第九届湖南省研究生数学建模竞赛承诺书

我们仔细阅读了湖南省高校研究生数学建模竞赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们完全清楚，在竞赛中必须合法合规地使用文献资料和软件工具，不能有任何侵犯知识产权的行为。否则我们将失去评奖资格，并可能受到严肃处理。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们授权湖南省研究生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

所属学校和学院（请填写完整的全名）：国防科技大学系统工程学院

参赛队员（打印后签名）：1. 王纪凯 王纪凯

2. 陈广州 陈广州

3. 欧阳伟俊 欧阳伟俊

指导教师或指导教师组负责人（打印后签名）：豆亚杰 豆亚杰

日期： 2024 年 7 月 2 日

（请勿改动此页内容和格式。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）

基于互信息与 Stacking 集成学习的人体活动状态判别研究

摘　要

智能手机已成为人手必备的移动终端，本文针对移动终端数据信息，采用**四元数解算、互信息、K-means 算法、Stacking 集成学习、案例推理**等方法，建立了基于互信息与 Stacking 集成学习的人体活动判定模型，利用 **Matlab、Python** 软件进行问题求解，对人物画像具有一定的参考意义。

针对问题 1，建立基于互信息特征选择的 K-means 聚类模型。首先，对数据进行野值修补与卡尔曼滤波降噪处理；然后，利用**四元数**对加速度计与陀螺仪数据进行融合姿态解算，**增加姿态角数据**，根据解算后的姿态角进行**重力去除**；接下来，对每一轴数据提取 7 种**时域特征**与 6 种**频域特征**，共提取到 117 个数据特征，再利用**互信息算法**进行特征选择；最后，基于 K-means 聚类得到活动状态分类模型。

针对问题 2，建立基于 Stacking 集成学习的活动状态判别模型。首先，将数据按问题一中的方法进行处理，利用问题一中的聚类模型对数据进行聚类，聚类正确率为 64.%；然后，针对机器学习的随机森林算法、支持向量机、K 近邻算法等 8 种算法构建单一模型，应用 **5 折交叉验证算法**筛选出分类效果最好的 4 种模型；接着，通过实验分析，发现以随机森林算法、支持向量机、K 近邻算法为初级分类器，梯度提升分类模型为次级分类器的 Stacking 组合判别模型的分类效果最好，**在测试集上的正确率为 92.67%**；最后，利用该判别模型对附件 3 中的数据进行判别。

针对问题 3，建立基于案例推理的人物画像模型。首先，对不同人员活动状态数据进行相关性分析，构建**多元线性回归模型**，得出实验人员的年龄、体重和身高对活动状态数据的影响程度；然后，增加历史活动状态数据作为特征，利用互信息、决策树、随机森林等 5 种分析方法，构建**集成式特征筛选模型**，选出显著性最强的 10 个特征，并对其使用 Spearman 系数进行独立性检验；最后，利用 **Stacking 集成学习**作为**相似案例检索**的基础，得出未知实验人员编号的匹配结果为 **10，7，6，9，13**。

本文的创新工作为利用四元数进行姿态解算，增加姿态角数据，根据姿态角进行重力去除，提取数据时域与频域特征，采用互信息进行特征选择，建立 Stacking 组合模型，并与单一模型进行对比，提出案例推理方法进行人物画像，成功完成人员匹配。

最后，本文对模型进行了评价和推广，所建立模型还可应用于情报监听、人物跟踪、故障诊断等领域，具有一定的普适性。

关键词：互信息　K-means 算法　Stacking 集成学习　案例推理

目录

图目录

表目录

1 绪论

1.1 研究背景

智能手机的推广和普及极大地改变了人们记录和分析日常活动的方式。通过安装健康应用，智能手机能够记录包括步数、运动时间、距离和耗能等在内的多种人体运动状态数据。在此数据基础上，智能手机可以发挥各种各样的健康检测职能，根据用户的活动习惯设置提醒，比如提醒用户站立、喝水或进行休息，整合来自不同应用和设备的数据，为用户提供全面的活动报告，帮助用户更好地了解自己的生活习惯，乃至于提供个性化的健康和运动建议，帮助用户制定更有效的健康计划，而用户也可以将活动数据分享到社交媒体，与朋友比较进度，增加运动的乐趣和动力。

智能手机之所以能够记录人体的运动状态信息，主要依靠内置在其中的加速度计和陀螺仪等运动监测传感器实时记录速度，加速度和角速度等活动状态数据。其中，加速度计是用于测量手机在三个轴向 (X, Y, Z) 上的线性加速度变化情况的传感器，当手机发生加速度变化时，加速度计会感应到这种变化并将加速度数据传输给手机处理器进行存储或分析。陀螺仪是用来测量手机绕着三个轴向 (X, Y, Z) 旋转的角速度的传感器，当携带手机的人发生转向时，陀螺仪会感知到转向的速度和方向，并将这些数据传输给手机处理器进行存储或处理。通过加速度计和陀螺仪的数据，智能手机处理器可以通过感知到手机的姿态、角度和方向的变化，并设计相应的算法对手机使用人的活动模式进行判断或跟踪。相较于其他定位方式，惯性定位模块可穿戴，不易受到外部环境干扰，因此定位精度较高。工作原理通常基于微机电系统（MEMS）技术，如下图所示。

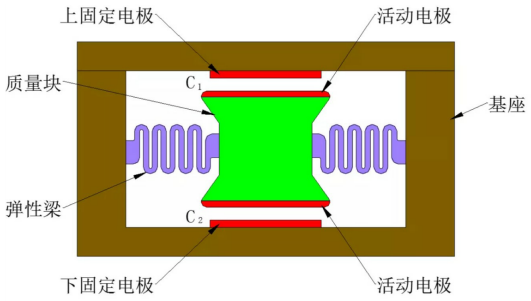


图 1.1　加速度计工作原理

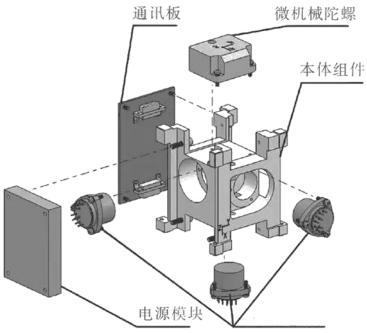


图 1.2　陀螺仪工作原理

查询相关资料可知，加速度计是一种用于测量加速度的传感器，它可以检测设备在各个方向上的加速度变化。MEMS 加速度计内部通常包含一个悬挂的、质量较小的物体（称为质量块），它通过弹簧悬挂在固定的框架上。当设备加速时，质量块会因为惯性作用而相对于框架移动。质量块的位移通常通过压电效应或电容变化来检测。压电效应是指某些材料在受到压力时会产生电荷，而电容变化则是指质量块的位移改变了与固定电极之间的距离，从而改变了电容器的电容，如图 (1.1)。加速度计将检测到的位移转

换为电信号，这些信号可以被进一步处理，以确定设备的加速度。陀螺仪是一种测量角速度的传感器，它可以检测设备绕各个轴的旋转速率。MEMS 陀螺仪通常使用振动的物体（如振动轮或振动梁）来检测旋转。这些振动物体在没有旋转时保持稳定，但当设备旋转时，由于科里奥利力的作用，它们会相对于设备发生偏转。与加速度计类似，陀螺仪中的位移也是通过电容变化来检测的。当振动物体偏转时，它与固定电极之间的距离变化，导致电容变化，如图 (1.2)。陀螺仪将检测到的角速度转换为电信号，这些信号可以用来确定设备的旋转速率和方向。加速度计和陀螺仪通常一起使用，以提供更全面的运动检测。在智能手机中，加速度计可以用来检测设备的线性加速度，而陀螺仪则用来检测设备的旋转运动，两者结合可以提供更精确的运动识别，追踪和分析功能。

1.2 实验设计

为收集人体日常活动状态数据，对人体活动状态进行判断或跟踪，本赛题实验邀请了 10 余名实验人员右下腹位置携带加速度计和陀螺仪两类运动状态传感器进行活动，常见的手机内置运动状态传感器如图 (1.3)。规定其完成“向前走，向左走，向右走、步行上楼、步行下楼、向前跑、跳跃、坐下、站立、躺下、乘坐电梯向上移动、乘坐电梯向下移动”12 种活动状态，各活动模式类型及编码如表 (1.1) 所示。

表 1.1　活动状态类型及编码

活动状态	编号	活动状态	编号
向前走	1	跳跃	7
向左走	2	坐下	8
向右走	3	站立	9
步行上楼	4	躺下	10
步行下楼	5	乘坐电梯向上移动	11
向前跑	6	乘坐电梯向下移动	12

每种活动状态记录了 5 组实验数据，每组数据按采样频率 100Hz 记录若干秒的线加速度和角速度数据，得到 X，Y，Z 三个轴向的加速度和角速度记录值共六类运动数据如表 (1.2)，其中 X 轴沿重力方向，Y 轴沿实验人员前进方向，Z 轴与 XY 平面垂直指向身体一侧，传感器三个轴向如图 (1.4)，速度 v 方向表示实验人员前进方向：

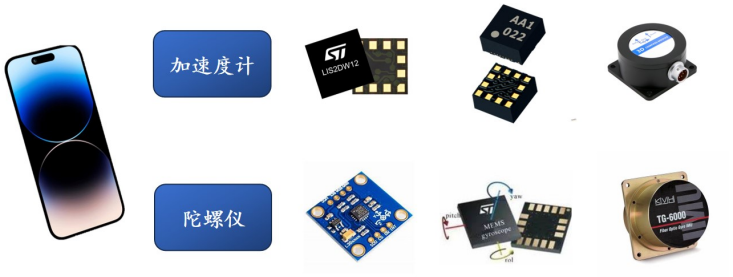


图 1.3　常见内置运动监测传感器

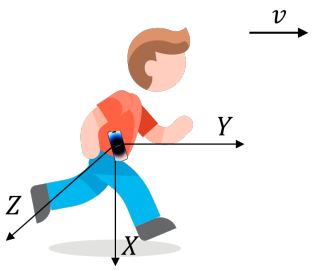


图 1.4　传感器轴向示意图

表 1.2　活动状态数据说明

	acc_x	acc_y	acc_z	gyro_x	gyro_y	gyro_z
设备采样率	100Hz					
运动传感器	加速度计			陀螺仪		
含义	加速度			角速度		
轴向	X 轴	Y 轴	Z 轴	X 轴	Y 轴	Z 轴
单位 (变化范围)	$g(\pm 6g)$			$dps(\pm 500dps)$		

1.3 问题重述

1.3.1 问题 1

问题 1 中，附件 1 采集了 3 名实验人员的活动状态数据，其中每名实验人员每种活动状态数据各 5 组，即每人 60 组数据，但实验时未记录每组数据所代表的活动状态。本题要求根据 3 名实验人员每次活动的的活动状态数据，利用分类模型得到每名实验人员每次活动状态数据所对应的活动状态类别。

1.3.2 问题 2

问题 2 中，附件 2 采集了 10 名实验人员的活动状态数据，其中每名实验人员每种活动状态数据各 5 组，即每人 60 组数据，实验时记录了每组数据所代表的活动状态。本题要求提取 12 类人员活动状态的典型特征，建立针对多用户的统一判别模型，在此基础上开展后续问题研究。

问题 2.1：利用问题 1 中求得的分类模型对该题中 10 名实验人员数据进行分类，比较问题 1 中分类模型与问题 2 中判别模型的求解结果，探究分类模型对不同活动状态类型的分类准确度。

问题 2.2：进一步使用附件 3 中某实验人员的 30 次活动状态数据，通过问题 2 中的判别模型，求得该实验人员每次活动对应的活动状态。

1.3.3 问题 3

问题 3 中，附件 4 给出了问题 1 和问题 2 中参与实验的 13 位实验人员的年龄、身高、体重等数据，要求我们分析同一活动状态对应的活动状态数据是否在不同人员身上存在差异？若有，这些差距与实验人员的年龄、身高、体重等身体特征有无关系？若有，是否可以利用加速度计和陀螺仪这类活动传感器数据对用户的身体特征进行辨别，实现用户画像？在此基础上，附件 5 给出了问题 2 中 10 名实验人员（人员编号为 4 13）中 5 位实验人员的某次活动数据，每次数据均包含了每名实验人员的 12 类活动状态，如何建立综合模型判断这 5 名实验人员分别最可能来自于编号 4 13 中的哪一名。

表 1.3　问题重述

问题序号	新增附件	研究对象			模型	输出
		人员数量	人均活动数据组数	活动状态		
问题 1	附件 1	3 名	60 组	未知	分类模型	各活动数据对应的活动状态分类
问题 2.1	附件 2	10 名	60 组	已知	判别模型	比较分类模型和判别模型准确度差异
问题 2.2	附件 3	1 名	30 组	未知	判别模型	各活动数据对应的活动状态
问题 3	附件 4、5	5 名	60 组	未知	综合模型	各活动数据对应的实验人员编号

