451	0.629129495	39.2	0.12381	7.5	106
27	16.60769231	20.5611452	0.550238348	-0.525228651	0.19230769	0.4038462	0.423077	13.32886	0.775423	0.441020831	36.7	0.365385	16.2	53
28	6.023076923	9.21122287	0.591607506	-0.739490897	0.34615385	0.3846154	0.307692	6.448274	0.8307	0.488245922	16.6	0.423077	3.7	27
29	11.51097805	18.9106995	0.400642895	-0.418061773	0.39130435	0.2981366	0.285714	10.62033	0.445734	0.350837999	29.8	0.329193	13.4	162
30	0.135294118	3.44827586	0.513632073	-1.026751027	0.96078431	0.0392157	0.019608	0.671499	0.374906	0.001026238	3.7	0.039216	0	52
31	0.868	2.41084324	0.430340385	-0.107625436	0.64	0.08	0.32	1.201572	0.291594	0.120624078	3.1	0.28	0	26
32	3.2625	10.439304	0.54357314	-0.54357314	0.6875	0.1458333	0.145833	4.860625	0.977845	0.95492161	12.6	0.3125	0	49
33	7.469060773	11.3604087	0.399683956	-0.365813009	0.34254144	0.2983425	0.325967	7.359553	0.430275	0.285674006	24.5	0.19337	7.3	182
34	5.697139948	21.3272078	0.451366321	-0.531014515	0.73287671	0.1369863	0.116438	10.46554	0.589584	0.425426981	31.9	0.150685	0	147
35	15.32632827	24.5218374	0.452494364	-0.26975977	0.375	0.2279412	0.382353	13.38489	0.497406	0.238807916	34.3	0.477941	17.77384	137
36	24.87537672	26.5336369	0.116031083	-0.278154485	0.0625	0.6076389	0.253472	11.26469	0.167963	0.38348049	35.9	0.715278	30.79383	289
37	0.03255814	1.3986014	0.388500389	-0.388500389	0.97674419	0.0232558	0.023256	0.211001	0.000388	0.000388306	1.4	0.023256	0	44
38	4.548623853	9.01438156	0.513262708	-0.421611927	0.49541284	0.2110092	0.256881	5.454408	0.610926	0.638189302	17.7	0.192661	3	110
39	0.01875	0.19993336	0.041645844	-0.041645844	0.90625	0.0625	0.0625	0.063431	0.013885	0.013885434	0.3	0.0625	0	33
40	18.31258041	20.3985655	0.457322021	-0.600230687	0.10227273	0.4772727	0.363636	12.95596	0.481499	0.927230101	40.1	0.431818	16.5	89
41	14.25294118	16.904258	0.585708201	-0.471045188	0.15686275	0.3529412	0.470588	9.416576	0.912251	0.296631266	28.9	0.431373	15.8	52
42	18.49224138	24.1019764	0.427620326	-0.345737325	0.23275862	0.3275862	0.405172	13.0285	0.304522	0.281968845	40.5	0.431034	22.5	117
43	15.97487437	19.6233357	0.399995	-0.415579018	0.18592965	0.4020101	0.386935	11.56914	0.327334	0.300792035	39.3	0.301508	16.2	200
44	2.930902381	9.46850242	0.423558166	-0.677642249	0.69047619	0.1904762	0.119048	4.499285	0.385815	1.10690957	11.1	0.261905	0	43
45	4.04375	11.4169755	0.552014332	-0.552014332	0.64583333	0.1666667	0.166667	5.985361	0.884203	0.399974006	15.8	0.291667	0	49
46	0.051851852	1.39953349	0.388500389	-0.388500389	0.96296296	0.0123457	0.012346	0.264394	0.000388	0.000388306	1.4	0.037037	0	82
47	11.96923077	18.0403183	0.438259431	-0.563471939	0.33653846	0.3461538	0.269231	11.51743	0.231489	0.577046688	32.4	0.326923	9.5	105
48	11.66583444	18.8445679	0.186220455	-0.264284837	0.38095238	0.3492063	0.246032	10.78892	0.162392	0.480338902	28.6	0.365079	11.7	127

图 10 运动学片段新加特征部分数据截取

6.2.2 小波变换模型

提取特征后，每个运动学片段的模型已经建立起来，可以依据这些特征进行构建工程曲线。参考文献[1]提出的基于小波变换方法来构建汽车行驶工况曲线。该方法使用离散小波变换（Discrete Wavelet Transform, DWT）模型，离散小波变换和离散傅里叶变换（Discrete Fourier Transform, DFT）不一样。DWT 是基于 Mallat 算法实现的，针对的离散时间信号，而 DWT 指的是将连续小波变换（Continuous Wavelet Transform, CWT）中的尺度参数 a 和时移参数 b 离散化，即将下式中的 a 和 b 离散化，但 t 仍然是连续的。

$$W_f(a,b)=\frac{1}{\sqrt{|a|}}\int_{-\infty}^{\infty}f(t)\overline{\Psi\left(\frac{t-b}{a}\right)}dt$$

(15)

Mallat 算法框图如下：

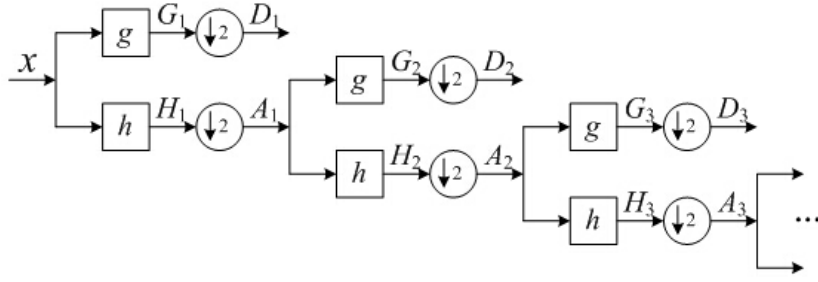


图 11 Mallat 算法框图

小波变换跟普通的正交变换不同，它是一个多层分解。如 Mallat 算法框图所示，一个长为 N 的信号 x ，第一层分解为高频部分 D_1 和低频部分 A_1 ，长度均为 $N/2$ ；第二层分解将 A_1 分解为高频部分 D_2 和低频部分 A_2 ，长度均为 $N/4$ ；第三层分解将 A_2 分解为高频部分 D_3 和低频部分 A_3 ，长度均为 $N/8$ ，依此类推，但分解所得到的所有结果长度总和仍为 N ，例如一层分解后得到 D_1 和 A_1 （两个长度为 $N/2$ 的序列），二层分解后得到 D_1 、 D_2 和 A_2 （ $N/2 + N/4 + N/4$ ），三层分解后得到 D_1 、 D_2 、 D_3 和 A_3 （ $N/2 + N/4 + N/8 + N/8$ ）。

在 Mallat 算法框图中，多层分解的过程相当于滤波的过程，例如由 x_0 分别得到 d_1 和 x_1 相当于让 x_0 分别通过滤波器 h_1 和 h_0 ，然后再下采样，每两个点抽取一个点。滤波的过程实际上就是卷积了，但这里就有一个问题： N 点长的序列与 M 点长的序列线性卷积后长度为 $N+M-1$ ，并不等于 N ，也就是说下采样后每两点抽取一点得到的结果并不是 $N/2$ ，而且线性卷积的四个步骤（反转、平移、相乘、相加）在计算卷积结果两侧的点时，反转平移后两个卷积序列只有部分点对应上。在 Mallat 算法框图中， $G_1(n) = x(n) * g(n)$ ， $H_1(n) = x(n) * h(n)$ ，其中“ $*$ ”表示卷积。

$$H_1(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} h(n-k) x(k) \quad (16)$$

$$G_1(n) = x(n) * g(n) = \sum_{k=-\infty}^{\infty} g(n-k) x(k) \quad (17)$$

设 $x(n)$ 的长度为 N ， $h(n)$ 的长度为 M ，则 $H_1(n)$ 的长度为 $N+M-1$ （假设 $N > M$ ），以上卷积关系写为矩阵形式：（ $H_1(n)$ 以 $y(n)$ 表示）：

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(M-1) \\ \vdots \\ y(N-1) \\ \vdots \\ y(N+M-2) \end{bmatrix} = \begin{bmatrix} h(0) & h(-1) & \cdots & h(-M+1) & \cdots & h(-N+1) \\ h(1) & h(0) & \cdots & h(-M+2) & \cdots & h(-N+2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ h(M-1) & h(M-2) & \cdots & h(0) & \cdots & h(M-N) \\ h(M) & h(M-1) & \cdots & h(1) & \cdots & h(M-N+1) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ h(N-1) & h(N-2) & \cdots & h(M-1) & \cdots & h(0) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ h(N+M-2) & h(N+M-2) & \cdots & h(2M-1) & \cdots & h(M-1) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(M-1) \\ \vdots \\ x(N-1) \end{bmatrix} \quad (18)$$

另外，为了实现多层分解，只要用一个矩阵连乘就可以了。从 x 得到 D_1 和 A_1 的过程（即第一层分解）可以用矩阵表示为：

$$\begin{bmatrix} [A_1]_{\frac{N}{2} \times 1} \\ [D_1]_{\frac{N}{2} \times 1} \end{bmatrix} = [W_1]_{N \times N} \bullet [x]_{N \times 1} \quad (19)$$

第二层分解是由 A_1 得到 D_2 和 A_2 的过程，可用矩阵表示如下：

$$\begin{bmatrix} [A_2]_{\frac{N}{4} \times 1} \\ [D_2]_{\frac{N}{4} \times 1} \end{bmatrix} = [W_2]_{\frac{N}{2} \times \frac{N}{2}} \bullet [A_1]_{\frac{N}{2} \times 1} \quad (20)$$

若把第一层分解和第二层分解写在一起，可用分块矩阵的概念表示如下：

$$\begin{bmatrix} [A_2]_{\frac{N}{4} \times 1} \\ [D_2]_{\frac{N}{4} \times 1} \\ [D_1]_{\frac{N}{2} \times 1} \end{bmatrix} = \begin{bmatrix} [W_2]_{\frac{N}{2} \times \frac{N}{2}} \bullet [A_1]_{\frac{N}{2} \times 1} \\ [D_1]_{\frac{N}{2} \times 1} \end{bmatrix} = \begin{bmatrix} [W_2]_{\frac{N}{2} \times \frac{N}{2}} & 0_{\frac{N}{2} \times \frac{N}{2}} \\ 0_{\frac{N}{2} \times \frac{N}{2}} & I_{\frac{N}{2} \times \frac{N}{2}} \end{bmatrix} \bullet \begin{bmatrix} [A_1]_{\frac{N}{2} \times 1} \\ [D_1]_{\frac{N}{2} \times 1} \end{bmatrix} \quad (21)$$

$$= \begin{bmatrix} [W_2]_{\frac{N}{2} \times \frac{N}{2}} & 0_{\frac{N}{2} \times \frac{N}{2}} \\ 0_{\frac{N}{2} \times \frac{N}{2}} & I_{\frac{N}{2} \times \frac{N}{2}} \end{bmatrix} \bullet [W_1]_{N \times N} \bullet [x]_{N \times 1} \quad (22)$$

第三层分解以此规律类推即可。

按照上面的原理对汽车行驶工况信号进行小波变换，得到相应的小波分析系数，对高频系数进行阈值处理，没一尺度的高频信号选择同阈值。高频小波变换系数阈值根据相应尺度上小波变换系数的最大绝对值来确定。各尺度上保留绝对值大于等于阈值的系数。

小波变换作为一种比较新的数据压缩理论，拥有与传统压缩方法类似的一些优点，具有高压缩比的特点。但是，考虑到小波变换实现难度较大。而且最重要的是，**本题的要求不是直接使用原始数据对汽车行驶工况曲线进行构建。而是使用预处理和分割好的运动学片段对工况曲线进行构建。**并且经过运动学片段的分割以及提出 1300 秒以上的运动学片段 22 条后。数据长度都较低，因此也不需要用到小波变换模型。

6.2.3 主成分分析模型

对小波变化进行理论分析后，判断在该题中不需要用到小波变换模型。因此参考文献[8]提出基于主成分分析^[9]和聚类分析模型^[10]的工况曲线构建。

主成分分析是采用一种数学降维的方法，将多项指标转化为少数几项综合指标，用综合指标来解释多变量的方差-协方差结构。综合指标即为主成分。所得出的少数几个主成分，要尽可能多地保留原始变量的信息，且彼此不相关。

主成分分析法是用少数的几个变量(因子)来综合反映原始变量(因子)的主要信息，变量虽然较原始变量少，但所包含的信息量却占原始信息的 95% 以上，所以即使用少数的几个新变量，可信度也很高，也可以有效地解释问题。并且新的变量彼此间互不相关，消除了多重共线性。其一般流程如图所示：

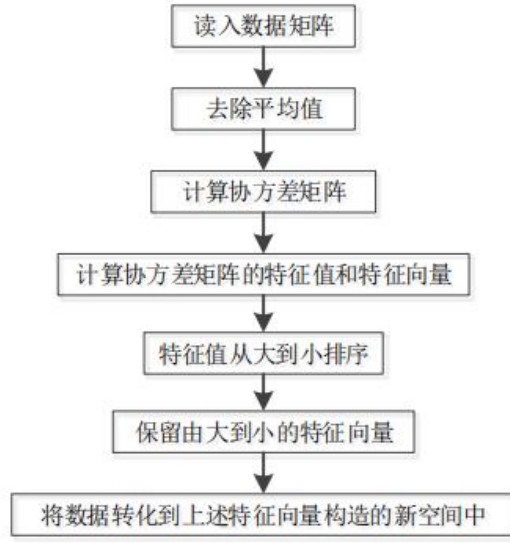


图 12 PCA 分析原理图

1) 对原始数据进行预处理后, 假设样本的观测矩阵如下:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \quad (23)$$

按照如下方法对数据进行标准化处理:

$$x_{ij}^* = \frac{x_{ij} - \bar{x}_j}{\sqrt{\text{Var}(x_j)}} \quad (i = 1, 2, 3 \cdots n; j = 1, 2, 3 \cdots p) \quad (24)$$

其中:

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij}, \text{Var}(x_j) = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2 \quad (j = 1, 2, 3, \cdots p) \quad (25)$$

2) 计算相关样本系数矩阵。

对原始数据进行标准化处理后的数据的相关系数矩阵如下:

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ r_{n1} & r_{n2} & \cdots & r_{np} \end{bmatrix} \quad (26)$$

其中:

$$r_{ij} = \frac{\text{Cov}(x_i, x_j)}{\sqrt{\text{Var}(x_i)}\sqrt{\text{Var}(x_j)}} = \frac{\sum_{k=1}^n (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\sqrt{\sum_{k=1}^n (x_{ki} - \bar{x}_i)^2} \sqrt{\sum_{k=1}^n (x_{kj} - \bar{x}_j)^2}} \quad (27)$$

3) 计算相关系数 R 的特征值和对应的特征向量。

4) 选择重要的主成分，并写出主成分的表达式。

主成分分析可以得到 p 个主成分但是由于各个主成分的方差是递减的，包含的信息量也是递减的，所以实际分析时一限不限选取 p 个主成分，而是根据各个主成分累计贡献率的大小选取前 k 个主成分。这里贡献率是指某个主成分的方差占全部方差的比重，实际也就是某个特征值占全部特征值合计的比重，所以：

$$\text{贡献率} = \frac{\lambda_i}{\sum_i^p \lambda_i} \quad (28)$$

贡献率越大，说明该主成分所包含的原始变量的信息越强。主成分个数 k 的选取主要根据主成分的累计贡献率来决定，一般要求累计贡献率达到 85% 上，这样才能保证处理后的变最能包括原始变量的绝大多数信息

5) 计算主成分得分。

根据标准化的原始数据，分析带入主成分表达式就可以得到各主成分下各个样本的新数据，得到的主成分具体形式如下：

$$\begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1k} \\ F_{21} & F_{22} & \cdots & F_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ F_{n1} & F_{n2} & \cdots & F_{nk} \end{bmatrix} \quad (29)$$

6) 依据主成分得分的数据，进一步对问题进行后续分析与建模。依照上述理论建立的主成分分析模型，求解各个主成分向量所对应的特征值，贡献率以及累计贡献率如表所示：

表 4 14 个特征的贡献率及累计贡献率

总方差解释						
成分	初始特征值			提取载荷平方和		
	总计	方差百分比	累积 %	总计	方差百分比	累积 %
1	7.341	52.435	52.435	7.341	52.435	52.435
2	2.275	16.252	68.687	2.275	16.252	68.687
3	1.403	10.024	78.711	1.403	10.024	78.711
4	.894	6.387	85.098			
5	.659	4.710	89.808			
6	.442	3.159	92.967			
7	.306	2.187	95.154			
8	.297	2.121	97.274			
9	.178	1.268	98.543			
10	.122	.870	99.413			
11	.043	.306	99.718			
12	.023	.164	99.882			
13	.010	.071	99.953			
14	.007	.047	100.000			

提取方法：主成分分析法。

可以看到，一共有 14 个特征，但是前七个特征的累积贡献率就达到了 95% 以上。这七个特征分别为：平均行驶速度 (km/h)、平均加速度 (m/s²)、平均减速度 (m/s²)、怠速时间比 (%)、加速时间比 (%)、速度标准差 (km/h)、大于峰值 60% 速度占比 (%)。

因此，可以认为用这 7 个特征进行聚类分析和后续的工况曲线构建是具有代表性的。

6.2.4 K-means 聚类模型

k 均值聚类算法 (k-means clustering algorithm) 是一种迭代求解的聚类分析算法，其步骤是随机选取 K 个对象作为初始的聚类中心，然后计算每个对象与各个种子聚类中心之间的距离，把每个对象分配给距离它最近的聚类中心。

该算法的主要思想是将一堆没有标签的数据自动划分成几类的方法，属于无监督学习方法，这个方法要保证同一类的数据有相似的特征，如下图所示：

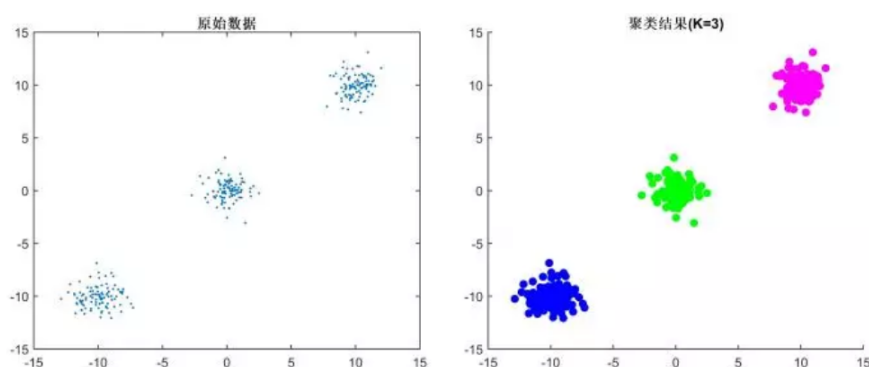


图 13 聚类示意图

根据样本之间的距离或者说是相似性（亲疏性），把越相似、差异越小的样本聚成一类（簇），最后形成多个簇，使同一个簇内部的样本相似度高，不同簇之间差异性高。基本步骤如下：

- 1、首先确定一个 k 值，即我们希望将数据集经过聚类得到 k 个集合。
- 2、从数据集中随机选择 k 个数据点作为质心。
- 3、对数据集中每一个点，计算其与每一个质心的距离（如欧式距离），离哪个质心近，就划分到那个质心所属的集合。
- 4、把所有数据归好集合后，一共有 k 个集合。然后重新计算每个集合的质心。
- 5、如果新计算出来的质心和原来的质心之间的距离小于某一个设置的阈值（表示重新计算的质心的位置变化不大，趋于稳定，或者说收敛），我们可以认为聚类已经达到期望的结果，算法终止。
- 6、如果新质心和原质心距离变化很大，需要迭代 3~5 步骤。

K-means 基本原理如下图所示：

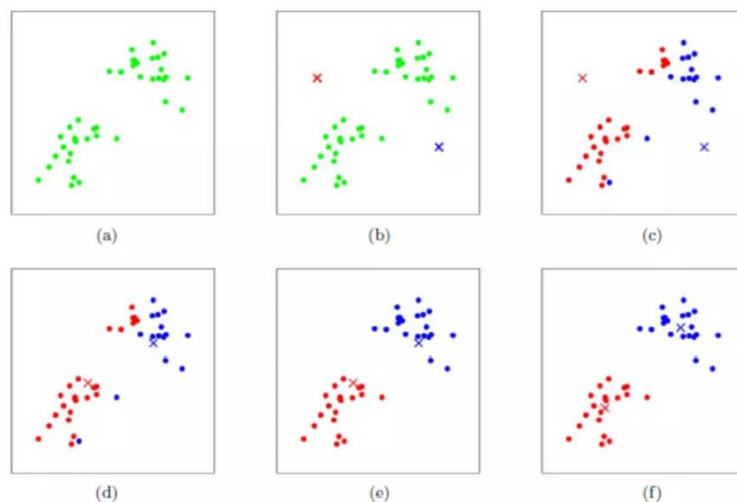


图 14 K-means 聚类示意图

上图 a 表达了初始的数据集，假设 $k=2$ 。在图 b 中，我们随机选择了两个 k 类所对应的类别质心，即图中的红色质心和蓝色质心，然后分别求样本中所有点到这两个质心的距离，并标记每个样本的类别为和该样本距离最小的质心的类别，如图 c 所示，经过计算样本和红色质心和蓝色质心的距离，我们得到了所有样本点的第一轮迭代后的类别。此时我们对当前标记为红色和蓝色的点分别求其新的质心，如图 d 所示，新的红色质心和蓝色质心的位置已经发生了变动。图 e 和图 f 重复了我们在图 c 和图 d 的过程，即将所有点的类别标记为距离最近的质心的类别并求新的质心。最终我们得到的两个类别如图 f。

根据主成分分析选取的 7 个特征进行聚类分析，分别选择 3 类、4 类和 5 类。聚类结果如下图所示：

最终聚类中心之间的距离				每个聚类中的个案数目	
聚类	1	2	3	聚类	
1		20.067	20.849	1	1252.000
2	20.067		40.797	2	1209.000
3	20.849	40.797		3	466.000
				有效	2927.000
				缺失	.000

图 15 3 类聚类结果

最终聚类中心之间的距离					每个聚类中的个案数目	
聚类	1	2	3	4	聚类	
1		53.193	20.528	35.998	1	148.000
2	53.193		32.828	17.387	2	1048.000
3	20.528	32.828		15.508	3	658.000
4	35.998	17.387	15.508		4	1073.000
					有效	2927.000
					缺失	.000

图 16 4 类聚类结果

最终聚类中心之间的距离						每个聚类中的个案数目	
聚类	1	2	3	4	5	聚类	
1		44.267	28.344	14.713	21.385	1	257.000
2	44.267		16.027	29.637	65.048	2	962.000
3	28.344	16.027		13.650	49.382	3	970.000
4	14.713	29.637	13.650		35.918	4	695.000
5	21.385	65.048	49.382	35.918		5	43.000
						有效	2927.000
						缺失	.000

图 17 5 类聚类结果

从每个聚类中个案数目统图中可以看出。3 类聚类中，每类分别为 1252、1209、466。4 类聚类中，每类分别为 148、1048、658、1073。5 类聚类中，每类分别为 257、962、970、695、43。对比来看 3 类聚类分离的结果是比较均匀的，其次是 4 类。

从最终聚类中心之间的距离统计图中可以看出。3 类聚类的分离距离也是最大的。其次是 4 类。

综合来看分类效果最好的是 3 类聚类，其次是 4 类聚类。但是考虑到运动学片段的复杂性，3 类聚类选取运动学片段进行工程曲线的拟合可能不能有较强的代表性，因此最终选择 4 类聚类作为分类数目。