1.4 文献调研

1.4.1 人体姿态识别技术研究现状

在人体姿态识别技术的研究历程中，国外科学家们早已展开了深入的探索。在可穿戴传感器技术发展的早期，研究者们主要依赖单轴加速度计来识别人体行为。例如，1997 年，Carlijn 等人[1] 采用三个正交安装的单轴压阻式加速度计，通过高通滤波器去除人体运动中的直流分量，并在腰椎位置佩戴设备，通过幅频信号区分不同活动，尽管初期识别率仅为 30% – 40%。进入新世纪后，2001 年 J. B. J. BUSSMANN 等人[2] 在大腿和胸部放置四个压阻式加速度计，利用角度、运动信号和频率等特征值，通过预设阈值计算特征信号值的距离，成功识别了 20 多种姿态，尽管此过程需半小时。技术的不断进步带来了三轴加速度计的应用，2004 年 Mathie 团队[3] 在腰部安装三轴加速度计，基于二叉树决策树，实现了高达 95% 的识别率。2005 年，Tognetti 等人[4] 开发了集成在服装中的 ULKG，以捕捉肩膀、肘部与腕部的运动。2007 年，Do-Un Jeong[5] 使用三轴加速度计和 Zigbee 无线传感器节点，通过滤波电路消除噪声，成功区分了站立、坐、躺三种状态。

2009 年，Zhenyu He[6] 将三轴加速度计放置于裤子口袋，利用 DCT 和 PCA 技术，结合支持向量机算法，识别了四种日常活动。2013 年，Putchana 等人[7] 利用三轴加速度计

保障老年人活动安全，通过蓝牙技术将数据发送至电脑端进行实时分析。同年，Qiang Li[8] 通过在胸部和大腿放置传感器，分析数据并计算躯干与重力矢量夹角，区分了静态姿势和跌倒状态。2008 年，Yeoh W S[9] 使用三个双轴加速度传感器检测姿态和步行速度，通过判断周期性步态活动和姿态分类器来判定基础活动。2016 年，Verma 团队[10] 使用九轴传感单元，通过 MSE 和 FL 融合技术，有效识别了六种人体姿态及跌倒状态。此外，张洁在 2016 年的研究中[11]，采用 PCA-ReliefF 特征提取算法，结合支持向量机分类器，提高了人体动作识别的准确性。徐川龙在 2013 年的研究中[12]，使用单个三轴加速度计和公开数据库 USC-HAD，通过滤波、特征提取和支持向量机算法，实现了高识别率。王昌喜通过在前臂与后臂放置加速度传感器，采用小波去噪和蚁群算法提取特征，使用支持向量机算法实现了 98% 的正确识别率。李路在 2013 年的研究中[13]，结合三轴加速度计和三轴陀螺仪，采用四元数法和校准技术，通过支持向量机算法进行日常行为姿态识别。

整体来看，人体姿态识别技术经历了从单轴到三轴加速度计的应用，从简单的幅频信号分析到复杂的算法和机器学习技术的发展，不断优化和提升识别的准确性和实时性。

1.4.2 惯性传感器误差修正研究现状

加速度计和陀螺仪作为惯性传感器，以位移计算为例，根据积分定位公式，将起始状态 a_0, v_0, x_0 作为初始条件，通过一次积分和二次积分分别得到手机的速度和位移，然后得到手机具体位置：

$$v = \int_{t_0}^{t} a dt \tag{1.1}$$

$$x = \int_{t_0}^{t} v dt \tag{1.2}$$

其中，t_0 为起始时间，t 为当前时间，a 为瞬时加速度，v 为瞬时速度，s 是人体部位位移。

不难发现，由于惯性传感器测量的加速度与实际情况存在一定差异，经过二次积分后测量得到的位移也会有一定偏差，随着时间积累，这些误差会不断累计，从而使得人体运动状态判断的准确度显著降低。

由于惯性传感器在测试过程中缺乏自适应的误差修正机制，其误差效应会随着时间增加不断积累。惯性传感器误差效应组成较为复杂，从来源上可以简要分为随机漂移，线性加速度干扰和磁力计扰动三个部分。

在短期时间内，惯性传感器误差主要来源于仪器量化误差，但随着时间推进，仪器自身的随机漂移误差将会对传感器精度造成较大影响。有研究也表明仪器自身的随机漂移误差会受到温度的干扰。在硬件自身性能和实验环境受限的基础上，当下研究常采用

零速度更新算法对零速度随机抖动误差进行补偿，希望建立一个基于累计方差平方和最小的隐马尔可夫过程模型，结合人体生物学约束，进而对仪器的随机漂移误差进行动态修正。

基于电容法等方法，加速度计通过测量出该时刻各方向加速度与重力加速度的比值，来估计这个时刻人体部位的俯仰角和翻滚角等状态参数，但人体运动的线加速度会对这个比值造成干扰。因此有研究将人体运动状态用一个状态向量表示，找到一种能够动态减弱甚至消除这种线性加速度的自适应滤波算法。主要思路包括两条，一是通过评判线性加速度相对于重力加速度的大小差距，基于阈值对加速度计数据进行合理权重分配，即在加速度计测量数据受线性加速度干扰较大的的情况下，削弱加速度计数据对人体总体运动状态评判的作用效果。二是基于人体运动的马尔可夫模型，构建卡尔曼滤波系统，基于预测值和实际值的最小均方误差原则，动态修正预测模型框架，调整每次加速度计的测量值。对人体姿态动作判断而言，方法一对数据作用的权重分配算法设计难度较大，方法二预测模型是针对人体部位运动状态的马尔可夫软约束，约束条件过强会使得修正值偏移真实值，过弱则难以起到约束效力，达不到线性加速度修正效果。

当环境周围存在磁性物质时，磁力计受到的磁力干扰则难以忽略。相关研究将磁力干扰分为硬磁干扰和软磁干扰。硬磁干扰产生于周围的永久磁体，其磁力和位置相对固定，在时域上可以认为是时不变干扰因子，一般做零偏处理；相比之下，软磁干扰是由于磁力计周围其他磁性材料作用而产生的，在时域上呈现明显的变化趋势，因此其对于惯性传感系统测量的影响不能忽略。首先根据磁力计自身误差和软硬磁特性建立其误差模型，通过椭球拟合后利用最小二乘平差方法估计出椭球方程系数，从而得到校准后的磁场强度值。

值得注意的是，数据融合的惯性传感器误差修正方法往往可以呈现更好的效果。基于陀螺仪，加速度计和磁力计的惯性测量数据，可以尝试从数据滤波和现代优化算法两个层面对误差进行二次修正。

数据滤波层面包括互补滤波器和卡尔曼滤波器等基本滤波器框架。互补滤波器基于三种惯性器件对于高频和低频噪声分量的敏感性差异，实现不同元器件测量的优势互补，进而实现姿态信息的融合估计。在此框架下，由于人体不同运动状态下的频率特性差异，利用互补滤波对数据进行融合后，再结合卡尔曼滤波得到自适应的互补卡尔曼滤波模型。但这个模型需要假设噪声满足高斯分布，相比之下，粒子滤波则更具有一般性，采用分层融合的思想估计上肢和下肢的运动姿态,第一层为数据融合层,第二层为几何约束融合层。

1.5 本章小结

本章针对该赛题题干内容，分别从研究背景，实验设计，问题重述和文献调研四个部分展开介绍。其中，研究背景部分介绍了智能手机监测人体活动状态数据的功能和意

义，引出了加速度计和陀螺仪这两类常用的运动监测传感器在智能手机运动分析中的关键作用，并介绍了微机电系统（MEMS）技术下加速度计和陀螺仪的主要工作原理；实验设计部分列出了题干中“数据说明”涉及的实验设备，实验方式，实验参数和实验编码等信息，并对这些信息进行了适当的结构化和可视化呈现；问题重述部分明确了题干中各问题的研究对象，输入，输出，模型等要素；文献调研部分分别对人体姿态识别技术，惯性传感器误差修正技术两个方面的研究现状进行了调研。总的来说，本章作为全文的引入章节，从应用情景引入，深入剖析了赛题需要本文解决的核心关键问题，进一步针对这些核心关键问题展开文献调研工作，为后文问题分析，建模和求解工作奠定了坚实基础。

2 问题分析

2.1 问题 1 分析

针对问题 1，挑战在于从原始的传感器数据中提取出能够表征不同活动状态的特征，并应用聚类算法来识别这些状态。这不仅需要对数据进行深入分析，还可能涉及到时间序列分析，以捕捉活动过程中的动态变化。此外，由于数据未标记，本文需要创造性地设计特征，以揭示不同活动的本质区别，除此以外聚类结果的准确性将直接影响到后续问题的解决。

问题 1 的核心是无监督学习，即在没有预先标记数据的情况下，通过算法自动识别数据中的模式。通常包括以下步骤：

(1) 数据预处理：由于传感器数据可能包含噪声和异常值，首先需要进行数据清洗，包括去噪、填补缺失值等。此外，数据标准化也是必要的，以消除不同传感器量纲的影响。

(2) 特征提取：从原始的加速度计和陀螺仪数据中提取有助于区分不同活动的特征。这可能包括时域特征和频域特征等。

(3) 聚类算法选择：选择合适的聚类算法对特征进行分组。K-means 是一种常见的选择，但可能还需要考虑其对初始中心敏感的问题。层次聚类或基于密度的聚类算法（如 DBSCAN）可能更适合处理不规则分布的数据。

模型评估：聚类结果需要通过某种方式进行评估。在没有标签的情况下，可以使用轮廓系数、戴维森堡丁指数等内部评价指标来评估聚类效果。

结果解释：将聚类结果与实际活动状态进行比较，以验证模型的有效性，并根据需要调整聚类算法或特征。

2.2 问题 2 分析

针对问题 2，挑战在于在已知活动标签的基础上，进一步提炼出每种活动状态的典型特征，并建立一个能够准确分类的判别模型。这不仅涉及到特征工程，还需要选择合适的机器学习算法。模型的建立需要考虑到过拟合和欠拟合的问题，可能需要采用正则化技术、交叉验证等策略来优化模型性能。此外，模型的验证工作不仅关注分类准确度，还要求分析模型的泛化能力和鲁棒性。

问题 2 的核心是泛化性能，即要求在已知活动标签的基础上建立判别模型。通常包括以下步骤：

(1) 特征选择：基于问题 1 中提取的特征，进一步选择对分类任务最有区分力的特征。可以使用特征重要性评估、递归特征消除等方法。

(2) 模型选择：选择合适的分类算法。除了传统的算法如 SVM、决策树、随机森林外，深度学习模型，如卷积神经网络（CNN）或循环神经网络（RNN），可能更适合处

理时间序列数据。

(3) 模型训练：使用交叉验证等技术来避免过拟合，并使用网格搜索等方法进行参数调优，以找到最优的模型参数。

(4) 模型验证：使用问题 1 中的分类结果作为基准，评估新模型的分类准确度。这可以通过混淆矩阵、精确度、召回率和 F1 分数等指标来完成。

(5) 模型应用：将模型应用于问题 1 中未标记的数据，以验证模型的泛化能力。此外，还需要在表 2 中对附件 3 中的新数据进行预测。

2.3 问题 3 分析

相比于问题 1 和问题 2，问题 3 的分析更为复杂，它要求参赛者不仅要分析活动状态数据，还要考虑个体差异以及与生理特征的关系。这可能涉及到更高级的统计方法，如混合效应模型，来同时考虑固定效应和随机效应。此外，使用活动传感器数据进行人员画像可能需要应用模式识别和机器学习技术，以识别不同个体的独特模式。最后，使用模型对未知数据进行判别，不仅考验了模型的准确性，还对数据融合和综合分析能力提出了较高要求。

问题 3 的核心是用户画像，要求分析个体差异并使用模型进行数据判别，这涉及到以下步骤：

(1) 个体差异分析：使用统计方法，如 ANOVA 或 ANCOVA，来分析不同实验人员在相同活动状态下的数据是否存在显著差异，并探索这些差异与个体生理特征的关系。

(2) 特征与生理特征关联分析：使用多元回归分析或机器学习算法，如梯度提升树（GBM）或 XGBoost，来建立活动数据与生理特征之间的预测模型。

(3) 人员画像构建：基于活动数据和生理特征，构建每个实验人员的画像。这可能涉及到主成分分析（PCA）或典型相关分析（CCA）等降维技术，以提取最具代表性的特征。

(4) 数据判别：使用之前建立的判别模型，对附件 5 中的未知来源活动数据进行判别，以确定它们最可能来源于问题 2 中的哪一名实验人员。

(5) 结果验证与应用：将判别结果与实际数据进行比较，验证模型的准确性，并探讨模型在实际应用中的潜力，如健康监测、个性化运动建议等。

2.4 本章小结

上述 3 个问题构成了一个从数据预处理到模型建立，再到模型应用和验证的完整流程，要求我们深入了解如何利用智能手机传感器数据进行人体健康和活动状态分析，同时具备数据处理、特征工程、机器学习、统计分析等多方面的知识，以及创新思维和问题解决能力。基于上述对各问题的分析，确立本文研究框架如图 (2.1)：