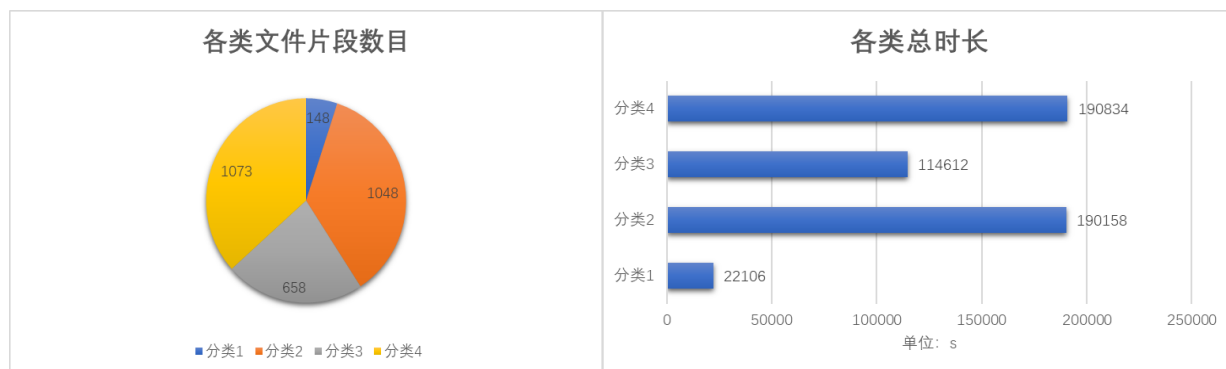


图 18 4 类聚类各文件分段数目与时长



图 19 聚类后各个特征归一化对比图

id	1平均速度	2平均行驶速	3平均加速度	4平均减速度	5总速时间比	6加速时间比	7减速度比	8速度标准	9加速度标准	10减速度标准差	11速度峰值	12大于峰值	13速度中位	14片段时长	3类	4类	5类
1	8.611627907	10.5796977	0.473587432	-0.947127509	0.18604651	0.4651163	0.232558	5.831359	0.245395	1.008206403	17.4	0.465116	9.6	44	2	2	2
2	0.096875	1.54922539	0.471750472	-0.235993115	0.9375	0.03125	0.0625	0.377065	0.000472	0.152739643	1.7	0.0625	0	33	2	2	2
3	16.49626168	24.1791208	0.248786862	-0.497562908	0.31775701	0.4299065	0.214953	14.60777	0.278685	0.481875096	39.4	0.383178	17.7	108	1	4	3
4	18.95677966	23.5460679	0.504717394	-0.539129161	0.19491525	0.3983051	0.372881	13.3888	0.385379	0.442870916	46.3	0.237288	22.6	119	1	4	3
5	10.03134328	22.726647	0.317194069	-0.409705151	0.55970149	0.2313433	0.179104	12.36366	0.395532	0.454820712	30.1	0.358209	0	135	1	4	3
6	0.637931034	2.31221097	0.458104281	-0.152752319	0.72413793	0.0689655	0.206897	1.056152	0.319365	0.154439237	2.8	0.275862	0	30	2	2	2
7	6.076829643	15.0651596	0.666629632	-0.428556123	0.59663866	0.1512605	0.235294	8.193108	0.739887	0.689950543	28.5	0.142857	0	120	2	4	3
8	10.39783784	19.045356	0.3349933	-0.398800029	0.45405405	0.2702703	0.227027	11.38313	0.309088	0.387266279	30.1	0.372973	5.2	186	1	4	3
9	12.88913043	14.0648908	0.366153294	-0.473842273	0.09782609	0.4782609	0.369565	9.43326	0.473266	0.415402732	25.5	0.48913	13.5	93	2	4	3
10	4.880952381	8.91265597	0.455093198	-0.591607506	0.45238095	0.3095238	0.238095	5.0791	0.326764	0.786130698	13.8	0.309524	5	43	2	2	2
11	1.284615385	4.45318489	0.248524931	-0.363219923	0.71153846	0.1826923	0.125	2.286111	0.235605	0.253048848	8	0.105769	0	105	2	2	2
12	6.362222222	8.17976629	0.329150209	-0.470204509	0.22222222	0.44444444	0.311111	4.539373	0.261794	0.362235622	14.5	0.377778	7	46	2	2	2
13	8.542307692	9.45086275	0.359938706	-0.375587535	0.09615385	0.4615385	0.442308	5.395701	0.490543	0.374543797	16.3	0.403846	8.5	53	2	2	2
14	27.5245098	27.52424	0.212296533	-0.295345446	0	0.4117647	0.480392	8.685749	0.131665	0.357698464	37.3	0.705882	30.2	103	1	4	3
15	15.56271186	19.9604356	0.52544107	-0.600500505	0.22033898	0.4067797	0.355932	12.21254	0.399638	0.435285373	32.4	0.440678	16	60	1	4	3
16	1.336666667	5.72775319	0.652614624	-0.652614624	0.76666667	0.13333333	0.133333	2.797318	0.494702	0.273897521	9.4	0.133333	0	31	2	2	2
17	1.716666667	3.168987	0.22681405	-0.194416671	0.45833333	0.25	0.291567	1.677465	0.21718	0.192493647	4.4	0.375	23	25	2	2	2
18	0.170833333	2.04897551	0.638250638	-0.319284802	0.91666667	0.0416667	0.063333	0.571168	0.000638	0.180510504	2.3	0.083333	0	25	2	2	2
19	3.6	5.27982401	0.39019306	-0.910392672	0.31818182	0.47727272	0.204545	8.692068	0.34326	0.703028131	12.4	0.204545	2.5	45	2	2	2
20	0.045454545	1.4980515	0.416250416	-0.416250416	0.96969697	0.030303	0.030303	0.25713	0.000416	0.000416042	1.5	0.030303	0	34	2	2	2
21	0.670833333	2.29967148	0.208281263	-0.416458437	0.70833333	0.1666667	0.083333	1.107354	0.181598	0.305479266	3	0.166667	0	25	2	2	2
22	4.508546793	7.76428813	0.47563499	-0.422792529	0.41935484	0.2580645	0.290323	6.463648	0.268242	0.235971894	13	0.354839	2.8	32	2	2	2
23	6.162711864	11.7286539	0.518483953	-0.518483953	0.47457627	0.2542373	0.254237	8.295637	0.681499	0.627590225	26.4	0.220339	2.4	60	2	2	2
24	6.881081081	14.1436587	0.486070605	-0.833214303	0.51351351	0.3243243	0.189189	8.100078	0.262657	0.664797923	21	0.297297	0	38	2	2	3
25	33.96734007	34.3295757	0.200311578	-0.318284385	0.01346801	0.4781145	0.40404	11.74681	0.192021	0.334735155	47.3	0.787879	37.4	298	1	3	4
26	9.563809524	16.7363877	0.504324518	-0.645529734	0.42857143	0.3047619	0.238095	11.3678	0.502451	0.629129495	39.2	0.12381	7.5	106	1	4	3
27	16.60769231	20.5614152	0.550238348	-0.525228651	0.19230769	0.4038462	0.423077	13.32886	0.775423	0.441020831	36.7	0.365385	16.2	53	1	4	3
28	6.023076923	9.21122287	0.591607506	-0.739490897	0.34615385	0.3846154	0.307692	6.448274	0.8307	0.488245922	16.6	0.423077	3.7	27	2	2	2
29	11.51097805	18.9106995	0.400642895	-0.418061773	0.39130435	0.2981366	0.285714	10.62033	0.445734	0.350837999	29.8	0.329193	13.4	162	1	4	3
30	0.135294118	3.44827586	0.513632073	-1.026751027	0.96078431	0.0392157	0.019608	0.671499	0.374906	0.001026238	3.7	0.039216	0	52	2	2	2
31	0.868	2.41084324	0.430340385	-0.107625436	0.64	0.08	0.32	1.201572	0.291594	0.120624078	3.1	0.28	0	26	2	2	2
32	3.2625	10.439304	0.54357314	-0.54357314	0.6875	0.1458333	0.145833	4.860625	0.977845	0.95492161	12.6	0.3125	0	49	2	2	2
33	7.469060773	11.3604087	0.39963956	-0.365813009	0.34254144	0.2983425	0.325967	7.359553	0.430275	0.285674006	24.5	0.19337	7.3	182	2	2	2
34	5.697139948	21.3272078	0.451366321	-0.531014515	0.73287671	0.1369863	0.116438	10.46554	0.589584	0.425426981	31.9	0.150685	0	147	1	4	3
35	15.3263287	24.5218374	0.452494364	-0.26975977	0.375	0.2279412	0.382353	13.38489	0.497406	0.238807916	34.3	0.477941	17.7384	137	1	4	3
36	24.87537672	26.5336369	0.116031083	-0.278154485	0.0625	0.6076389	0.253472	11.26469	0.167963	0.38348049	35.9	0.715278	30.79933	289	1	4	3
37	0.03255814	1.3986014	0.388500389	-0.388500389	0.97674419	0.0232558	0.023256	0.211001	0.000388	0.000388306	1.4	0.023256	0	44	2	2	2
38	4.548623853	9.01438156	0.513262708	-0.421611927	0.49541284	0.2110092	0.256881	5.454408	0.610926	0.638189302	17.7	0.192661	3	110	2	2	2
39	0.01875	0.19993336	0.041645844	-0.041645844	0.90625	0.0625	0.0625	0.063431	0.013885	0.013885434	0.3	0.0625	0	33	2	2	2
40	18.31258041	20.3986555	0.457322021	-0.600230687	0.10227273	0.4772727	0.363636	12.95596	0.481499	0.927230101	40.1	0.431818	16.5	89	1	4	3
41	14.25294118	16.904258	0.585708201	-0.471045188	0.15686275	0.3529412	0.470588	9.416576	0.912251	0.296631266	28.9	0.431373	15.8	52	1	4	3
42	18.49242138	24.1019764	0.427620326	-0.345737325	0.23275862	0.3275862	0.405172	13.0285	0.304522	0.281968845	40.5	0.431034	22.5	117	1	4	3
43	15.97487437	19.6233357	0.399995	-0.415579018	0.18592965	0.4020101	0.386935	11.56914	0.327334	0.300792035	39.3	0.301508	16.2	200	1	4	3
44	2.930952381	9.46850242	0.423558166	-0.677642249	0.69047619	0.1904762	0.119048	4.499285	0.385815	1.10690957	11.1	0.261905	0	43	2	2	2
45	4.04375	11.4169755	0.552014332	-0.552014332	0.64583333	0.1666667	0.166667	5.985361	0.884203	0.399974406	15.8	0.291667	0	49	2	2	2
46	0.051851852	1.39953349	0.388500389	-0.388500389	0.96296296	0.0123457	0.012346	0.264394	0.000388	0.000388306	1.4	0.037037	0	82	2	2	2
47	11.96923077	18.0403183	0.438259431	-0.563471939	0.33653846	0.3461538	0.269231	11.51743	0.231489	0.577046688	32.4	0.326923	9.5	105	1	4	3
48	11.66583444	18.8445679	0.18620455	-0.264284837	0.38095238	0.3492063	0.246032	10.78892	0.162392	0.480338902	28.6	0.365079	11.7	127	1	4	3