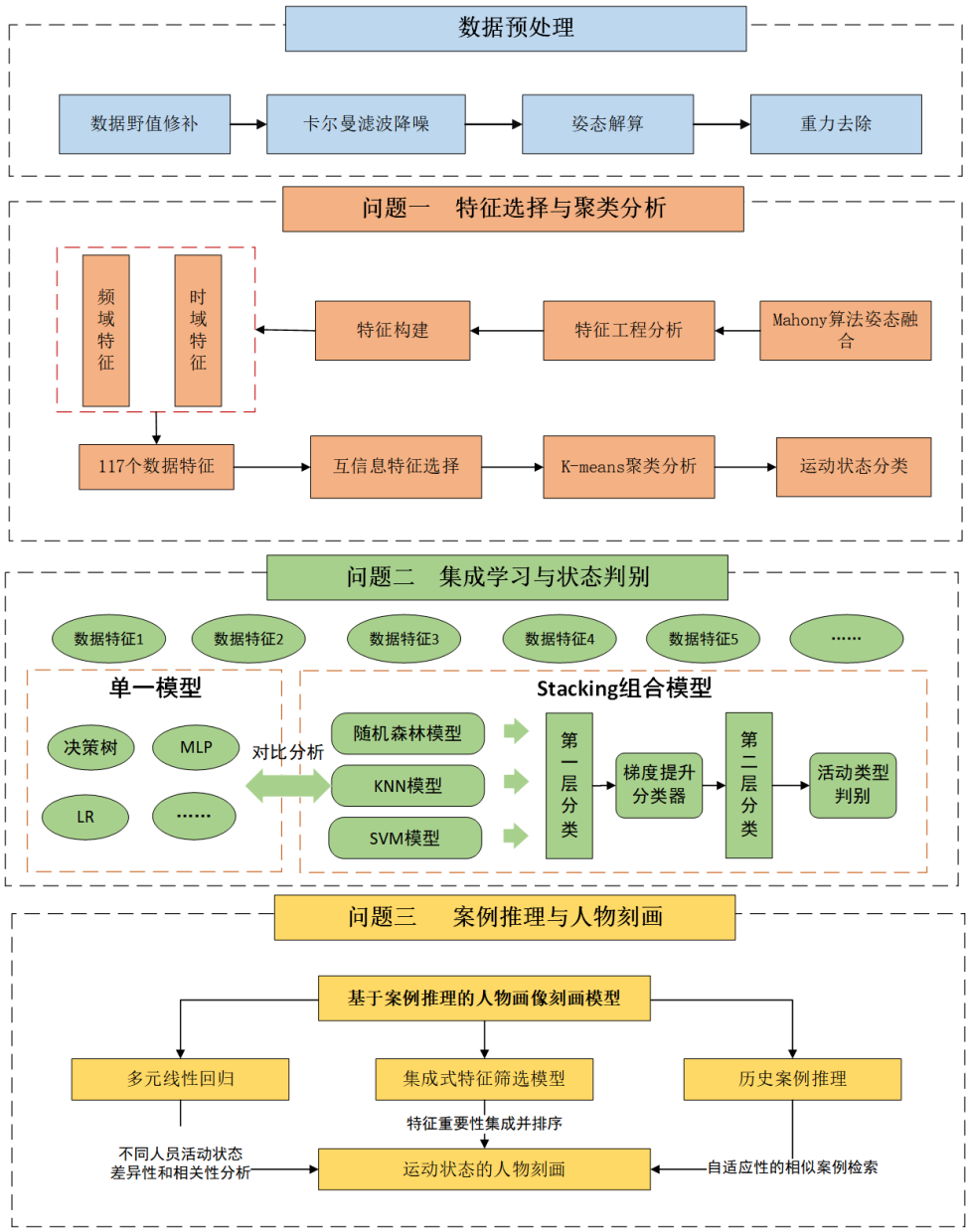


图 2.1　研究框架

3 数据预处理

附件中的数据集是在完全真实的生活状态中采集的数据，存在一定的不一致、不平衡、噪声值、异常值的数据，需要先对数据进行预处理。数据预处理主要包括数据野值修补，卡尔曼滤波，姿态解算和重力去除。

3.1 预备知识

定义 3.1 四元数定义

四元数最早由 *William Rowan Hamilton* 于 *1843* 年提出，基本定义如式(3.1)–(3.2)：

$$q = q_0 + q_1 i + q_2 j + q_3 k \tag{3.1}$$

$$i^2 = j^2 = k^2 = ijk = -1 \tag{3.2}$$

其中，$q_0, q_1, q_2, q_3 \in \mathbb{R}$，$q_0$ 表示四元数 q 的实部，$q_1 i + q_2 j + q_3 k$ 表示四元数 q 的虚部。

定义 3.2 四元数向量形式

和复数类似，四元数也可以写成向量形式，如式(3.3)：

$$q = (q_0, q_1, q_2, q_3)^T \tag{3.3}$$

定义 3.3 四元数有序对形式

若将实部与虚部分开，通过一个三维向量表示虚部，四元数还可以表示为标量与向量的有序对形式，如式(3.4)：

$$q = (s, v) \ \ (v = (x, y, z)^T, s, x, y, z \in \mathbb{R}) \tag{3.4}$$

定义 3.4 四元数乘法

如果有两个四元数 $q_1 = a + bi + cj + dk$ 和 $q_2 = e + fi + gj + hk$ 相乘，结合式(3.2)，可得式(3.5)：

$$\begin{aligned} q_1 q_2 =& (ae - bf - cg - dh) + (be + af - dg + ch)i + \\ & (ce + df + ag - bh)j + (de - cf + bg + ah)k \end{aligned} \tag{3.5}$$

显然，任意两个四元数之间相乘遵循乘法分配律和乘法结合律，但不一定遵循乘法交换律。

四元数乘法运算写成矩阵形式则有：

$$q_1 q_2 = \begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix} \begin{bmatrix} e \\ f \\ g \\ h \end{bmatrix} \tag{3.6}$$

3.2 数据野值修补

在工程实践中，受设备，环境，数据精度等多重因素影响，采集的数据中往往不可避免地存在一定偏离被测信号目标真值的成分，即数据野值，根据是否连续情况，可将数据野值分为孤立型和斑点型两类。本文采取改进的均方误差野值识别算法，包括数据局部化处理，矩阵线性中心化，以及野值判别与修补三个步骤：

(1) 数据局部化处理

将没 n 个测量数据划分为一个数据帧，依次得到 m 个连续局部数据帧，随后对每一个局部帧分别实施野值判断与修补

定义向量矩阵化算子 mat，对向量 x 进行矩阵化运算，可得局部信号矩阵 X。X 由 m 个 n 维向量构成，即 m 个 n 维数据帧，如式(3.7)。

$$X = \text{mat}(x) = (x_1, x_2, x_3, \ldots, x_m) = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{m1} \\ x_{12} & x_{22} & \cdots & x_{m2} \\ x_{13} & x_{23} & \cdots & x_{m3} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n} & x_{2n} & \cdots & x_{mn} \end{bmatrix} \tag{3.7}$$

(2) 矩阵线性中心化

定义向量均值求解算子 mean 和方差求解算子 std，可得局部信号 x_i 的均值和方差分别表示为 $\mu = \text{mean}(x_i)$ 和 $\sigma = \text{std}(x_i)$。在此基础上进行线性变换，由向量中心化公式 $\overline{x_i} = x_i - \mu$ 求得中心化矩阵 $\overline{X} = (\overline{x_1}, \overline{x_2}, \overline{x_3}, \ldots, \overline{x_m})$

(3) 野值判别与修补

对于中心化矩阵 $\overline{X}$ 中元素 $\overline{x_{ij}}$，若满足式(3.8)：

$$\overline{x_{ij}} > k\sigma \tag{3.8}$$

则判断 x_{ij} 为野值，其中，k 的取值与局部向量 x_i 的长度 n 有关，经验取值公式如式(3.9)：

$$k = 2.50 + n/160 \tag{3.9}$$

如果 x_{ij} 被判别为野值，采用该野值点前后各两个数值的加权值来替代该野值点数据，如式(3.10)：

$$x_{ij} = 0.2x_{i,j-2} + 0.3x_{i,j-1} + 0.3x_{i,j+1} + 0.2x_{i,j+2} \tag{3.10}$$

3.3 基于卡尔曼滤波的降噪处理

卡尔曼滤波 (Kalman fitering，KF) 是一种利用线性系统状态方程，通过系统输入输出观测数据，对系统状态进行最优估计的算法。由于附件中的观测数据中包括系统中的

噪声和干扰的影响，所以最优估计也可看作是滤波过程。卡尔曼滤波在测量方差已知的情况下能够从一系列存在测量噪声的数据中，估计动态系统的状态，通过调整观测值和估计值的权重，使修正后的值更趋向于相信，流程如图 (3.1)：

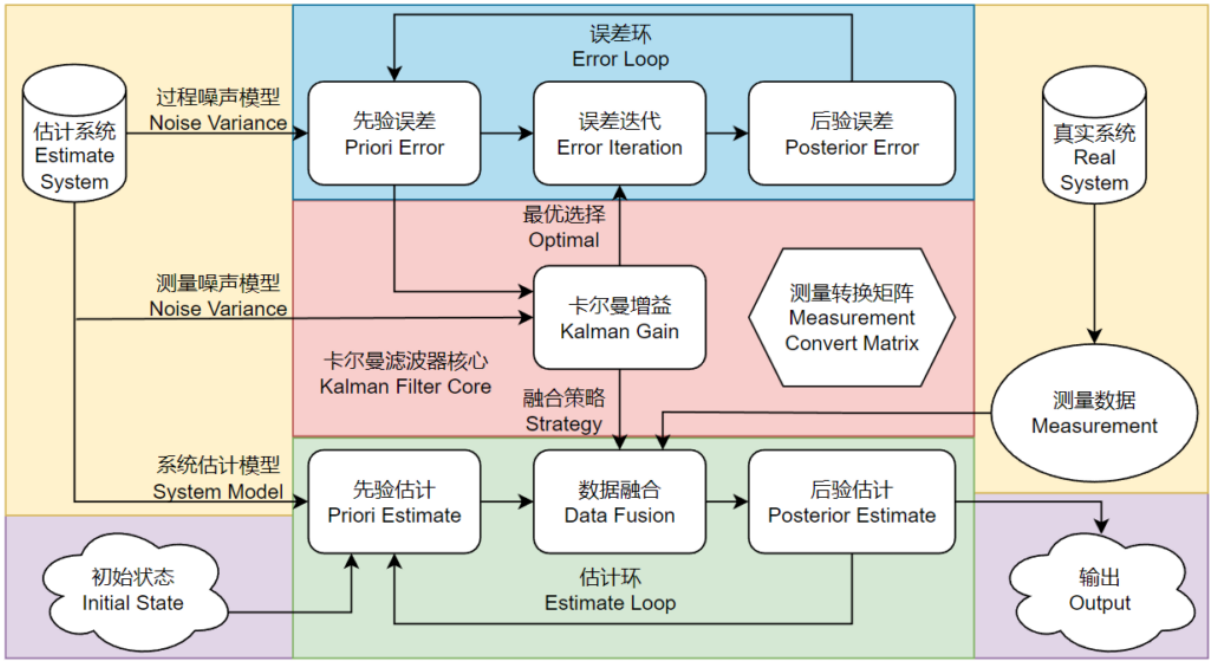


图 3.1　卡尔曼滤波原理

卡尔曼滤波建立在线性代数和隐马尔可夫模型上，其基本动态系统可以用一个马尔可夫链表示，该马尔可夫链建立在一个被高斯噪声（即正态分布的噪声）干扰的线性算子上的，系统状态可以用一个元素为实数的向量表示。随着离散时间的每一个增加，这个线性算子就会作用在当前状态上，产生一个新的状态，并也会带入一些噪声，同时系统的一些已知的控制器的控制信息也会被加入。同时，另一个受噪声干扰的线性算子产生出这些隐含状态的可见输出。

为了从一系列有噪声的观察数据中用卡尔曼滤波器估计出被观察过程的内部状态，必须把这个过程在卡尔曼滤波的框架下建立模型。也就是说对于每一步 k，定义矩阵 F_k, H_k, Q_k, R_k，有时也需要定义 B_k ，具体滤波过程如下。

卡尔曼滤波模型假设 $k+1$ 时刻的真实状态是从 k 时刻的状态演化而来，符合状态方程(3.11):

$$x_k = F_k x_{k-1} + B_k u_k + w_k \tag{3.11}$$

其中，F_k 是作用在 x_{k-1} 上的状态变换模型（矩阵或向量），B_k 是作用在控制器向量 u_k 上的输入-控制模型，w_k 是过程噪声，并假定其符合均值为零，协方差矩阵为 Q_k 的多元正态分布，满足式(3.12)：

$$w_k \sim N\left(0, Q_k\right) \tag{3.12}$$

其中 $Q(k) = \text{cov}\{w(k)\}$ 表示协方差矩阵。

在时刻 k，对真实状态 x_k 的一个测量 z_k 满足观测方程(3.13)：

$$z_k = H_k x_k + v_k \tag{3.13}$$

其中 H_k 是观测模型，它把真实状态空间映射成观测空间，v_k 是观测噪声，其均值为零，协方差矩阵为 R_k 且服从正态分布，满足式(3.14)：

$$v_k \sim N(0, R_k) \tag{3.14}$$

其中 $R(k) = \text{cov}\{v(k)\}$ 表示协方差矩阵，初始状态以及每一时刻的噪声 $\{x_0, w_1, \ldots, w_k, v_1, \ldots, v_k\}$ 都认为是互相独立的，比较三个轴向上加速度去重力和卡尔曼滤波后波形变化如图(3.2)：

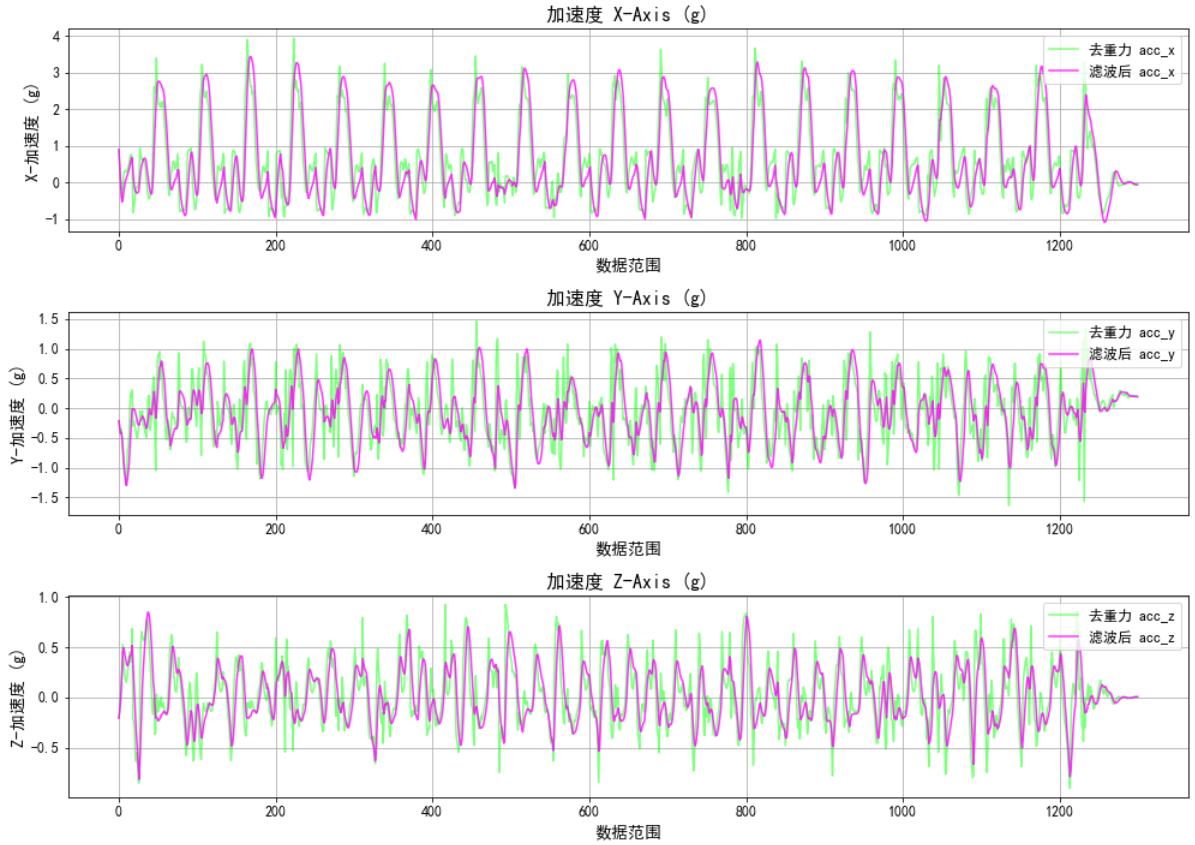


图 3.2　三个轴向上加速度去重力和卡尔曼滤波后波形变化

3.4 姿态解算

惯性传感器的姿态解算是指利用加速度计、陀螺仪等传感器数据，通过数学模型和算法计算出物体相对于某一参考坐标系的方向和姿态变化。以相对于大地坐标系静止的初始轴向坐标系为默认坐标系，姿态解算中的姿态实际上是人体坐标系与默认坐标系之间的旋转关系。常用旋转描述形式包括欧拉角、方向余弦矩阵和四元数三类，考虑到

欧拉角存在万向节死锁问题，而方向余弦矩阵中大量的三角函数运算会显著降低计算效率，相较之下，四元数同时满足描述完备和计算方便两个特点，因此本文基于四元数展开姿态解算研究。

设初始姿态角向量为 $\Phi=(\phi_x,\phi_y,\phi_z)^T$,绕着以单位向量定义的旋转轴 $\mathbf{u}=(u_x,u_y,u_z)^T$ 旋转 θ 度，则新的姿态角 Φ' 计算如式(3.15)：

$$v'=qvq^{-1} \tag{3.15}$$

即罗德里格四元数旋转公式，其中 v 表示初始姿态角四元数，q 表示旋转轴四元数，符号定义如式(3.16)：

$$v=[0,\Phi],\quad q=\left[\cos\left(\frac{1}{2}\theta\right),\sin\left(\frac{1}{2}\theta\right)\mathbf{u}\right] \tag{3.16}$$

由定义3.4，求得罗德里格四元数旋转公式的矩阵形式如式(3.17)：

$$v'=qvq^{-1}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1-2q_2^2-2q_3^2 & 2q_1q_2-2q_0q_3 & 2q_0q_2+2q_1q_3\\ 0 & 2q_1q_2+2q_0q_3 & 1-2q_1^2-2q_3^2 & 2q_2q_3-2q_0q_1\\ 0 & 2q_1q_3-2q_0q_2 & 2q_0q_1+2q_2q_3 & 1-2q_1^2-2q_2^2\end{bmatrix}v \tag{3.17}$$

其中，旋转轴四元数 q 的参数 q_0,q_1,q_2,q_3 取值如式(3.18)：

$$q_0=\cos\frac{\theta}{2},q_1=\sin\frac{\theta}{2}u_x,q_2=\sin\frac{\theta}{2}u_y,q_3=\sin\frac{\theta}{2}u_z \tag{3.18}$$

由于矩阵的第一行和第一列不会对 v 进行任何变换，所以我们可以将式(3.17)压缩为 3 维方阵，求得矩阵旋转公式如式(3.19)：

$$\Phi'=\begin{bmatrix}1-2q_2^2-2q_3^2 & 2q_1q_2-2q_0q_3 & 2q_0q_2+2q_1q_3\\ 2q_1q_2+2q_0q_3 & 1-2q_1^2-2q_3^2 & 2q_2q_3-2q_0q_1\\ 2q_1q_3-2q_0q_2 & 2q_0q_1+2q_2q_3 & 1-2q_1^2-2q_2^2\end{bmatrix}\Phi \tag{3.19}$$

可得人体坐标系 P' 相较于默认坐标系 P 的姿态矩阵 $C_P^{P'}$ 如式(3.20)：

$$C_P^{P'}=\begin{bmatrix}1-2q_2^2-2q_3^2 & 2q_1q_2-2q_0q_3 & 2q_0q_2+2q_1q_3\\ 2q_1q_2+2q_0q_3 & 1-2q_1^2-2q_3^2 & 2q_2q_3-2q_0q_1\\ 2q_1q_3-2q_0q_2 & 2q_0q_1+2q_2q_3 & 1-2q_1^2-2q_2^2\end{bmatrix} \tag{3.20}$$

基于姿态矩阵 $C_P^{P'}$，反解出人体坐标系的欧拉角如式(3.21)：

$$\begin{cases}\phi_x=arcsin[2(q_0q_2-q_1q_3)],\\ \phi_y=arctan(\dfrac{q_0q_3+q_1q_2}{1-2(q_2^2+q_3^2)}),\\ \phi_z=arctan(\dfrac{q_0q_1+q_2q_3}{1-2(q_1^2+q_2^2)}).\end{cases} \tag{3.21}$$

定义一个单位四元数如式(3.22)：

$$Q = cos\frac{\theta}{2} + \vec{u}sin\frac{\theta}{2} \tag{3.22}$$

Q 对时间 t 进行微分，如式(3.23)：

$$\frac{dQ}{dt} = \frac{1}{2}\begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{3.23}$$

其中，w_x, w_y, w_z 分别表示 X, Y, Z 轴向上的角速度值

求解该微分方程即可得到当前四元数的值。为便于计算机求解，我们需要对该微分方程进行离散化处理，如式(3.24)：

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_{t_Q+\Delta t} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_{t_Q} + \frac{1}{2}\Delta t \begin{bmatrix} -w_x q_1 - w_y q_2 - w_z q_3 \\ w_x q_0 + w_z q_2 - w_y q_3 \\ w_y q_0 - w_z q_1 + w_x q_3 \\ w_z q_0 + w_y q_1 - w_x q_2 \end{bmatrix} \tag{3.24}$$

其中，t_Q 表示 Q 状态对应的实际时刻，Δt 表示离散采样间隔，则 $t_Q + \Delta t$ 表示下一离散采样时刻。

对于六轴数据，对姿态角进行计算的方法主要有两种，一种是对角速度积分，另一种是对加速度进行正交分解角度。但这两种方式均存在不足，通过角速度积分得到角度时，角速度的误差会在积分过程中被不断放大从而影响数据准确性。而加速度计是一种特别敏感的传感器，电机旋转产生的震动会给加速度计的数据中带来高频噪声。第一种方法测得的数据中存在结果偏差，而第二种方法测得的数据中存在高频噪声。因此可通过融合两种数据以获得准确姿态，这里通过加速度计补偿角速度。

不难看出，将重力加速度向量变换至机体坐标系后，恰好是矩阵的最后一列。这样一来，我们就得到了由描述刚体姿态的四元数推导出的理论重力加速度向量。另外，我们还可以通过加速度计测量出实际重力加速度向量显然，这里的理论重力加速度向量和实际重力加速度向量之间必然存在偏差，而这个偏差很大程度上是由陀螺仪数据产生的角速度误差引起的，所以根据理论向量和实际向量间的偏差，就可以补偿陀螺仪数据的误差，进而解算出较为准确的姿态，即将隐含在四元数中的角速度误差显化，如式(3.25)。

$$\hat{\boldsymbol{v}} = C_R^b \cdot \mathbf{g} = (C_b^R)^{-1} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2\left(q_1 q_3 - q_0 q_2\right) \\ 2\left(q_2 q_3 + q_0 q_1\right) \\ 1 - 2\left(q_1^2 + q_2^2\right) \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{3.25}$$

理论重力加速度向量和实际重力加速度向量均是向量，反应向量间夹角关系的运算有两种：内积（点乘）和外积（叉乘），考虑到向量外积模的大小与向量夹角呈正相关，故通过计算外积来得到向量方向差值 $|\boldsymbol{\rho}|$ 如式(3.26)：

$$|\boldsymbol{\rho}| = |\overline{\boldsymbol{v}}| \cdot |\hat{\boldsymbol{v}}| \cdot \sin\theta \tag{3.26}$$

在进行叉乘运算前，应先将理论向量和实际向量做单位化处理，如式(3.27)：

$$|\boldsymbol{\rho}| = \sin\theta \tag{3.27}$$

考虑到实际情况中理论向量和实际向量偏差角不会超过 45°，而当 θ 在 ±45° 内时，sinθ 与 θ 的值非常接近，因此上式可进一步简化为式(3.28)：

$$|\boldsymbol{\rho}| = \theta \tag{3.28}$$

得到向量偏差后，即可通过构建 PI 补偿器来计算角速度补偿值如式(3.29)：

$$GyroError = K_P \cdot \text{ error } + K_I \cdot \int \text{error} \tag{3.29}$$

其中，比例项 K_P 和 K_I 用于控制传感器的“可信度”，积分项用于消除静态误差。K_P 越大，意味着通过加速度计得到误差后补偿越显著，即是越信任加速度计。反之，K_P 越小时，加速度计对陀螺仪的补偿作用越弱，也就越信任陀螺仪。而积分项则用于消除角速度测量值中的有偏噪声，故对于经过零偏矫正的角速度测量值，一般选取很小的 K_I。最后将补偿值补偿给角速度测量值，带入四元数差分方程中即可更新当前姿态角。