图 20 3、4、4 类聚类结果部分数据截取

6.2.5 长片段抽样

尽管在运动学片段处理中，对部分特别长（超过 1300 秒）的片段进行了删除。但是仍然保留了大量的长片段。对于这部分数据，在保证每个片段被选取的概率一定和抽取数量一定的同时，需要对这部分数据进行缩减。由于离散小波变换压缩模型实现起来比较困难，而且这些数据也不适用于高密度压缩的小波变换。在此采取了最简单的办法，适用等间隔抽样。对于数据长度在 1200 秒以上的，每 4 个点抽取一个点，将数据压缩成 300 秒左右。对于 800~1200 秒的数据，每 3 个点抽取一个点将数据压缩为 250~400 秒左右。对于 400~800 秒的数据，每 2 个点抽取一个点将数据压缩为 200~400 秒左右。

id	1平均速度	2平均行驶速度	3平均加速度	4平均减速度	5怠速时间比	6加速时间比	7减速时间比	8速度标准差	9加速标准差	10减速度标准差	11速度峰值	12大于峰值60%速度占比	13速度中位数	14片段时长
1313	67.726061	69.17154232	0.228668687	-0.209456028	0.02089783	0.4303406	0.469814241	19.040657	0.3057143	0.307971454	92.8	0.812693498	71.899977	1293
1202	80.115923	83.09993076	0.189182695	-0.180985886	0.03590945	0.4309133	0.450429352	21.3654018	0.2371299	0.28686593	108	0.866510539	87.7	1282
483	37.763114	41.64899178	0.358866171	-0.589572487	0.09330144	0.3787879	0.268740032	20.6374111	0.637506	1.215497531	261.4	0.003987241	40.2	1255
1042	79.218455	83.7957226	0.223795112	-0.187595948	0.01444623	0.3908507	0.468699839	19.9819865	0.3100618	0.327211923	101.7	0.876404494	86.5	1247
308	59.685721	60.67221161	0.301120217	-0.346867239	0.01626016	0.4869919	0.422764228	18.0184993	0.2856124	0.388047414	89	0.734146341	63.9	1231
595	44.333001	45.11465308	0.300227894	-0.388253964	0.01732673	0.4727723	0.360561056	14.6109625	0.3129231	0.774527686	68.9	0.683168317	46.5	1213
616	42.082165	42.5453532	0.255023922	-0.33773303	0.01088777	0.5	0.376884422	17.2808858	0.2795311	0.514122378	122.4162	0.008375209	44	1195
2813	33.475289	37.86649842	0.300843177	-0.31828167	0.11596639	0.4201681	0.399159664	19.17758	0.3448958	0.360848372	64.4	0.548739496	40.8	1191
342	39.063414	40.67530104	0.372703186	-0.367696451	0.03962901	0.4359191	0.457841484	20.719386	0.6148025	0.692811613	83.8	0.301011804	39	1187
2169	65.191565	70.48534173	0.248597461	-0.168632078	0.07510549	0.3434599	0.506329114	26.3628771	0.2649738	0.2909299	91.51795	0.737552743	76.2	1186
964	85.084078	88.5816467	0.142186682	-0.211511359	0.03948498	0.5133047	0.345064378	23.413029	0.2124825	0.309450478	110.3	0.867811159	94.9	1166
1768	87.183258	87.86846415	0.15707225	-0.194755418	0.00779896	0.4792028	0.386481802	15.1363518	0.2303748	0.323300696	111.5	0.948873484	90	1155
307	56.70911	58.07371561	0.268578688	-0.340651463	0.02349869	0.5265448	0.415143603	18.6736709	0.3789835	0.476677523	90.5	0.695387293	60.989383	1150
363	74.779687	77.27902895	0.143445823	-0.164152394	0.03234266	0.4781469	0.417832168	21.7590583	0.2099632	0.214101181	103.7	0.811188811	80.06374	1145
967	64.163785	65.02383075	0.202572279	-0.256748461	0.01322751	0.4805996	0.379188713	18.5015421	0.2277055	0.3170328	105.7	0.422398589	60.7	1135
580	82.469921	82.91244008	0.17881548	-0.12556426	0.00533808	0.3692171	0.525800712	17.5176611	0.225325	0.2099310549	103.9	0.902135231	86.1	1125
1671	41.173791	42.00098814	0.269187322	-0.261826746	0.01969561	0.4458371	0.458370636	15.281321	0.312181	0.361993821	63.5	0.63831692	44.3	1118
2165	83.158698	84.15133291	0.18945398	-0.153974481	0.01179673	0.4056261	0.499092559	20.1897494	0.2648775	0.247535971	109.9	0.872958258	87.7	1103
297	59.305232	60.87468951	0.315240963	-0.31087993	0.02578269	0.4594843	0.465930018	21.2493668	0.3164275	0.515719572	105.2	0.391344383	56.6	1087
2135	40.06402	40.250673	0.622820214	-0.774038419	0.00463822	0.4499072	0.441558442	10.9256143	1.0221837	1.456998363	61.8	0.728200371	41.948736	1079
2221	48.152543	51.46999164	0.200485139	-0.248460051	0.06445498	0.4663507	0.376303318	15.4005205	0.2013031	0.361007789	68.3	0.83507109	52	1056
2689	25.120062	28.01018955	0.154352541	-0.325481913	0.10318226	0.5612343	0.266152363	13.2184045	0.1853007	0.327673609	53.8	0.280617165	27	1038
1866	42.019871	42.43138592	0.197614619	-0.156154378	0.00969932	0.3908826	0.494665373	13.5005782	0.2230679	0.246457055	60.6	0.622696411	45.9	1032
1620	42.936319	43.78903298	0.259400396	-0.336641017	0.0194742	0.5092502	0.392405063	13.8250093	0.2152104	0.412638052	65.7	0.714703019	46.5	1028
487	55.77929	58.10573257	0.235833466	-0.274294815	0.04003906	0.4306641	0.389648438	23.6283272	0.4441646	0.6051309	84.6	0.631835938	67	1025
330	54.200529	57.71189308	0.23527323	-0.145857962	0.06084396	0.3395486	0.547595682	24.4566511	0.2552976	0.238209511	99	0.42983317	52.742952	1020
1462	51.192407	53.9797908	0.169718797	-0.189269689	0.05163853	0.4518371	0.405163853	22.0369204	0.2303167	0.299645009	87.4	0.347567031	45.244828	1008
2841	55.977926	56.14614012	0.231093194	-0.322071202	0.002997	0.5084915	0.36963037	22.3882587	0.2877386	0.493540323	84	0.612387612	66.1	1002
2345	56.72623	60.58550897	0.292424823	-0.22251051	0.06370071	0.3700708	0.486349848	19.9806012	0.3292322	0.368472219	79.7	0.864509606	60.651252	990
1103	44.125185	46.54750749	0.276853122	-0.385481844	0.05204082	0.5214286	0.374489796	19.1503059	0.2823883	0.428194166	79.1	0.423469388	44.2	981
2312	37.158611	40.10964136	0.298226345	-0.413469814	0.07357513	0.4870466	0.351295337	18.4033741	0.2812062	0.560045638	65.8	0.558549223	40.9	966
822	43.450323	47.94865289	0.317536879	-0.281686024	0.09381663	0.4104478	0.462686567	20.8657527	0.3011777	0.31710175	75.9	0.517057569	46.063882	939
1021	85.318711	85.9635757	0.155766138	-0.203173086	0.00750268	0.4823151	0.36977492	17.952497	0.2619207	0.337671762	104	0.91318328	91	934
759	47.736306	49.49832888	0.240408851	-0.321673585	0.03559871	0.5124056	0.382955771	20.9972667	0.2409764	0.40710267	81.3	0.513484358	51.7	928
2313	58.866397	60.03328596	0.289638657	-0.334056813	0.01943844	0.4848812	0.421166307	17.78451	0.38206	0.468740879	90	0.750539957	62.2	927
1055	49.122308	50.82750438	0.304864178	-0.241628296	0.03354978	0.3928571	0.495670996	19.7445631	0.3388494	0.316161867	76.9	0.626623377	52.8	925
344	37.682725	42.32848732	0.418437676	-0.436590776	0.1097561	0.3758315	0.416851441	16.7617804	0.6416603	0.920459265	74	0.375831486	41.5	903
1864	32.107103	35.16487851	0.24119796	-0.375290063	0.08695652	0.5091533	0.327231121	13.7817942	0.2816434	0.436089712	55.3	0.550343249	34.5	875
231	46.250121	49.99384186	0.264809971	-0.366208405	0.07488479	0.5034562	0.3640553	23.9102632	0.2790548	0.424962575	85.3	0.498847926	51.1	869
1156	37.836648	43.26356208	0.259312821	-0.364439285	0.12543962	0.4876905	0.347010551	19.1511546	0.320985	0.405901706	66.5	0.631887456	44.898135	854
744	45.366037	46.35819073	0.32075396	-0.402365673	0.02140309	0.5041617	0.401902497	18.1901673	0.2931491	0.409086789	78	0.474435196	45.2	842
1924	39.426962	39.90193728	0.281407021	-0.06450961	0.01190476	0.1821429	0.794047619	10.7409989	0.2836482	0.238626392	52.4	0.805953192	43.789209	841
2187	47.776317	53.61053004	0.3174055304	-0.373179496	0.10882709	0.4448199	0.378476421	25.8030642	0.3065976	0.45478692	84.3	0.483675937	49.1	828
1139	32.511119	33.15447984	0.212248326	-0.728186591	0.00613497	0.5263804	0.175460123	15.0662168	0.5087883	1.55034248	62.7	0.392638037	28.4	816
1453	33.377653	34.83439939	0.205288654	-0.210104796	0.04182042	0.4292743	0.419434194	11.9562499	0.2380882	0.302387723	51.6	0.704797048	36.7	814
1229	40.786006	52.40219113	0.285772967	-0.145472892	0.22167488	0.2438424	0.498768473	28.4476227	0.2706327	0.250308783	93.6	0.331280788	45	813
953	34.497895	37.56957342	0.37696831	-0.595125425	0.08176101	0.5018868	0.324528302	17.7060946	0.5360509	0.818681415	71.5	0.374842767	37.6	796
229	16.836157	21.81232512	0.058984699	-0.542960266	0.22813688	0.6869455	0.081115336	13.9946118	0.1458091	0.431920127	62.7	0.086185044	14.698037	790

图 21 部分长片段数据截取

对长片段进行处理后，基本上数据长度集中在 200~500 秒之间。

6.2.6 基于遗传算法的工况曲线合成

通过主成分分析，从 14 个特征中提取出 7 个主要特征，这 7 个特征对主体的贡献率能达到 95%，因此由这 7 个特征进行聚类具有高度的代表性。再由等间隔抽样对较长的数据片段进行压缩。得到一系列长度较为均匀的运动学片段。

在 6.2.4 小结的聚类分析中，对聚类数量做了一个简单的讨论。对分成 3 类、4 类、5 类分别进行了实验，最后得出结论，分成 4 类的效果较好，因此在之后的文中对于聚类类别就不再论述，默认为 4 类。最终的工况曲线的合成就是从四类中抽取合适的样本。考虑到每类样本的样本容量不同，我们分别设置每类样本的抽取比例为 1:1:1:1、1:2:1:2、1:3:2:3。

在 4 类聚类中抽取特点数量的样本，而样本具有明确的优化函数：归一化加权累积均方误差。通过对主成分分析得到的 7 个特征：平均行驶速度 (km/h)、平均加速度 (m/s²)、平均减速度 (m/s²)、怠速时间比 (%)、加速时间比 (%)、速度标准差 (km/h)、大于峰值 60%速度占比 (%)。计算归一化均方误差，然后依据主成分分析中得到的贡献率 (L) 对均方误差加权并累加得到最终的累积均方误差。

具体计算公式如下：

$$\begin{aligned}
f(x) = & \\
& l_1 \times \left(\frac{v_{seqz} - v_{seqz}(x)}{v_{seqz}} \right)^2 + l_2 \times \left(\frac{a_{useq} - a_{useq}(x)}{a_{useq}} \right)^2 + \\
& l_3 \times \left(\frac{a_{dseq} - a_{dseq}(x)}{a_{dseq}} \right)^2 + l_4 \times \left(\frac{P_{idle} - P_{idle}(x)}{P_{idle}} \right)^2 + \\
& l_5 \times \left(\frac{P_{au} - P_{au}(x)}{P_{au}} \right)^2 + l_6 \times \left(\frac{v_{csd} - v_{csd}(x)}{v_{csd}} \right)^2 + \\
& l_7 \times \left(\frac{P_{vmax} - P_{vmax}(x)}{P_{vmax}} \right)^2
\end{aligned} \tag{30}$$

其中 $f(x)$ 中 x 代表合成的工况曲线， x 的计算公式有三种，分别对应三种取值比例。

$$\begin{aligned}
x = & \\
& pick(pool_1) + pick(pool_2) + pick(pool_3) + pick(pool_4)
\end{aligned} \tag{31}$$

$$\begin{aligned}
x = & \\
& pick(pool_1) + pick(pool_2) + pick(pool_2) + pick(pool_3) \\
& + pick(pool_4) + pick(pool_4)
\end{aligned} \tag{32}$$

$$\begin{aligned}
x = & \\
& pick(pool_1) + pick(pool_2) + pick(pool_2) + pick(pool_2) \\
& + pick(pool_3) + pick(pool_3) + pick(pool_4) \\
& + pick(pool_4) + pick(pool_4)
\end{aligned} \tag{33}$$

得出优化函数后，就可以使用元启发算法，在此选择遗传算法来实现。

遗传算法 (Genetic Algorithm) ^[11] 是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。遗传算法是从代表问题可能潜在的解集的一个种群 (population) 开始的，而一个种群则由经过基因 (gene) 编码的一定数目的个体 (individual) 组成。每个个体实际上是染色体 (chromosome) 带有特征的实体。染色体作为遗传物质的主要载体，即多个基因的集合，其内部表现 (即基因型) 是某种基因组合，它决定了个体的形状的外部表现，如黑头发的特征是由染色体中控制这一特征的某种基因组合决定的。按照适者生存和优胜劣汰的原理，逐代 (generation) 演化产生出越来越好的近似解，在每一代，根据问题域中个体的适应度 (fitness) 大小选择 (selection) 个体，并借助于自然遗传学的遗传算子 (genetic operators) 进行组合交叉 (crossover) 和变异 (mutation)，产生出代表新的解集的种群。这个过程将导致种群像自然进化一样的后生代种群比前代更加适应于环境，末代种群中的最优个体经过解码 (decoding)，可以作为问题近似最优解。

遗传算法流程图如下：

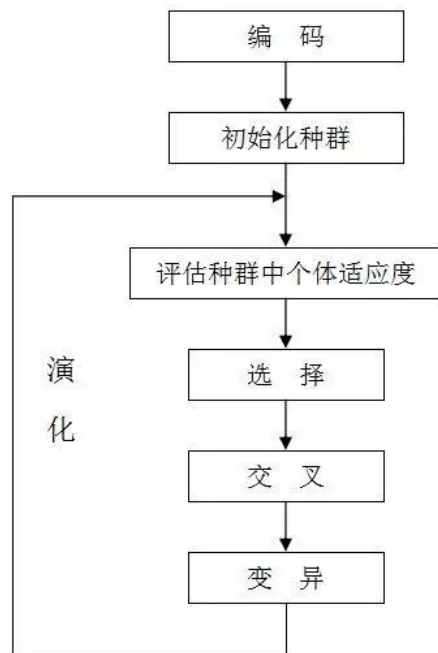


图 22 遗传算法框图

根据如上原理图，首先随机从聚类中选取几个样本进行曲线合成、优化函数计算。保留优化函数计算值，然后使用轮盘赌法，交叉替换一部分样本，再次计算优化函数值。若此次优化函数值小于保留的优化函数值，则将此次优化函数值保留，再对样本进行轮盘赌法交叉替换样本。再计算优化函数值，依此迭代 100 次。

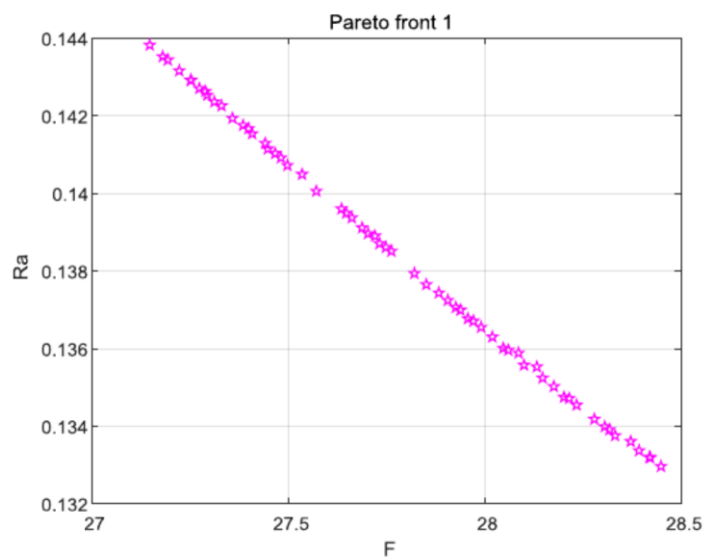


图 23 优化目标函数（部分）

经过三种比例（1:1:1:1、1:2:1:2、1:3:2:3）的分别计算，通过遗传算法得出了近似最优解。最终优化函数：归一化加权累积均方误差值为 0.073。而对于题目要求的 9 组数据的误差统计分别为：

表 5 各特征误差统计

数据类型	误差 (%)	数据类型	误差 (%)
平均速度 (km/h)	1.517	加速时间比 (%)	0
平均行驶速度 (km/h)	1.480	减速时间比 (%)	5.294
平均加速度 (m/s^2)	15.238	速度标准差 (km/h)	0.714
平均减速度 (m/s^2)	12.912	加速度标准差等 (m/s^2)	3.5
怠速时间比 (%)	0.435	—	—

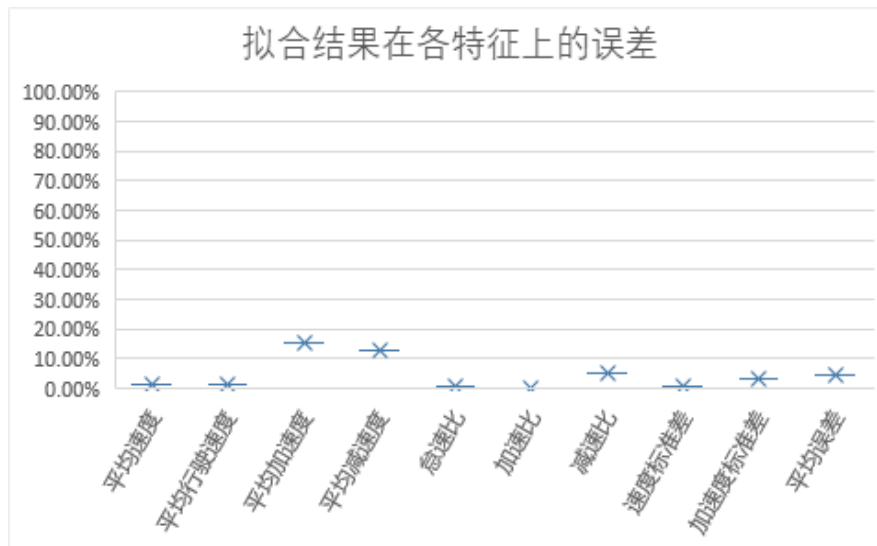


图 24 各特征误差大小

6.3 汽车运动特征的评估体系

经过 6.2 节的提取，最终得到了较好的汽车行驶工况曲线。

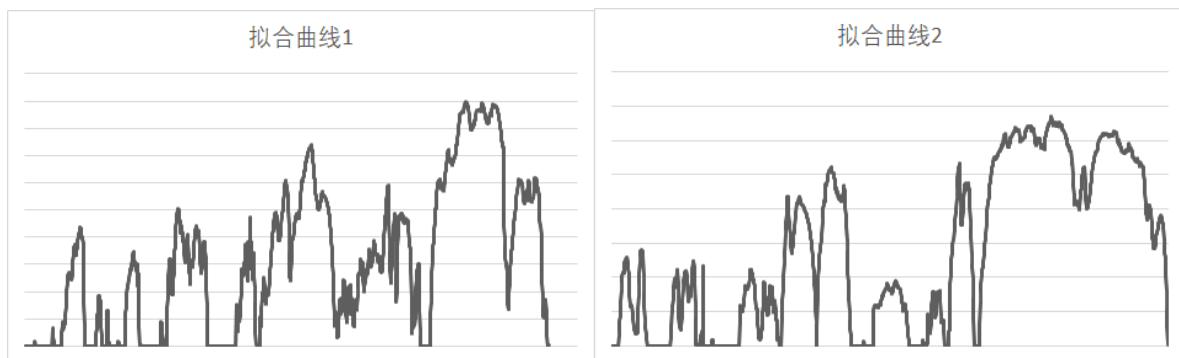


图 25 构建的较好的汽车工况曲线

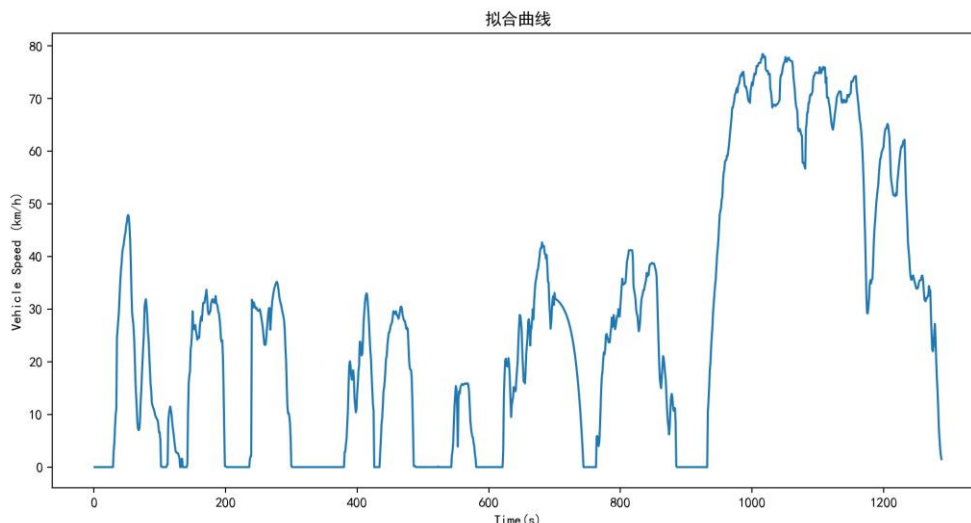


图 26 最终合成的汽车工况曲线

经过误差比对分析，选择了拟合曲线 1 作为最终合成结果。

6.3.1 归一化加权累积均值误差

归一化加权累积均方误差作为遗传算法的优化函数，事实上也纳入到汽车工况曲线的评估体系。通过对主成分分析得到的 7 个特征：平均行驶速度 (km/h)、平均加速度 (m/s^2)、平均减速度 (m/s^2)、怠速时间比 (%)、加速时间比 (%)、速度标准差 (km/h)、大于峰值 60% 速度占比 (%)。计算归一化均方误差，然后依据主成分分析中得到的贡献率对均方误差加权并累加得到最终的累积均方误差。这个参数能够对以这七个特征为标准，判断与原始数据中这 7 个特征的综合误差。由于这 7 个特征在主成分分析里占总体贡献率的 95% 以上。因此，有理由相信该评价标准的有效性。

经过多次遗传算法的迭代，归一化加权累积均方误差的值已经下降到 0.073。因此在归一化加权累积均方误差这一评估体系下，可以认为汽车工况曲线基本能代表原始数据中的真实行车情况。

6.3.2 特征误差与误差均值

对于题目中给出的 9 个特征：平均速度 (km/h)、平均行驶速度 (km/h)、平均加速度 (m/s^2)、平均减速度 (m/s^2)、怠速时间比 (%)、加速时间比 (%)、减速时间比 (%)、速度标准差 (km/h)、加速度标准差等 (m/s^2)。分别计算其误差，然后取平均得到均值误差。其中 9 个特征的误差在表 5 中已经详细给出，也作图进行了直观的判断。可以看到，误差最小的为加速度时间比，为 0%。误差最大的为平均加速度，为 15.24%。而得出的误差均值为：4.56%。