3.5 基于姿态角的重力去除

手机的加速度传感器，集成在手机中的三轴 (X, Y, Z) 加速度传感器分为两种：一种是由于恒定重力作用产生的加速度，另一种是用户握持设备运动产生的加速度，为了获取真实运动状态需将重力剔除，X, Y, Z 三个轴向上加速度数值进行重力去除后如式(3.30)−(3.32)：

X 轴的加速度：

$$a_{xg} = a_x + 0.98 * \sin(y) \tag{3.30}$$

Y 轴的加速度：

$$a_{yg} = a_y - 0.98 * \sin(x) * \cos(y) \tag{3.31}$$

Z 轴的加速度：

$$a_{zg} = a_z - 0.98 * \cos(x) * \cos(y) \tag{3.32}$$

其中，三角函数中的 x, y, z 分别为绕三轴的姿态角 (pitch，yaw，roll)，单位为弧度制。

本文利用附件 1 中的 Person1 的 SY1 数据进行可视化展示，如图 (3.3)：

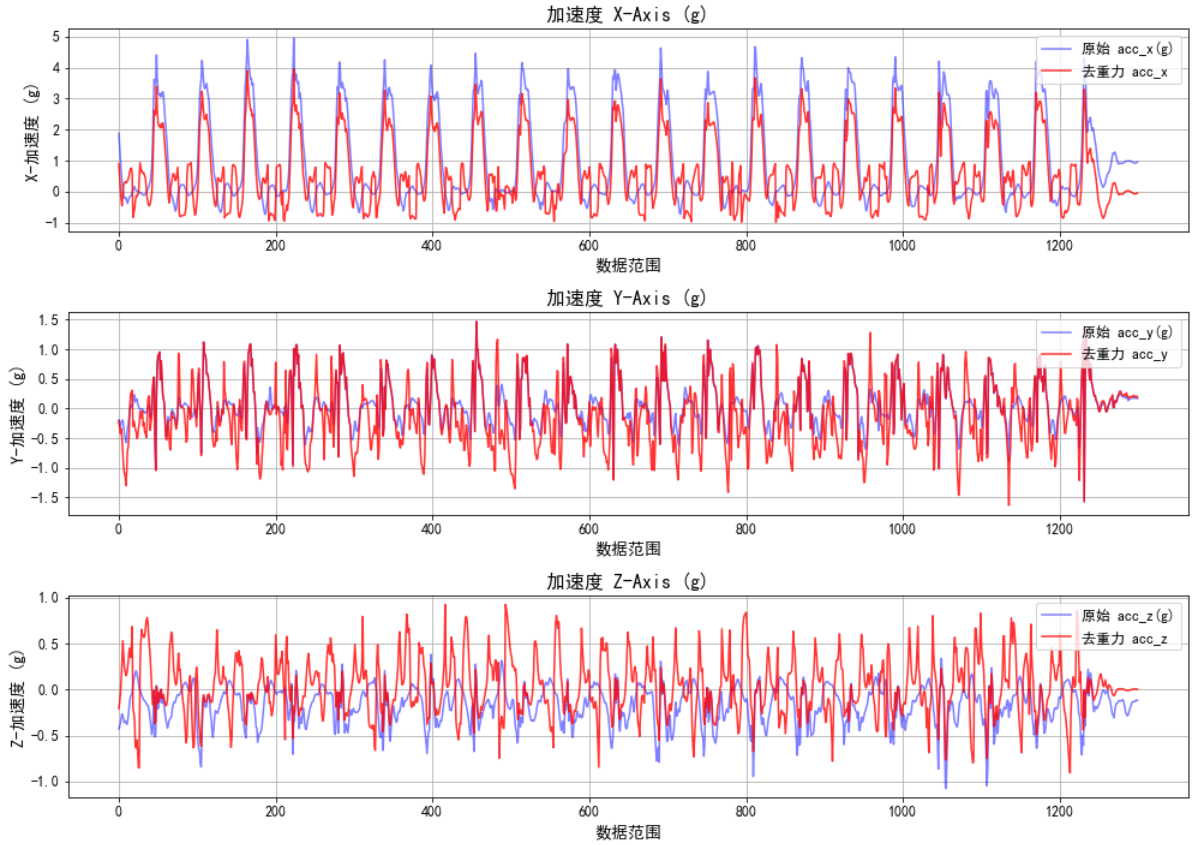


图 3.3　SY1 可视化

附件中的原始数据在各轴上均存在稳定的偏移量，这主要是由于重力分量的存在。去除重力后的数据中心接近于零，波动数值上更能反映实际运动。在 Z 轴方向上的数据波动最为显著，表明传感器在该方向上的运动特征最为明显。处理后的数据更加适用于进一步的运动特征分析和异常检测，有助于准确识别手机传感器的真实运动状态，帮助对人体活动状态的判断。

4 问题 1 建模与求解

根据问题 1 要求，需要对提供的活动状态数据（每人 60 组数据），对每一位实验人员的活动状态的数据进行分类。观察数据，每组数据包含加速度计和陀螺仪在 X, Y, Z 三个轴向上的加速度和角速度的记录值，测量过程会存在重力的影响，由于传感器本身也会存在测量误差，需要进行去重力和去噪处理，接着从时域和频域角度提取数据的特征。数据特征提取的好坏将直接影响到活动状态分类效果，而数据特征过多又会导致模型的过拟合。因此，为了优化模型，对于这样的高维多样本数据，我们通常需要对提取的特征进行预处理，从中筛选出重要且独立性好的特征。

特征选择与特征提取并不一致，特征提取是一种基于变换的方法，通过坐标变换的方式将原始的高维数据映射到低维空间，而特征选择可以看作从初始特征空间搜索出一个最优特征子集的过程，其并不改变原始特征空间，只是选择一些分辨力好的特征，组成一个新的低维空间，可以在保留原始特征空间大部分性质的前提下去除不相关特征和冗余特征，在一定程度上将噪声数据对分类器性能的影响降到最低。

基于上述分析，按照第 3 章方法完成数据预处理，得到 X, Y, Z 三个轴向上姿态角数据后，我们从特征提取和特征选择两个角度切入展开问题 1 的建模与求解。

4.1 特征提取

通过加速度计和陀螺仪进行活动状态判断时，这些惯性传感器能够准确捕捉到人体运动的细微变化，对活动状态数据的时域特征和频域特征进行提取能够获得不同角度的特征信息，有助于更准确地识别和分类用户不同的活动状态。

4.1.1 时域特征

时域特征指的是在时间序列数据中，通过观察信号在时间轴上的变化来提取的特征。这些特征通常反映了信号的时序特性，如周期性、趋势、稳定性等。对于离散活动状态数据帧，根据 X, Y, Z 轴上的加速度，角速度和姿态角数据计算时域特征。具体而言，对于某组活动状态数据中一个传感器某一轴的感知数据，用 N 表示其在采样时间内的样本数，用 C_i 表示第 i 个样本数据的绝对值，max 表示最大值，min 表示最小值，num 表示符合条件的元素个数，获得某一轴的数据片段对应的均值、标准差、众数、最大值、最小值、极差和过均值率这 7 类时域特征.

(1) 均值：数据集中所有数值的总和除以数值的个数。它是数据集中所有数据点的平均值，用来衡量数据的中心位置。

$$\mu = \frac{1}{N}\sum_{i=1}^{N} C_i \tag{4.1}$$

(2) 标准差：数据集中所有数值的总和除以数值的个数。它是数据集中所有数据点

的平均值，用来衡量数据的中心位置。

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(C_i - \mu)^2} \tag{4.2}$$

(3) 众数：数据集中出现次数最多的数值，若有多个众数则取平均值作为唯一众数。

(4) 最大值：也被称作峰值，数据集中所有数值中的最大值。它代表了数据集中的最高点或最极端的情况。

$$f_{\max} = \max\ \{C_i | i \in (1, N)\} \tag{4.3}$$

(5) 最小值：也被称作谷值，数据集中所有数值中的最小值。它代表了数据集中的最低点或最极端的相反情况。

$$f_{\min} = \min\ \{C_i | i \in (1, N)\} \tag{4.4}$$

(6) 极差：数据集中最大值和最小值之间的差。它提供了数据集中数值变化的范围，是衡量数据波动大小的一种简单方法。

$$f_{\text{diff}} = f_{\max} - f_{\min} \tag{4.5}$$

(7) 过均值率：数据集中最大值和最小值之间的差。它提供了数据集中数值变化的范围，是衡量数据波动大小的一种简单方法。

$$f_{\text{above}} = \frac{\text{num}\ \{C_i | i \in (1, N), C_i > \mu\}}{N} \tag{4.6}$$

4.1.2 频域特征

频域特征是从信号的频率成分中提取的，它们描述了信号在频率域中的分布特性。这些特征通常通过傅里叶变换（Fourier Transform, FT）或其他频域分析方法获得。对于分割后的活动状态数据片段，除了提取时域特征，还将提取频域特征。由于获得的活动状态数据是离散时域数据，需要对其进行离散傅里叶变换。对于一个传感器某一轴的感知数据，G_i 表示时域数据经过傅里叶变换后得到的第 i 个频域数据如式(4.1.2)：

$$G_i = \sum_{n=1}^{N} C_n \text{exp}\left[-\text{j}\left(2\pi/N\right)^{in}\right],\ i \in (1, N) \tag{4.7}$$

按照如表 2 所示的计算方法，获得某一轴的数据片段对应的均值、标准差、最大值、最小值、偏度和峰度这 6 类特征，其中均值、标准差、最大值和最小值被称为幅度统计特征，与时域特征含义说明一致，在此不加以赘述；偏度和峰度被称为形状统计特征。

(1) 均值

$$\mu_{\text{amp}} = \frac{1}{N}\sum_{i=1}^{N} G_i \tag{4.8}$$

(2) 标准差

$$\sigma_{\text{amp}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(G_i - \mu_{\text{amp}})^2} \tag{4.9}$$

(3) 最大值

$$f_{max_{amp}} = \max\{G_i | i \in (1, N)\} \tag{4.10}$$

(4) 最小值

$$f_{min_{amp}} = \min\{G_i | i \in (1, N)\} \tag{4.11}$$

(5) 偏度：衡量数据分布不对称性的统计量。如果数据分布的尾部向右延伸（正偏态），则偏度为正值；如果尾部向左延伸（负偏态），则偏度为负值。偏度的绝对值越大，表示数据分布的不对称性越强。

$$\gamma_{\text{ske}} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{G(i) - \mu_{amp}}{\sigma_{amp}}\right]^3 \tag{4.12}$$

(6) 峰度：衡量数据分布不对称性的统计量。如果数据分布的尾部向右延伸（正偏态），则偏度为正值；如果尾部向左延伸（负偏态），则偏度为负值。偏度的绝对值越大，表示数据分布的不对称性越强。

$$\gamma_{\text{kur}} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{G_i - \mu_{\text{amp}}}{\sigma_{\text{amp}}}\right]^4 - 3 \tag{4.13}$$

对于数据片段中某一轴的感知数据，可以提取 7 种时域特征和 6 种频域特征. 由于一个数据片段包含 9 个轴的感知数据，可以提取 $(7+6)\times 9 = 117$ 种时域和频域特征，形成 117 维的特征向量。

4.2 基于互信息的特征选择

互信息（Mutual Information, MI）是一种统计量，用于度量两个变量之间的相互依赖性。在特征工程中，互信息可以用来评估特征与目标变量之间的相关性，从而帮助我们选择最有信息量的特征，构建更好的预测模型互信息最大化 (Mutual Information Maximization, MIM) 是最原始的基于信息度量的特征选择算法，其准则函数如式(4.14)，为后续算法的生成奠定了基础：

$$I(X;Y) = \sum_{x,y} p(x,y) \times \log \frac{p(x,y)}{p(x) \times p(y)} \tag{4.14}$$

其中，X, Y 为两个随机变量，联合分布为 $p(x,y)$，边缘分布分别为 $p(x), p(y)$。$I(X;Y)$ 表明随机变量 X, Y 之间的互信息，互信息 $I(X;Y)$ 的值越大, 说明两者之间互相依赖程度越高。MIM 算法计算每个特征和类标签的互信息函数, 并按它们的值从大到小排序, 然后选择所需的特征数。

4.3 基于 K-means 的特征聚类

K-means 算法是一种迭代求解的聚类分析算法，其核心思想是将数据集中的 n 个对象划分为 K 个聚类，使得每个对象到其所属聚类的中心（或称为均值点、质心）的距离之和最小。

(1) 初始化：选择 K 个初始聚类中心在算法开始时，需要随机选择 K 个数据点作为初始的聚类中心。这些初始聚类中心的选择对最终的聚类结果有一定的影响，因此在实际应用中，通常会采用一些启发式的方法来选择较好的初始聚类中心，如 K-means++ 算法。