因此，在特征误差和误差均值这一评估体系下，可以认为汽车工况曲线基本能代表原始数据中的真实行车情况。

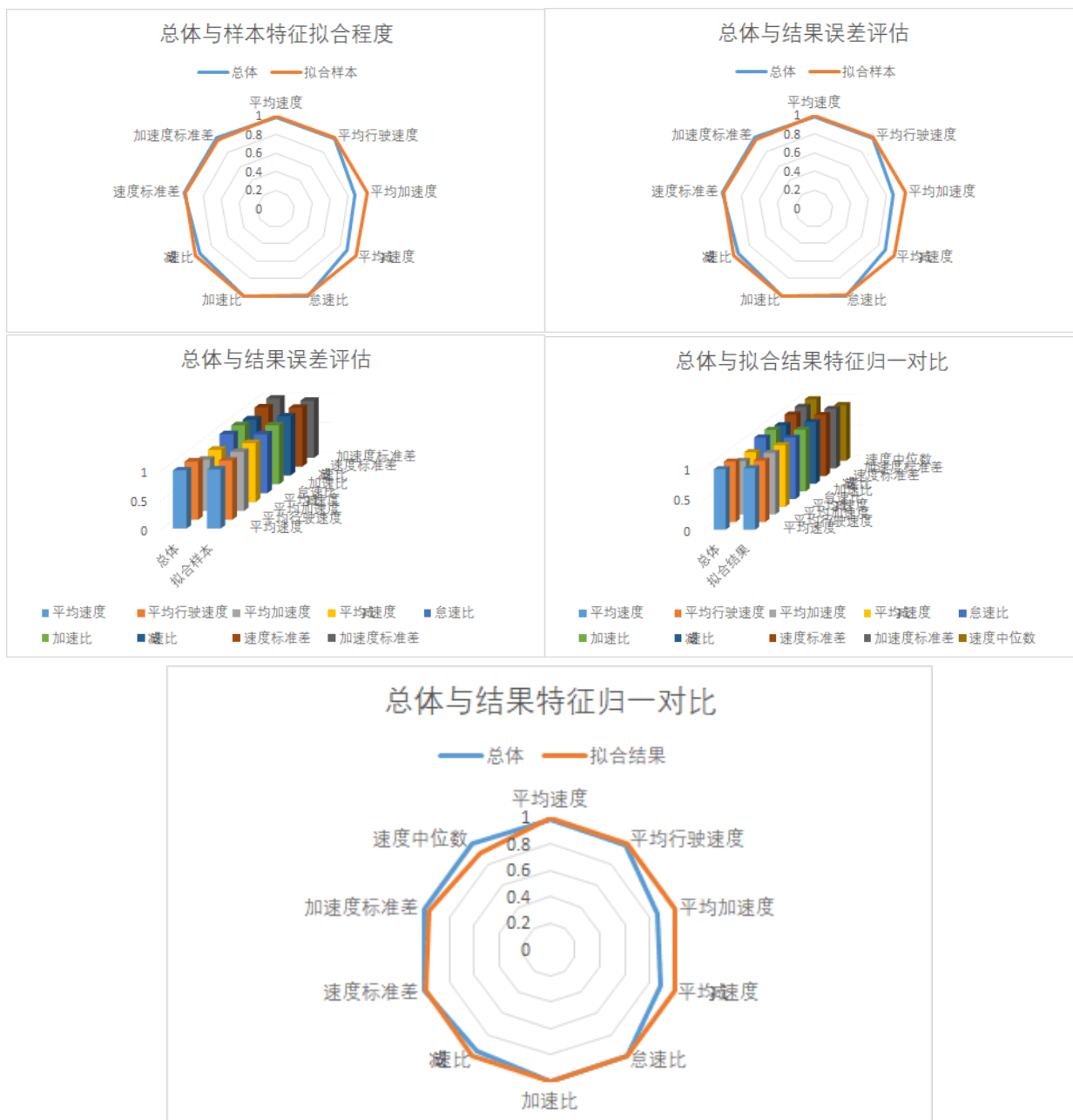


图 27 总体特征分析

6.4 汽车行驶工况的构建结果分析

6.4.1 工况曲线的建立小结

根据本题要求，在第一题中较为严谨的数据预处理以及第二题对运动学片段的分离和选取是基础上，对运动学片段进行合成工程曲线。原题目中给出了 9 种特征，依据特征间的关系以及统计分析，又建立了新的 5 中特征，分别为：速度标准差 (m/s^2)、速度峰值 (m/s)、速度中位数 (m/s)、大于峰值 60%速度占比 (%)，运动学片段的时长。

然后对离散小波压缩模型对于工程曲线的建立，以及主成分分析加 K 均值聚类分析模型对于工程曲线的建立，之间的利弊进行了分析，考虑到本题并非直接使用原始数据对汽车行驶工况曲线进行构建。而是使用预处理和分割好的运动学片段对工况曲线进行构建。

于是最终选择了主成分分析加 K 均值聚类分析模型。

使用主成分分析，对 14 个特征

通过对主成分分析产生的 7 个主要特征进行聚类，并对聚类数目进行了论证。最终确定，将数据分为四类，每类运动学片段分别为：148、1048、658、1073。对于数据片段过长（大于 1300 秒）和数据过短（小于 20 秒）的部分进行了剔除。对于数据长度在 1200 秒以上的，每 4 个点抽取一个点，将数据压缩成 300 秒左右。对于 800~1200 秒的数据，每 3 个点抽取一个点将数据压缩为 250~400 秒左右。对于 400~800 秒的数据，每 2 个点抽取一个点将数据压缩为 200~400 秒左右。

然后分别设置每类样本的抽取比例为 1:1:1:1、1:2:1:2、1:3:2:3。建立优化函数：归一化加权累积均方误差（6.2.6 节）。通过遗传算法进行迭代寻找最优结果。

6.4.2 评价模型小结

汽车工程曲线的建立与寻优是依据主成分分析中提取的 7 个特征来进行的。寻优过程中使用的优化函数：归一化加权累积均方误差。也是一种评价模型。在遗传算法迭代寻优过程中就取得了较好的结果，得到的近似最优值为 0.073。可以认为构建的汽车工程曲线 7 个典型特征都是基本符合原数据中的特征的。

而对于题目要求的九组数据其误差大小分别为：平均速度（1.52%）、平均行驶速度（1.48%）、平均加速度（15.24%）、平均减速度（12.91%）、怠速时间比（0.43%）、加速时间比（0%）、减速时间比（5.29%）、速度标准差（0.71%）、加速度标准差等（3.5%）。可以看到，误差最小的为加速度时间比，为 0%。误差最大的为平均加速度，为 15.24%。9 组特征误差均值为：4.53%。可见构建的汽车工程曲线与原数据各特征之间的误差非常小。

6.4.3 评价模型与工况曲线评估

在上述实验中，不仅设计了主成分分析加 K 均值聚类算法对汽车运动学片段进行处理和分类，并结合遗传算法对运动学片段进行选取。完成题目要求的对汽车工况曲线的合理构建。还对基础的汽车评估体系（9 组特征）进行了计算并建立文件存储，同时根据已有的信息对特征进行扩充，得到一共 14 组特征。

得到的汽车工况曲线完整，且归一化加权累积均方误差低于 0.1。各特征误差均值不到 5%。比较成功地对曲线进行了构建与评价。不过限于时间原因，模型有些地方还有待改进，现将其列出：

1) 考虑到构建工况曲线抽样数量不能太少，对于 400 秒以上的数据进行了压缩。压缩过程直接使用了等间隔抽样。这种抽样方法虽然简单，但是在数据变化较大的时候，会加剧数据的变化率。比如加速度原来可能是 2m/s，但是抽样后可能就会变成 4m/s。所以等间隔抽样有比较大的风险参数不良数据。

2) 评价模型只有两，分别为：归一化加权累积均方误差、各特征误差比值以及误差均值。如果时间充裕的话可以将小波变换模型实现，然后计算与小波变换模型之间的误差。小波变换模型是直接原始数据进行压缩，与之相比由较大的参考性。

七、总结与展望

7.1 实验结论

对于问题一提出的数据预处理要求，设计了三个步骤：数据插值与预分割、数据扩充、异常处理和数据修正。首先通过分割，将三个文件一共分割为成 212 段数据片段，然后进行指数插值和线性插值，一共插入 45624 行数据。在此解决了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的问题。通过数据扩充，补充了加速度、位移、瞬时油耗、停车判别 4 条新信息。因此异常处理和数据修正就可以依据扩充的数据和原始数据进行综合判断。通过加速度信息，剔除了汽车加、减速度异常的数据 476 行；通过停车判别和 GPS 车速，剔除了长期停车所采集的异常数据 187 段；通过连续的低速判别，将连续的低速片段 GPS 车速置为 0m/s，一共处理了 148 段；通过遍历数据，寻找 180 秒以上的片段，删去了怠速超过 180 秒的数据 300 段。对于问题一提出的要求，非常成功的解决了。

对于问题二提出的运动学片段的提取，依据统计分析，使用连续的 5 个怠速运行时间点作为运动学片段的开始和结尾标志。遍历数据，判断出怠速区间，然后对两个怠速区间内的片段进行分离，得到独立的运动学片段并进行保存。再剔除无效的片段后一共得到 2950 条运动学片段。数据长度分布非常离散，考虑到实际意义以及方便后续处理，又删除了 550 条长度大于 20 秒或者长度大于 1300 秒的运动学片段。

对于问题三提出的构建汽车行驶工况曲线并建立合理的评估体系要求，分了 6 个阶段进行分别处理。首先对特征进行扩展和计算，每个原始运动学片段都有 9 个特征，通过分析，填充了 5 条新的特征，分别为：减速度标准差 (m/s^2)、速度峰值 (m/s)、速度中位数 (m/s)、大于峰值 60%速度占比 (%)，运动学片段的时长。然后依据表 3 中的公式对特征进行运算存储。其次对小波变换模型是否适用于本题进行了论述，考虑到离散小波变换模型适用范围为对原始数据进行直接压缩，与本题不符，因此在本题中不使用小波变换模型。然后提出新的主成分分析法和 K-means 聚类法组合求解模型，分别对这两种方法进行了介绍并展示了中间过程。对于主成分分析，取了 7 个特征，累积贡献率为超过 95%。对于聚类分析，经过论述，产生了 4 个类别。接着使用遗传算法模型，对目标函数：归一化加权累积均方误差进行迭代优化，提取出了优良的，具有代表性的运动学片段。最后构建出工况曲线。按照题目要求，设计了两种评估体系，分别为归一化加权累积均方误差和特征误差以及误差均值。论述了两种评估体系的合理性。并求出归一化加权累积均方误差的值为 0.073。特征误差分别为：平均速度 (1.52%)、平均行驶速度 (1.48%)、平均加速度 (15.24%)、平均减速度 (12.91%)、怠速时间比 (0.43%)、加速时间比 (0%)、减速时间比 (5.29%)、速度标准差 (0.71%)、加速度标准差等 (3.5%)。误差均值为：4.53%。

就实验结果来看，比较成功的完成了三个问题的求解，而且数据预分割和归一化加权累积均方误差的提出，完善了题目的要求，也提高了实验精度。

7.2 模型评价

第一题中设计的三个步骤：数据插值与预分割、数据扩充、异常处理和数据修正。能够很好的对题目中提出的五种不良数据进行处理，同时也能对数据填充和数据插值可能造成的不良数据进行处理和修正。该模型考虑到了各种数据不良情况并进行解决，对后续操

作也进行了数据扩充和分割。总的来说，还是比较成功的。

第二题中设置连续的 5 个怠速运行时间点作为运动学片段的开始和结尾标志。有其合理性，也有其局限性。合理性在于，考虑到了偶然停车或者数据未很好修正的情况。其局限性在于 5 个怠速运行时间点的设置仅仅是基于统计分析得出的结论。

第三题中综合考虑了多种情况，最终选取的模型为主成分分析加 K-means 聚类模型，对聚类的论述，选择了 4 类分类，为每类模型取值做了三种比例，对于运动片段的选取，使用了遗传算法。最后进行工程曲线的合成。该题中每个模型都经过了比较严禁的论述和选取，可以说这一部分模型也是比较成功的。

总体而言，模型的设计中大部分的选择与参数设置都是有依据的。而限于时间原因，小部分参数仅仅是通过统计分析进行选取。不过就最后而言，模型还是比较精准的。能很好的反应实际行车状况，也具有一定的通用性。