(2) 分配：将每个数据点分配給最近的聚类中心对于数据集中的每个数据点，计算其与每个聚类中心的距离，并将其分配给距离最近的聚类中心。这一步通常使用欧距离作为距离度量，计算公式如式(4.15)：

$$\text{dist}\,(x, c_i) = \sqrt{\sum_{j=1}^{d} (x_j - c_{ij})^2} \tag{4.15}$$

其中，x 是数据点，c_i 是第 i 个聚类中心，d 是数据的维度，x_j 和 c_{ij} 分别是 x 和 c_i 在第 j 维上的值。

(3) 更新：对于每个聚类，重新计算其聚类中心。新的聚类中心是该聚类内所有数据点的均值，计算公式如式(4.16)：

$$c_i = \frac{1}{|S_i|} \sum_{x \in S_i} x \tag{4.16}$$

其中，S_i 是第 i 个聚类的数据点集合，$|S_i|$ 是该集合中数据点的数量。

(4) 迭代：重复执行分配和更新步骤，直到满足某种终止条件，当满足终止条件时，算法停止迭代并输出最终的聚类结果。

4.4 问题求解

综上，首先对加速度，角速度，姿态角三类数据在 X, Y, Z 三个轴向上的 9 种初始数据在均值，标准差，偏度，峰度等 13 种时域和频域特征进行提取，共获取 117 种特征，然后利用互信息方法对这 117 种特征进行选择，选取显著性更强的一些特征作为聚类初始数据进行 Kmeans 聚类，最后，得到了 3 名实验人员的活动状态分类情况如表 (4.1)：

表 4.1　问题 1 结果 (活动状态数据编号省去 SY)

分类	Person1	Person2	Person3
第 1 类	4, 5, 10, 13, 33	33, 48	16, 33, 37, 48, 59
第 2 类	7, 11, 19, 38, 39, 46	11, 16, 17, 24, 27, 31, 39, 41, 53, 58	1, 7, 8, 9, 13, 14, 19, 21, 27, 32, 35, 40, 41, 56, 58
第 3 类	18, 27, 41, 45, 52, 55, 59	3, 15, 52	10, 15, 53, 54, 55
第 4 类	15, 26, 28, 29, 35	1, 34, 43, 44, 46	23, 26, 51
第 5 类	24, 40, 56	2, 4, 12, 19, 42	4, 12, 28, 42, 43
第 6 类	14, 22, 49	29, 45, 50, 54, 56	5, 20, 22, 47, 52
第 7 类	20, 25, 31, 43, 44, 48	6, 7, 9, 10, 18, 36, 51, 59	2, 36, 45, 49
第 8 类	1, 12, 16, 30	5, 20, 21, 28, 37	11, 17, 29, 30
第 9 类	3, 8, 23, 47, 50	25, 26, 55	34
第 10 类	2, 17, 3, 51, 53, 57, 60	8, 14, 22, 35, 38	6, 24, 31, 38, 60
第 11 类	21, 32, 366, 37, 54, 58	13, 23, 32, 47, 57	3, 18, 25, 39, 44, 57
第 12 类	6, 9, 42	30, 40, 49, 60	46, 50

5 问题 2 建模与求解

针对问题二中的第 1 小问，首先对附件 2 中的数据按照数据预处理中的方法进行数据野值修补，卡尔曼滤波，姿态解算和重力去除，然后对数据进行时域特征和频域特征的提取，最后基于互信息进行特征选择，形成初始数据集。

5.1 单一模型建立

(1) 随机森林。

随机森林（Random Forest, RF）是一种基于决策树的集成学习方法。它通过构建多棵决策树并结合其输出结果来提高模型的准确性和稳定性。随机森林的基本原理。随机森林的基本原理如算法 (5.1)：

Algorithm 5.1 随机森林算法

Step 1: 从原始数据中以有放回的方式随机取样得到 n 个训练数据集；

Step 2: 从每个训练数据集中随机选择 K 个特征（K 小于原始数据全部的特征）；

Step 3: 反复根据这 K 个特征建立起来 m 棵决策树；

Step 4: 应用每个决策树来预测结果，并且保存所有预测的结果；

Step 5: 对分类模型进行投票，计算每个预测结果的得票数，选择得票数最高的模型。

(2) 支持向量机。

支持向量机（Support Vector Machine, SVM）是一种保持经验风险的值恒定不变，使得可信区间最小化的方法。其目标是通过数据空间中的超平面来实现两类数据的分离。一旦确定了超平面，分类过程就变得相对简单，即判断数据点位于超平面的哪一侧。在存在多个满足条件的超平面上，SVM 会选择支持向量之间的距离较大的超平面，这个超平面应确保其与最近支持向量的距离最小，同时每个类仅有一个支持向量。

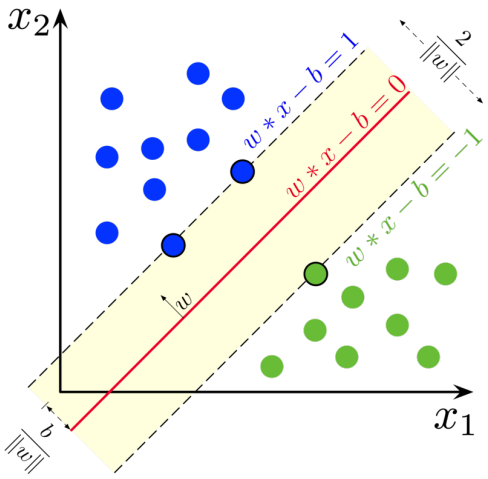


图 5.1　支持向量机原理

SVM 的核心旨在求出向量 $\vec{w}$ 和 $\vec{b}$，使得超平面 $w^T\vec{x}+\vec{b}=0$ 成为“最大边缘超平面”，即该超平面与最近数据点 $\vec{x_i}$ 之间的距离被最大化。若强制所有数据点都满足条件 $\left|w^T\vec{x}+\vec{b}\right| \geq 1$，那么优化问题就等同于寻找最小化向量 $||w||$。

(3) K 近邻

K 近邻（K Nearest Neighbors, KNN）是一种基于实例的学习方法，通过计算样本与训练集中各点的距离，选择距离最近的 K 个邻居进行分类。K 近邻分类器的基本原理如算法 (5.2)：

Algorithm 5.2 K 近邻算法

Step 1: 计算预测数据样本与各个训练数据样本之间的距离；

Step 2: 按照距离的递增关系进行排序；

Step 3: 选取距离最小的 K 个数据样本（前面 K 个）；

Step 4: 确定前 K 个数据样本所在类别的出现频率；

Step 5: 前 K 个数据样本中出现频率最高的类别作为预测数据样本的类别。

(4) 梯度提升

梯度提升（Gradient Lift, GL）是一种集成学习方法，通过逐步添加弱分类器，每一步都对前一步的错误进行修正，从而提高模型的预测性能。其核心思想是通过加法模型和前向分步算法，优化目标函数。

梯度提升分类器是一种集成学习方法，通过逐步添加弱分类器，每一步都对前一步的错误进行修正，从而提高模型的预测性能。其核心思想是通过加法模型和前向分步算法，优化目标函数。假设弱分类器为 $h_m(x)$，则第 m 步的模型为：

$$F_m(x) = F_{m-1}(x) + \nu\, h_m(x) \tag{5.1}$$

其中，$F_{m-1}(x)$ 是前 $m-1$ 步的模型，ν 是学习率。每一步都通过优化损失函数 $L(y, F(x))$ 的负梯度来选择 $h_m(x)$。

(5) 朴素贝叶斯

朴素贝叶斯（Naive Bayes, NB）作为一种基于概率的简洁模型，常被应用于解决垃圾邮件分类、文字分类等各种问题。朴素贝叶斯分类模型基本思想是在对由特征向量 $x = (x_1, x_2, \cdots, x_n)$ 表示的新实例进行分类时，考虑每个可能类 $c_k\,(k = 1, 2, \cdots, K)$ 的后验概率，根据贝叶斯定理，后验概率由下式表示：

$$P(c_k|\, x) = \frac{P(c_k)\, P(x|\, c_k)}{P(x)} \tag{5.2}$$

对式(5.2)进一步改写为：

$$P(c_k|\, x_1, x_2, \cdots, x_n) = \frac{P(c_k)\, P(x_1, x_2, \cdots, x_n|\, c_k)}{P(x_1, x_2, \cdots, x_n)} \tag{5.3}$$

假设所有特征之间相互独立，那式(5.3)可进一步写为：

$$\begin{aligned} P(c_k|\, x_1, x_2, \cdots, x_n) &= \frac{P(c_k)\, P(x_1|\, c_k)\, P(x_2|\, c_k)\, \cdots P(x_n|\, c_k)}{P(x_1, x_2, \cdots, x_n)} \\ &= \frac{P(c_k) \prod_{i=1}^{n} P(x_i|\, c_k)}{P(x_1, x_2, \cdots, x_n)} \end{aligned} \tag{5.4}$$

在实际计算过程中，由于贝叶斯公式的分母代表特征向量 x 的证据概率，对于各个类别而言，其值均保持不变。在比较不同类别的后验概率时，这一项并不会对最终结果产生影响。因此，可以在计算过程中忽略这一项。那么，可以得出类别为

$$y^* = \underset{c_k}{argmax}\; P(c_k) \prod_{i=1}^{n} P(x_i|\, c_k) \tag{5.5}$$

(6) 决策树

决策树（Decision Tree，DT）是一种递归划分特征空间的分类方法，通过选择最优特征及其划分点，构建树形结构进行分类。常用的划分标准包括信息增益、基尼系数等，算法流程如算法 (5.3)。

其中，ΔI 是信息增益或基尼系数的增量。

(7) 逻辑回归

逻辑回归（Logistic Regression，LR）是一种线性模型，通过对线性回归模型的输出应用逻辑函数，将其映射到 0 到 1 之间，从而实现分类。逻辑回归常用于二类分类，通过扩展可用于多类分类。

Algorithm 5.3 决策树算法

Step 1: 选择最优特征 f 及其划分点 t：$(f,t)=\arg\max_{f,t}\Delta I$；

Step 2: 根据最优特征及划分点将数据集划分为两个子集，满足

$$D_1=\{(x,y)\mid x_f\le t\},\quad D_2=\{(x,y)\mid x_f>t\}$$

Step 3: 对子集 D_1 和 D_2 递归构建子树，直到满足停止条件（如节点纯度或树的最大深度）。

对于二类分类问题，假设训练集为 $\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$，其中 $y_i\in\{0,1\}$。逻辑回归模型为：

$$P(y=1\mid x)=\sigma(w\cdot x+b)$$

$$\sigma(z)=\frac{1}{1+e^{-z}}$$

其中，w 和 b 是模型参数，$\sigma(z)$ 是逻辑函数。对于多类分类，通常使用 softmax 函数：

$$P(y=c\mid x)=\frac{e^{w_c\cdot x+b_c}}{\sum_j e^{w_j\cdot x+b_j}} \tag{5.6}$$

(8) 多层感知机

多层感知机（Multilayer Perceptron, MLP）是一种线性模型，通过对线性回归模型的输出应用逻辑函数，将其映射到 0 到 1 之间，从而实现分类。逻辑回归常用于二类分类，通过扩展可用于多类分类。

多层感知机是一种前馈神经网络，由输入层、一个或多个隐藏层和输出层组成。每层由若干神经元组成，通过非线性激活函数连接，能够表示复杂的非线性关系。给定输入 x，MLP 的输出为：

$$h_1=\sigma(W_1x+b_1)$$

$$h_2=\sigma(W_2h_1+b_2)$$

$$\vdots$$

$$\hat{y}=\sigma(W_Lh_{L-1}+b_L)$$

其中，W_i 和 b_i 是第 i 层的权重和偏置，σ 是激活函数，如 ReLU 或 sigmoid。

5.2 基于 Stacking 集成学习建立活动状态判别模型

Stacking 方法通过元学习器（次级分类器）进行泛化，综合降低偏差以提高模型的分类能力。Stacking 方法的基本原理为从一系列模型中选择多个作为初级分类器，并以原始数据集为基础对这些初级分类器进行训练，输出一组新的分类数据。初级分类器的

输出是次级分类器的输入特征，同时由初级分类器产生的新数据集的标签仍使用初始数据的标签

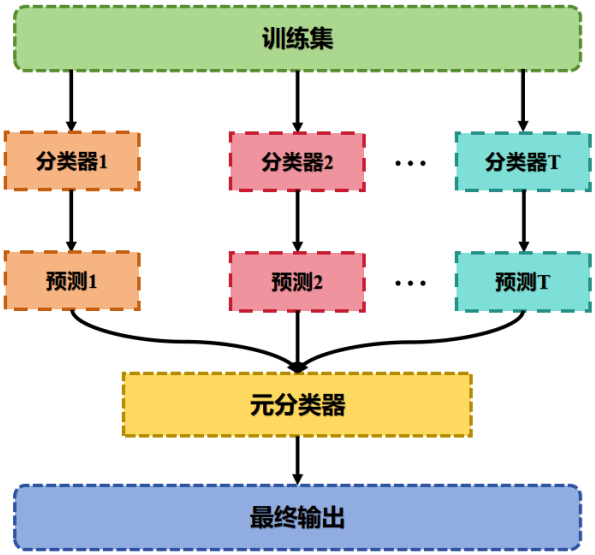


图 5.2　Stacking 算法的基本原理

为避免严重的过拟合风险，在 Stacking 方法训练过程中，使用 5 折交叉验证算法产生初级分类器的训练数据。这种交叉验证与模型评估中的交叉验证的目的不一样，Stacking 方法需要将原始数据通过一定的方式划分为一组训练集 D_{train} 和测试集 D_{test}，接着对训练集 D_{train} 用交叉验证算法产生次级分类器的训练集，最后在测试集 D_{test} 上计算评价指标来评估模型的性能。Stacking 方法通过精妙地改进交叉验证算法的方式来避免模型发生过拟合，增强模型的稳定性。

5.3 问题求解

5.3.1 K-means 聚类结果评估

将附件 2 中的数据经过处理后，采用问题 1 建立的 K-means 聚类模型进行聚类，其聚类结果如表 (5.1)：

通过表 (5.1) 的聚类结果进行计算，得出采用 Kmeans 聚类方法进行分类的正确率为 64.5%，分类效果表现较好。

5.3.2 单一模型训练与评估

对数据按照 80% 的训练集、20% 的测试集进行划分，再分别对提出的 8 种单一分类器进行 5 折交叉验证进行训练，其测试结果如表 (5.2)：

通过对表 4 进行分析，随机森林模型表现最优，其准确率、精确率、召回率和 F1-score 均达到了 0.90。其次是 SVM 模型，尽管稍逊于随机森林模型，但其各项指标均在

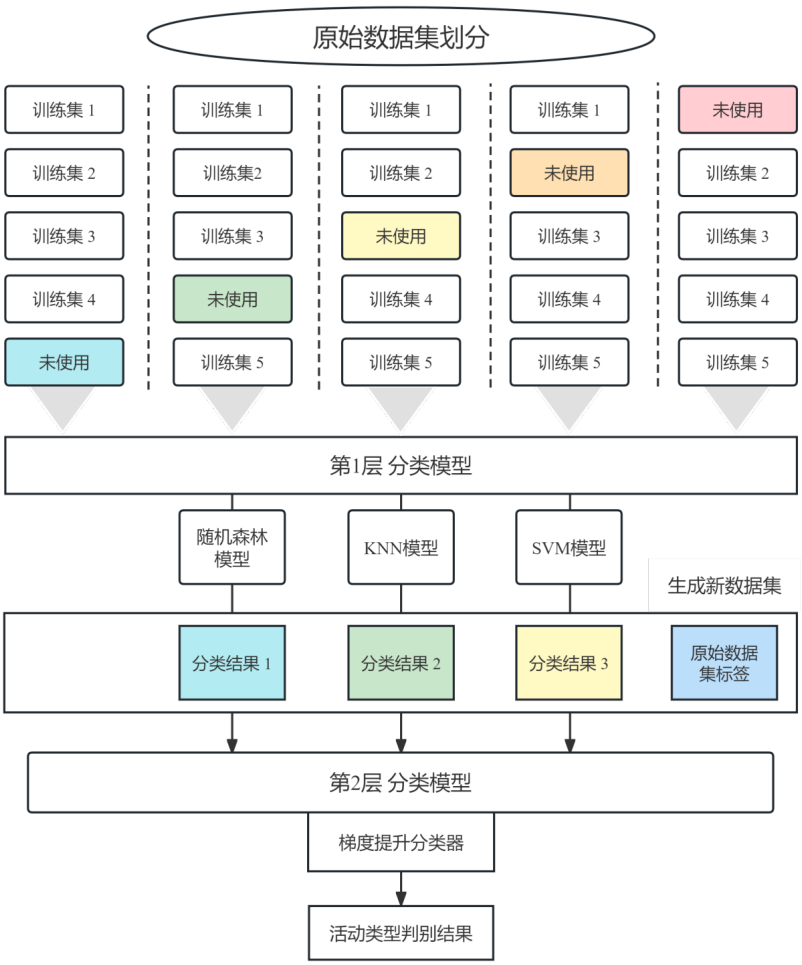


图 5.3　Stacking 算法详细原理

0.85 左右，表现依然优异。KNN 和梯度提升分类模型表现较为稳定，四项指标均在 0.81 至 0.83 之间。多层感知机模型的性能稍逊，指标约为 0.78。决策树和逻辑回归模型的表现中等，指标均在 0.73 至 0.75 之间。相较之下，朴素贝叶斯模型的表现最差，各项指标均在 0.65 至 0.67 之间。

5.3.3 基于 Stacking 集成学习的活动状态判别模型的训练与评估

根据单一模型的评估结果，选取分类效果最好的随机森林模型、SVM 模型、KNN 模型和梯度提升分类模型构建 Stacking 组合模型，以随机森林模型、SVM 模型、KNN 模型为初级分类器，梯度提升分类模型为次级分类器的组合模型分类效果最好，其在测试集上的表现如表 (5.3)：

通过对表 (5.3) 的模型评估结果进行分析，模型在向前行走、步行上楼以及向前奔跑等动作的分类中表现优异，准确率、召回率和 f1-score 均达到 1.00，充分体现了模型在这些动作分类任务中的高效性和准确性。在向左行走和跳跃动作的分类中，模型同样表现出色，该分类模型的整体分类正确率达到了 92.67%。

表 5.1　聚类结果

	1	2	3	4	5	6	7	8	9	10	11	12
1	25	21	4									
2	9	40		1								
3	11	3	32	1				1	1			
4	8	3	17	46					6			
5	5	15		11	18		2		5			
6		5		1		26	9					
7				27			21		2			
8								32	4	8		10
9	1	1						22	22			4
10	4							10		45		
11									4		32	29
12									2			48

表 5.2　分类器效果统计

分类器名称	正确率	精确率	召回率	f1-score
随机森林	0.88	0.90	0.90	0.90
支持向量机	0.78	0.76	0.81	0.77
k 近邻	0.72	0.75	0.73	0.72
梯度提升	0.82	0.83	0.81	0.82
朴素贝叶斯	0.67	0.66	0.66	0.65
决策树	0.75	0.75	0.74	0.73
逻辑回归	0.81	0.82	0.84	0.82
多层感知机	0.83	0.85	0.87	0.85

5.3.4 活动状态预测

对附件 3 中的数据按照上述方法进行处理，利用上文训练的 Stacking 集成学习模型对该名实验人员的 30 组活动状态数据进行活动状态类型判别，得到判别结果如表 (5.4)：

表 5.3　基于 Stacking 集成学习的活动状态判别模型的评估结果

状态编号	准确率	召回率	f1-score	状态编号	准确率	召回率	f1-score
1	1.00	1.00	1.00	7	0.90	1.00	0.95
2	0.86	0.86	0.86	8	0.83	0.83	0.83
3	0.90	1.00	0.95	9	0.93	0.93	0.93
4	1.00	1.00	1.00	10	0.92	1.00	0.96
5	1.00	0.75	0.86	11	0.66	0.69	0.71
6	1.00	1.00	1.00	12	0.83	0.67	0.74

表 5.4　问题 2.2 结果

活动类型	判别状态	活动类型	判别状态	活动类型	判别状态
SY1	5	SY11	8	SY21	2
SY2	10	SY12	1	SY22	6
SY3	8	SY13	8	SY23	7
SY4	7	SY14	6	SY24	5
SY5	3	SY15	2	SY25	11
SY6	3	SY16	10	SY26	7
SY7	3	SY17	2	SY27	10
SY8	1	SY18	2	SY28	2
SY9	4	SY19	9	SY29	6
SY10	5	SY20	8	SY30	7

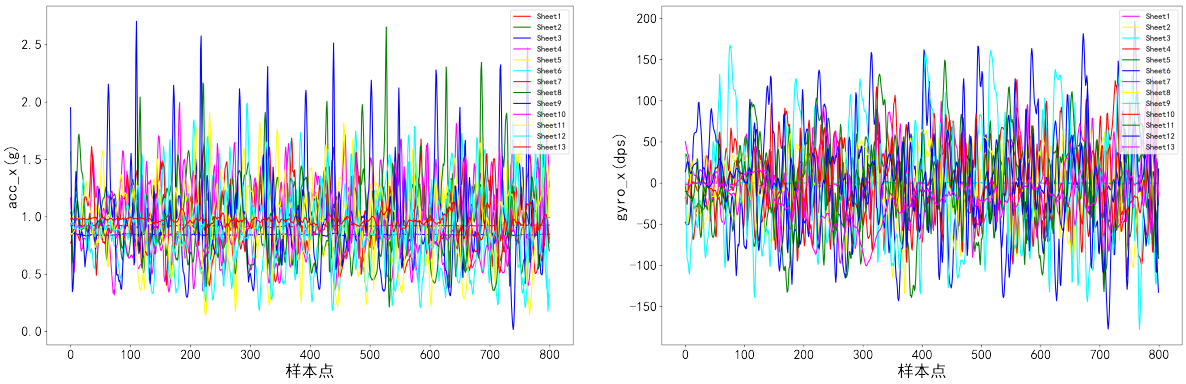
6 问题 3 建模与求解

6.1 活动状态差异性分析

结合前文的预测结果，对 13 位实验人员统一活动进行对比分析，用 acc_x，acc_y，acc_z，gyro_x，gyro_y，gyro_z，6 组原始数据进行对比分析，每个实验人员的数据集大小不一样，本文采取相同的数据大小进行分析，绘制各原始数据变化趋势如图 (6.1)：

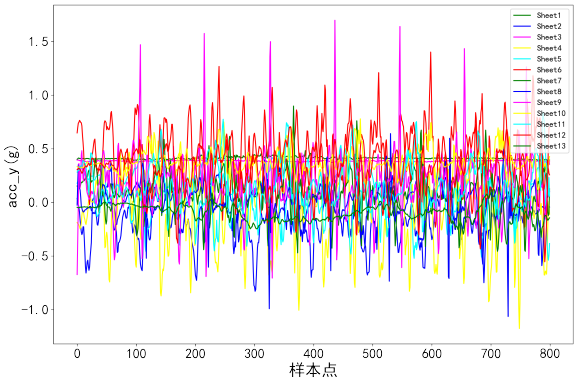
从图 (6.2) 可以发现，不同人员的同一种运动状态存在明显的差异性，

针对加速度数据 acc_x(g) 的标准差较大人员为 3, 10, 13，acc_y(g) 标准差较大人员为 2, 12, 13，acc_z(g) 标准差较大人员为 2, 11, 13。针对陀螺仪数据 gyro_x(dps) 标准差较大为人员 12, 13, 9，gyro_y(dps) 标准差较大人员为 12, 13, 9，gyro_z(dps) 的为标准差较大：人员 12, 13, 9。总体来看，人员 12 和 13 在各个传感器数据变量中的波动较大，尤其是陀螺仪数据。人员 2 在加速度数据中的波动较大。

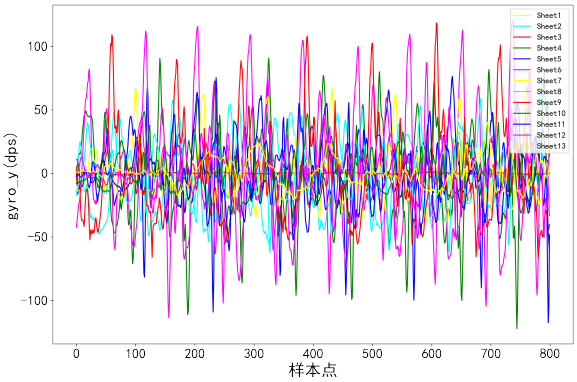


(a) acc_x(g)

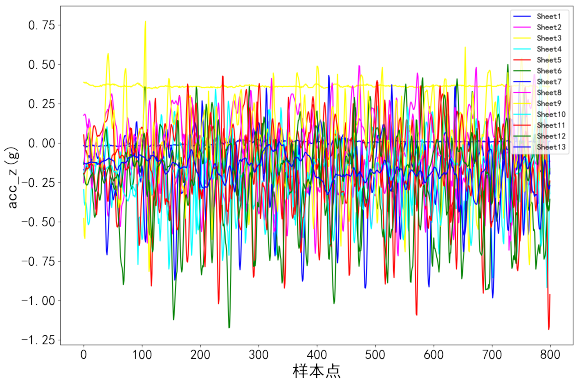
(b) gyro_x(g)