7.3 模型改进

有与时间有限，模型有小部分参数的选择没有严格论证，而有些部分还需要进行改进的。现将其列举出来：

1) 第一题中，对 GPS 信号丢失片段小于 180 秒时进行插值。对于速度，采用的是指数插值模型，对于其它信息采用的是线性插值模型。速度用指数插值模型事实上也基本满足实际情况，但是如果采用 5 次多项式插值，得到的插值曲线可能会更加平滑。至于其它信息，可能也有更好的插值办法。

2) 第一题中，对于长时间低速行驶视为怠速情况的判别时间 T_{lsp} ，阈值设置为 20 秒。对于长时间停车视为异常数据的判别时间 T_{stop} ，阈值设置为 30 秒。这两个参数的设置都是基于统计分析得出的。因此可能存在更好的参数。

3) 第二题中，将 5 个连续的怠速片段作为运动学片段的开始和结束标志。这个参数的得出也是基于统计分析得来的，不够严谨。

4) 第二题中，对于运动学片段的删除，小于 20 秒以下的片段删除基本上影响不大。但是大于 1300 秒的 22 条数据直接删除未免有点可惜了，若时间宽裕可以使用离散小波变换模型来对这部分的数据进行压缩。

5) 第三题中，考虑到构建工况曲线抽样数量不能太少，对于 400 秒以上的数据进行了压缩。压缩过程直接使用了等间隔抽样。这种抽样方法虽然简单，但是在数据变化较大的时候，会加剧数据的变化率。比如加速度原来可能是 2m/s，但是抽样后可能就会变成 4m/s。所以等间隔抽样有比较大的风险参数不良数据。

6) 评价模型只有两种分别为：归一化加权累积均方误差、各特征误差比值以及误差均值。如果时间充裕的话可以将小波变换模型实现，然后计算与小波变换模型之间的误差。小波变换模型是直接原始数据进行压缩，与之相比由较大的参考性。

参考文献

- [1] 姜平, 石琴, 陈无畏, 黄志鹏. 基于小波分析的城市道路行驶工况构建的研究[J]. 汽车工程, 2011(1):70-73.
- [2] Lin J, Niemeier D A. Exploratory analysis comparing a stochastic driving cycle to California's regulatory cycle[J]. Atmospheric Environment, 2002, 36(38):5759-5770.
- [3] Karande, S., Olson, M., and Saha, B. Development of Representative Vehicle Drive Cycles for Hybrid Applications[J]. SAE Technical Paper 2014-01-1900, 2014, doi:10.4271/2014-01-1900.
- [4] Knez M, Muneer T, Jereb B, et al. The estimation of a driving cycle for Celje and a comparison to other European cities[J]. Sustainable Cities and Society, 2014, 11:56-60.
- [5] - 柚子皮 -. 数值分析：数据插值方法 . <https://blog.csdn.net/pipisorry/article/details/62227459>. 2019.9.20
- [6] 百度百科 . 半正矢 . <https://baike.baidu.com/item/%E5%8D%8A%E6%AD%A3%E7%9F%A2/6510709?fr=aladdin>. 2019.9.21
- [7] IT 屋. 计算两个纬度, 经度点之间的距离? . <http://www.it1352.com/61856.html>. 2019.9.21
- [8] 石琴, 郑与波, 姜平. 基于运动学片段的城市道路行驶工况的研究[J]. 汽车工程, 2011, 33(3):256-261.
- [9] 芮芮杰 . 主成分分析 (PCA) 原理详解 . <https://blog.csdn.net/lrt366/article/details/82964326>. 2019.9.21
- [10] 11h_1178. 深入浅出聚类算法值 K-means 算法 . https://blog.csdn.net/11h_1178/article/details/81633396. 2019.9.21
- [11] jingsupo. 一文读懂遗传算法工作原理 (附 Python 实现) . <https://www.cnblogs.com/jingsupo/archive/2018/04/23/genetic-algorithm-python.html>. 2019.9.22

附录

最终构建工况曲线数据（截取部分）

2017/11/06 09:49:51.0	1	0	0	-0.432	-0.882	119.2438	26.04511	0	650	18	53.49	0	0.1471	24	2.12	1	0
2017/11/06 09:49:52.0	1	0	0	-0.432	-0.882	119.2438	26.04511	0	637	19	53.22	0	0.147	24	2.11	1	0
2017/11/06 09:49:53.0	1	0	0	-0.468	-0.882	119.2438	26.04511	0	637	19	54.49	0	0.147	24	2.16	1	0
2017/11/06 09:49:54.0	1	0	0	-0.45	-0.864	119.2438	26.04511	0	650	19	57.27	0	0.147	24	2.27	1	0
2017/11/06 09:49:55.0	1	0	0	-0.45	-0.846	119.2438	26.04511	0	650	19	53.73	0	0.147	25	2.13	1	0
2017/11/06 09:49:56.0	1	0	0	-0.432	-0.864	119.2438	26.04511	0	662	18	51.22	0	0.1469	24	2.03	1	0
2017/11/06 09:49:57.0	1	0	0	-0.432	-0.882	119.2438	26.04511	0	650	19	57.27	0	0.1469	25	2.27	1	0
2017/11/06 09:49:58.0	1	0	0	-0.432	-0.882	119.2438	26.04511	0	637	19	49.95	0	0.1469	25	1.98	1	0
2017/11/06 09:49:59.0	1	0	0	-0.432	-0.864	119.2438	26.04511	0	637	19	50.71	0.015	0.1469	25	2.01	1	0
2017/11/06 09:50:00.0	1	0	0	-0.342	-0.936	119.2438	26.04511	0	1375	36	0.73	0.125	0.1469	43	7.66	1	0
2017/11/06 09:50:01.0	1	0	0	-0.324	-0.936	119.2438	26.04511	0	1512	37	0.87	0.115	0.1468	45	8.88	1	0
2017/11/06 09:50:02.0	1	0	0.09	-0.414	-0.882	119.2438	26.04511	0	1612	32	0.8	0.105	0.1466	38	8.2	1	0
2017/11/06 09:50:03.0	1	0	0.126	-0.414	-0.9	119.2438	26.04511	0	1500	37	0.9	0.115	0.1463	43	9.26	1	0.861111
2017/11/06 09:50:04.0	1	3.1	0.144	-0.396	-0.918	119.2439	26.04516	5.067201	1550	37	0.9	0.12	0.1466	45	9.21	0	0.388889
2017/11/06 09:50:05.0	1	4.5	0.198	-0.432	-0.9	119.2438	26.04516	0.817563	1587	37	0.93	0.115	0.1468	45	9.51	0	0.833333
2017/11/06 09:50:06.0	1	7.5	0.18	-0.378	-0.882	119.2438	26.04518	4.602072	1625	37	0.95	0.12	0.1467	44	9.81	0	0.666667
2017/11/06 09:50:07.0	1	9.9	0.234	-0.414	-0.774	119.2438	26.04521	3.902436	1687	37	0.98	0.12	0.1466	44	10.04	0	0.305556
2017/11/06 09:50:08.0	1	11	0.198	-0.468	-0.864	119.2438	26.04524	4.323726	1775	40	1.11	0.125	0.1453	46	11.37	0	3.777778
2017/11/06 09:50:09.0	1	24.6	0.09	-0.432	-0.882	119.2437	26.04536	13.05615	1925	46	1.38	0.145	0.1457	53	14.09	0	0.444444
2017/11/06 09:50:10.0	1	26.2	0.054	-0.378	-0.918	119.2438	26.04543	8.075454	2100	49	1.61	0.15	0.1456	56	16.44	0	0.472222
2017/11/06 09:50:11.0	1	27.9	0.054	-0.396	-0.756	119.2438	26.0455	8.174537	2237	47	1.7	0.15	0.1452	53	17.46	0	0.472222
2017/11/06 09:50:12.0	1	29.6	0	-0.414	-0.9	119.2438	26.04558	9.139375	2337	44	1.61	0.145	0.1456	48	16.45	0	0.75
2017/11/06 09:50:13.0	1	32.3	0	-0.45	-0.954	119.2439	26.04567	10.46005	1812	40	1.11	0.125	0.1468	48	11.39	0	0.805556
2017/11/06 09:50:14.0	1	35.2	0	-0.45	-0.954	119.2439	26.04576	11.48036	1675	32	0.85	0.105	0.1474	39	8.81	0	0.583333
2017/11/06 09:50:15.0	1	37.3	-0.072	-0.414	-0.9	119.244	26.04586	12.00741	1487	39	0.93	0.125	0.1468	44	9.66	0	0.361111
2017/11/06 09:50:16.0	1	38.6	-0.072	-0.486	-0.9	119.244	26.04596	12.10943	1587	37	0.88	0.115	0.1463	46	9.15	0	0.555556
2017/11/06 09:50:17.0	1	40.6	0	-0.432	-0.864	119.2441	26.04608	14.31899	1412	27	0.59	0.085	0.1463	34	6.12	0	0.277778
2017/11/06 09:50:18.0	1	41.6	-0.09	-0.432	-0.846	119.2441	26.04619	13.66184	1550	39	0.93	0.115	0.1462	46	9.56	0	0.138889
2017/11/06 09:50:19.0	1	42.1	0	-0.378	-0.9	119.2442	26.0463	13.34913	1575	46	1.21	0.145	0.1462	50	12.41	0	0.277778
2017/11/06 09:50:20.0	1	43.1	0	-0.45	-0.9	119.2442	26.04642	14.03896	1762	46	1.23	0.135	0.1464	55	12.58	0	0.25
2017/11/06 09:50:21.0	1	44	0.054	-0.414	-0.828	119.2443	26.04653	13.24825	1750	38	0.87	0.105	0.1464	54	8.86	0	0.166667
2017/11/06 09:50:22.0	1	44.6	-0.072	-0.486	-0.846	119.2443	26.04663	12.90377	1587	37	0.92	0.11	0.1459	45	9.43	0	0.361111
2017/11/06 09:50:23.0	1	45.9	0	-0.432	-0.864	119.2444	26.04675	14.71091	1550	40	1.06	0.14	0.1451	42	10.93	0	0.166667
2017/11/06 09:50:24.0	1	46.5	0	-0.432	-0.828	119.2445	26.04686	13.32248	1762	43	1.11	0.13	0.1457	52	11.45	0	0.305556
2017/11/06 09:50:25.0	1	47.6	0	-0.432	-0.864	119.2445	26.04697	14.67161	1675	24	0.32	0.015	0.1464	43	3.4	0	0.083333
2017/11/06 09:50:26.0	1	47.9	0	-0.504	-0.936	119.2446	26.04708	14.26268	1300	14	0.31	0	0.1454	17	3.19	0	-0.11111
2017/11/06 09:50:27.0	1	47.5	0	-0.54	-0.828	119.2447	26.04718	12.97113	1175	14	0.26	0	0.1456	16	2.78	0	-0.47222
2017/11/06 09:50:28.0	1	45.8	-0.108	-0.504	-0.846	119.2448	26.04727	12.90811	1100	14	0.26	0	0.1455	17	2.67	0	0.72222
2017/11/06 09:50:29.0	1	43.2	-0.072	-0.588	-0.846	119.2448	26.04736	12.49465	1075	14	0.26	0	0.1454	17	2.75	0	-1.11111
2017/11/06 09:50:30.0	1	39.2	0	-0.648	-0.684	119.2449	26.04745	11.48693	1025	15	63.84	0	0.1455	18	2.53	0	-1.58333
2017/11/06 09:50:31.0	1	33.5	0	-0.522	-0.864	119.2449	26.04753	10.05873	925	17	0.26	0	0.1457	21	2.8	0	-1.13889
2017/11/06 09:50:32.0	1	29.4	-0.09	-0.504	-0.864	119.245	26.04761	9.995721	1037	15	0.26	0	0.1458	19	2.73	0	-0.36111
2017/11/06 09:50:33.0	1	28.1	-0.054	-0.486	-0.846	119.245	26.04768	8.555524	1037	15	0.26	0	0.1459	19	2.69	0	-0.30556
2017/11/06 09:50:34.0	1	27	0	-0.522	-0.774	119.245	26.04775	7.680226	1025	14	0.24	0	0.147	18	2.63	0	-0.63889
2017/11/06 09:50:35.0	1	24.7	0	-0.576	-0.774	119.245	26.04781	6.796928	962	16	0.26	0	0.1469	20	2.68	0	-0.91667
2017/11/06 09:50:36.0	1	21.4	0	-0.612	-0.756	119.2451	26.04786	6.167724	950	17	0.24	0	0.1469	22	2.59	0	-1.30556
2017/11/06 09:50:37.0	1	16.7	0	-0.504	-0.81	119.2451	26.0479	4.119064	900	17	0.24	0	0.1468	21	2.56	0	-0.77778
2017/11/06 09:50:38.0	1	13.9	0	-0.504	-0.828	119.2451	26.04794	4.230131	800	20	0.26	0	0.1469	25	2.79	0	-0.69444
2017/11/06 09:50:39.0	1	11.4	0	-0.504	-0.864	119.2451	26.04796	3.003927	1025	16	0.26	0	0.1471	21	2.71	0	-0.77778
2017/11/06 09:50:40.0	1	8.6	0	-0.54	-0.81	119.245	26.04798	2.303262	762	19	0.24	0	0.147	24	2.58	0	-0.36111
2017/11/06 09:50:41.0	1	7.3	0	-0.432	-0.882	119.245	26.048	1.983072	700	23	0.26	0	0.1471	29	2.69	0	-0.08333
2017/11/06 09:50:42.0	1	7	0	-0.342	-0.918	119.245	26.04802	1.790304	1350	36	0.73	0.105	0.1468	45	7.52	0	0.083333
2017/11/06 09:50:43.0	1	7.3	0	-0.342	-0.918	119.245	26.04803	1.900845	1525	38	0.9	0.125	0.1458	44	9.27	0	0.444444
2017/11/06 09:50:44.0	1	8.9	-0.072	-0.342	-0.918	119.245	26.04806	3.017187	1875	40	1.18	0.13	0.1453	48	12.06	0	1
2017/11/06 09:50:45.0	1	12.5	0	-0.396	-0.9	119.2451	26.0481	4.320009	2200	32	1.08	0.12	0.1456	38	11.02	0	0.611111
2017/11/06 09:50:46.0	1	14.7	0	-0.378	-0.846	119.2451	26.04814	4.884634	1687	41	1.08	0.125	0.1461	49	11.07	0	0.888889
2017/11/06 09:50:47.0	1	17.9	0	-0.378	-0.918	119.2451	26.0482	6.388304	1787	40	1.13	0.13	0.1464	49	11.53	0	0.694444
2017/11/06 09:50:48.0	1	20.4	-0.072	-0.396	-0.9	119.2451	26.04826	6.809328	2000	53	1.67	0.155	0.1458	60	17.1	0	0.75
2017/11/06 09:50:49.0	1	23.1	0	-0.36	-0.918	119.2451	26.04833	7.925755	2137	50	1.65	0.155	0.1458	59	16.91	0	0.722222
2017/11/06 09:50:50.0	1	25.7	0	-0.45	-0.882	119.2451	26.0484	8.166877	2087	30	1	0.12	0.1471	36	10.18	0	1.305556
2017/11/06 09:50:51.0	1	30.4	-0.072	-0.432	-0.99	119.2451	26.04851	12.23798	1662	35	0.92	0	0.1472	46	9.42	0	0.277778
2017/11/06 09:50:52.0	1	31.4	0	-0.468	-0.81	119.2451	26.0486	9.836023	1337	14	0.31	0	0.1463	17	3.17	0	0.138889
2017/11/06 09:50:53.0	1	31.9	0	-0.54	-0.828	119.2451	26.0487	10.78777	1112	15	0.27	0	0.1471	18	2.91	0	-0.38889
2017/11/06 09:50:54.0	1	30.5	0	-0.54	-0.882	119.2451	26.04878	9.120179	1000	16	0.26	0	0.1469	19	2.73	0	-0.44444
2017/11/06 09:50:55.0	1	28.9	0	-0.54	-0.828	119.2451	26.04886	8.460262	975	16	0.27	0	0.1471	20	2.84	0	-0.83333
2017/11/06 09:51:00.0	1	15.9	-0.09	-0.486	-0.828	119.2451	26.04912	4.519031	950	16	0.24	0	0.1467	20	2.58	0	-0.36111
2017/11/06 09:51:01.0	1	14.6	-														

主要实现代码（输出各个特征误差）

```
#path = r'C:\Users\dell\Desktop\拼接数据.xlsx'
path = r'C:\Users\dell\Desktop\分类数 4\拼接 4.xlsx'
data = pd.read_excel(path, header=None)
speed_avg = 0      #平均速度      1  29.98
speed_run_avg = 0  #平均行驶速度  2  38.92
temp1 = 0
a_up_avg = 0       #平均加速度    3  0.315
temp2 = 0
a_down_avg = 0     #平均减速度    4  -0.364
temp3 = 0
speed0_ratio = 0   #怠速比      5  0.23
a_up_ratio = 0     #加速比      6  0.38
a_down_ratio = 0   #减速比      7  0.34
speed_s = 0        #速度标准差   8  26.468
a_s = 0            #加速度标准差  9  0.4
speed_max = 0
l = len(data)
for i in range(0,len(data)):
    speed_avg += data[2][i]
    if data[2][i] > 0:
        speed_run_avg += data[2][i]
        temp1 += 1
    if data[2][i] > speed_max:
        speed_max = data[2][i]
    if data[17][i] > 0:
        a_up_avg += data[17][i]
        temp2 += 1
        a_up_ratio += 1
    if data[17][i] < 0:
        a_down_avg += data[17][i]
        temp3 += 1
        a_down_ratio += 1
    if data[2][i] == 0:
        speed0_ratio += 1
speed_avg = speed_avg/l
speed_run_avg = speed_run_avg/temp1
a_up_avg = a_up_avg/temp2
a_down_avg = a_down_avg/temp3
a_up_ratio = a_up_ratio/l
a_down_ratio = a_down_ratio/l
speed0_ratio = speed0_ratio/l
ss = 0
```

```

sss = 0
nnn = 0
for i in range(0,len(data)):
    ss += (data[2][i]-speed_avg)**2
    if data[17][i] > 0:
        sss += (data[17][i] - a_up_avg) ** 2
speed_s = (ss/len(data))**0.5
a_s = (sss/temp2)**0.5
print('平均速度: ', speed_avg)
print('平均行驶速度:', speed_run_avg)
print('平均加速度', a_up_avg)
print('平均减速度', a_down_avg)
print('怠速比', speed0_ratio)
print('加速比', a_up_ratio)
print('减速比', a_down_ratio)
print('速度标准差', speed_s)
print('加速度标准差', a_s)

```

主要实现代码（数据拼接）

```

filePath = "C:/Users/dell/Desktop/分类数 3/3/"
paths = [i for i in os.listdir(filePath)]
ret = pd.DataFrame()
i = 0
for path in paths:
    data = pd.read_excel(filePath+path, header=None)
    ret = ret.append(data, ignore_index=True)
    i+=1
    print(i)
ret.to_excel("C:/Users/dell/Desktop/分类数 3/拼接 3.xlsx", index=None, header=None)
'''

```

```

统计 怠速比 加速比 减速比
path = 'C:\\Users\\dell\\Desktop\\分类数 4\\拼接 1.xlsx'
data = pd.read_excel(path, header=None)

```

```

sum=0
a=0
b=0
for i in range(0,len(data)):
    if data[16][i] == 1:
        sum += 1

```

```

        if data[17][i] > 0:
            a+=1
        if data[17][i] < 0:
            b+=1
    print(sum/len(data)*100)
    print(a/len(data)*100)
    print(b/len(data)*100)

```

主要实现代码（数据处理）

平均速度计算

```

filePath = "C:/Users/dell/Desktop/所有片段按顺序/"
paths = [i for i in os.listdir(filePath)]
f=[]
aaa=0
for path in paths:
    data = pd.read_excel(filePath+path, header=None)
    sum = 0
    for i in range(0,len(data[2])):
        sum += data[2][i]
    speed = sum/len(data[2])
    f.append(speed)

    aaa+=1
print(aaa)
f = pd.DataFrame(f)
f.to_excel("C:/Users/dell/Desktop/save/" + '1 平均速度.xlsx', index=None, header=None)

```

平均行驶速度计算

```

filePath = "C:/Users/dell/Desktop/所有片段按顺序/"
paths = [i for i in os.listdir(filePath)]
f=[]
aaa=0
for path in paths:
    data = pd.read_excel(filePath+path, header=None)
    sum = 0
    l = 0
    for i in range(0,len(data[2])):
        if data[2][i] != 0:
            sum += data[2][i]

```

```

        l += 1
    speed = sum/(l+0.001)
    f.append(speed)

    aaa+=1
    print(aaa)
f = pd.DataFrame(f)
f.to_excel("C:/Users/dell/Desktop/save/" + '2 平均行驶速度.xlsx', index=None, header=None)

```

平均加速度计算

```

filePath = "C:/Users/dell/Desktop/所有片段按顺序/"
paths = [i for i in os.listdir(filePath)]
f=[]
aaa=0
for path in paths:
    data = pd.read_excel(filePath+path, header=None)
    sum = 0
    l = 0
    for i in range(0,len(data[17])):
        if data[17][i] > 0:
            sum += data[17][i]
            l += 1
    speed = sum/(l+0.001)
    f.append(speed)

    aaa+=1
    print(aaa)
f = pd.DataFrame(f)
f.to_excel("C:/Users/dell/Desktop/save/" + '3 平均加速度.xlsx', index=None, header=None)

```

平均减速度计算

```

filePath = "C:/Users/dell/Desktop/所有片段按顺序/"
paths = [i for i in os.listdir(filePath)]
f=[]
aaa=0
for path in paths:
    data = pd.read_excel(filePath+path, header=None)
    sum = 0
    l = 0
    for i in range(0,len(data[17])):
        if data[17][i] < 0:
            sum += data[17][i]

```

```

        l += 1
    speed = sum/(l+0.001)
    f.append(speed)

    aaa+=1
    print(aaa)
f = pd.DataFrame(f)
f.to_excel("C:/Users/dell/Desktop/save/" + '4 平均减速度.xlsx', index=None, header=None)

```

怠速时间比计算

```

filePath = "C:/Users/dell/Desktop/所有片段按顺序/"
paths = [i for i in os.listdir(filePath)]
f=[]
aaa=0
for path in paths:
    data = pd.read_excel(filePath+path, header=None)
    sum = 0

    for i in range(0,len(data[16])):
        if data[16][i] != 0:
            sum += 1
    speed = sum/len(data[16])
    f.append(speed)

    aaa+=1
    print(aaa)
f = pd.DataFrame(f)
f.to_excel("C:/Users/dell/Desktop/save/" + '5 怠速时间比.xlsx', index=None, header=None)

```

加速时间比计算

```

filePath = "C:/Users/dell/Desktop/所有片段按顺序/"
paths = [i for i in os.listdir(filePath)]
f=[]
aaa=0
for path in paths:
    data = pd.read_excel(filePath+path, header=None)
    sum = 0

    for i in range(0,len(data[17])):
        if data[17][i] > 0:
            sum += 1
    speed = sum/len(data[17])

```

```

f.append(speed)

aaa+=1
print(aaa)
f = pd.DataFrame(f)
f.to_excel("C:/Users/dell/Desktop/save/" + '6 加速时间比.xlsx', index=None, header=None)

```

减速时间比计算

```

filePath = "C:/Users/dell/Desktop/所有片段按顺序/"
paths = [i for i in os.listdir(filePath)]
f=[]
aaa=0
for path in paths:
    data = pd.read_excel(filePath+path, header=None)
    sum = 0

    for i in range(0,len(data[17])):
        if data[17][i] < 0:
            sum += 1
    speed = sum/len(data[17])
    f.append(speed)

    aaa+=1
    print(aaa)
f = pd.DataFrame(f)
f.to_excel("C:/Users/dell/Desktop/save/" + '7 减速时间比.xlsx', index=None, header=None)

```