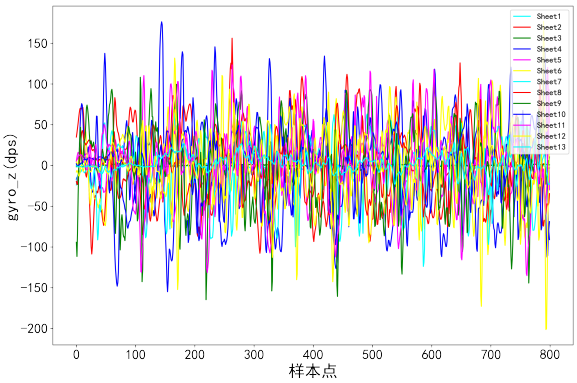
(c) acc_y(g)



(d) gyro_y(g)



(e) acc_z(g)

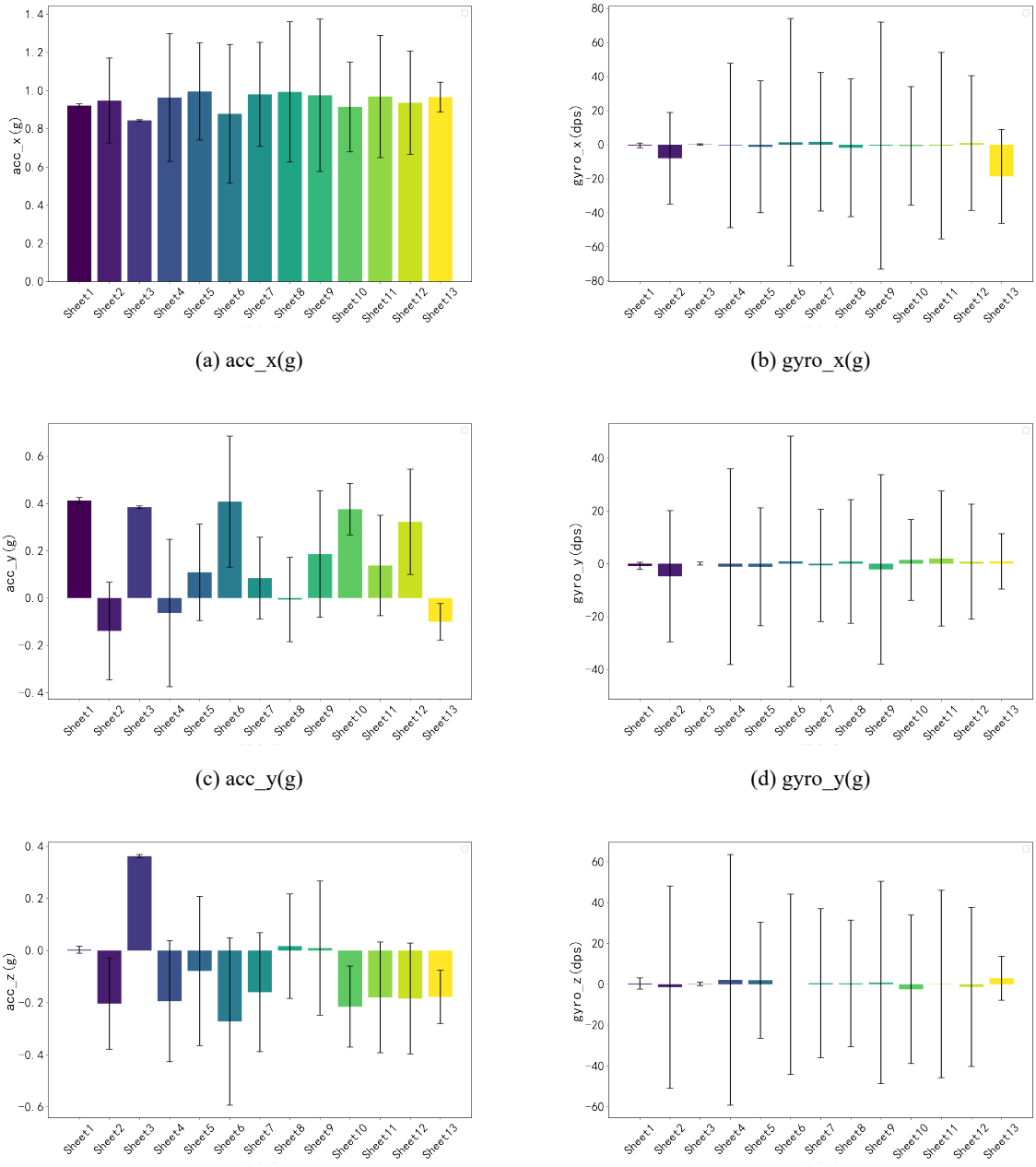


(f) gyro_z(g)

图 6.1　原始数据变化趋势

进一步利用 PCA 方法进行数据降维，同时进行 K-means 聚类并获取聚类中心点，使用 Seaborn 绘制 PCA 和聚类结果的散点图，如图 (6.3)：

从聚类结果及权重分析可以看出，不同人员在各个聚类中的分布比例不同。Cluster 1 和 Cluster 7 的人员主要来自实验人员 1 和实验人员 2，表明这些人员在这些聚类中占据较大比例。Cluster 6 和 Cluster 11 中的主要人员来自实验人员 6 和实验人员 9，表明这些人员在这些聚类中占据较大比例。每个聚类中包含的人员数量和比例不同，这表明不同人员在同一活动状态下的行为和数据特征存在差异。Cluster 1 中的主要人员是来自



(a) acc_x(g)　(b) gyro_x(g)

(c) acc_y(g)　(d) gyro_y(g)

(e) acc_z(g)　(f) gyro_z(g)

图 6.2　原始数据在不同人员中的均值和标准差

实验人员 1 和实验人员 3，而 Cluster 6 中的主要人员则是来自实验人员 6 和实验人员 9。这说明不同人员在同一活动状态下的传感器数据有明显差异，导致他们被分到不同的聚类中。通过分析不同人员在同一活动状态下的标准差，可以进一步确认数据的变异性。某些人员的加速度或陀螺仪数据标准差较大，说明他们的运动幅度和方向变化较多，表现出更大的行为差异。总体来看，聚类结果清晰地展示了不同人员在同一活动状态下的行为差异。尽管他们参与的是相同的活动，但各自的传感器数据特征可能由于运动习惯、身体特征等原因而不同，从而导致他们被分到不同的聚类中。这些差异在聚类中心

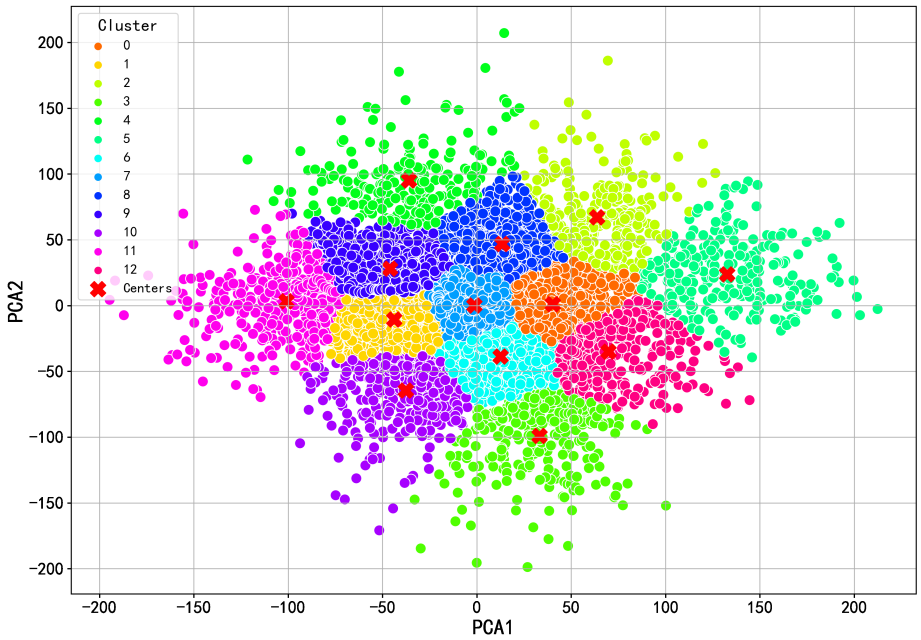


图 6.3　13 名实验人员带聚类中心的 K-means 聚类（PCA 处理后）

的分布和每个聚类中的人员构成上表现得非常明显。

6.2 活动状态相关性分析

针对活动状态数据与实验人员的年龄、身高、体重相关性分析，本文构建多元线性回归模型如式(6.1)：

$$y = X\beta + \epsilon \tag{6.1}$$

其中，y 是因变量矩阵（传感器数据），X 是自变量矩阵（年龄、体重和身高），β 是回归系数矩阵，ϵ 是误差项。

试图找到每个传感器数据变量（如 acc_x(g)、gyro_y(dps) 等）与实验人员属性（年龄、体重、身高）之间的线性关系，构建热力图如图 (6.4)，其中热力图的每个单元格表示特定传感器数据变量与特定实验人员属性之间的回归系数。

通过多元线性回归分析和热力图可视化，我们能够直观地看到实验人员的年龄、体重和身高对不同传感器数据的影响程度，根据数据，传感器特征与年龄、体重、身高之间的回归系数分析如下，可以发现不同传感器数据对个体属性的敏感度不同。

(1) acc_x(g): 身高对加速度计 x 轴影响最大 (0.004195)。

(2) acc_y(g): 年龄对加速度计 y 轴影响最大 (0.004481)。

(3) acc_z(g): 体重对加速度计 z 轴影响最大 (0.010427)。

(4) gyro_x(dps): 体重对陀螺仪 x 轴影响最大 (-0.251259)。

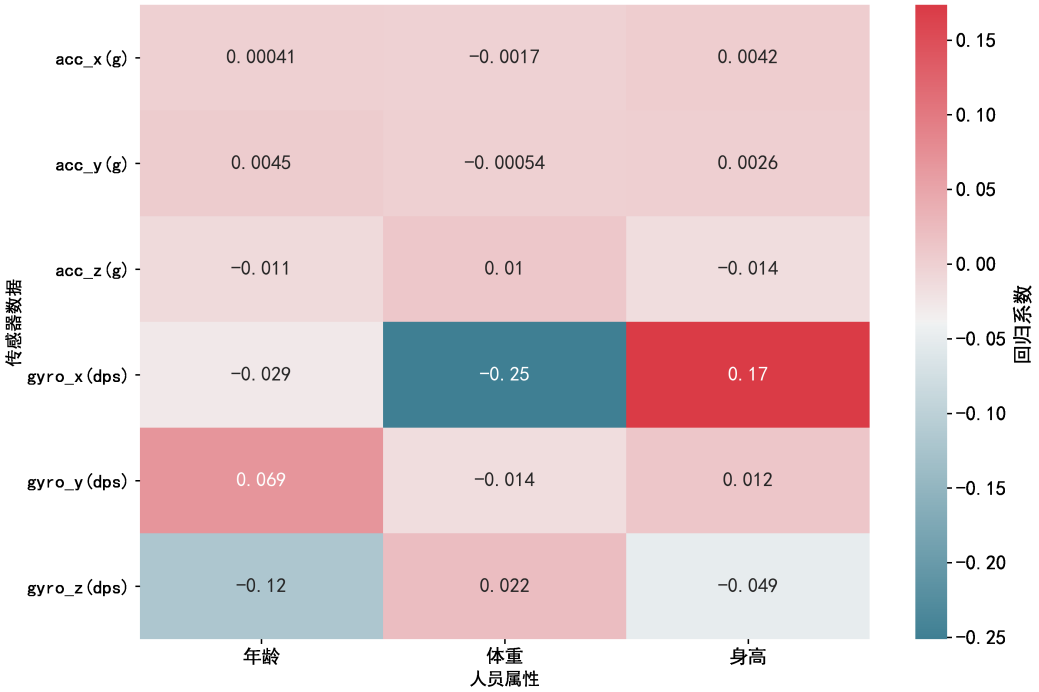


图 6.4　传感器数据与年龄、体重、身高的回归系数

(5) gyro_y(dps): 年龄对陀螺仪 y 轴影响最大 (0.068752)。

(6) gyro_z(dps): 年龄对陀螺仪 z 轴影响最大 (-0.119280)。

随着特征选择方法不断出现，有一系列用于对数据进行特征选择的方法。大多数情况下，最常用的特征选择方法属于过滤法。另一方面，由于资源的大量计算消耗和过度拟合的高风险，包装器方法在文献中被大量避免。尽管嵌入式模型在数学建模的初期没有得到足够的重视，但近年来也出现了很多方法，不同特征选择方法的优缺点如表 (6.1)。值得注意的是，对最新文献的回顾显示无论集成方法在一些数据集上都取得了不错的效果。

表 6.1　不同特征选择方法的优缺点

名称	优点	缺点
过滤式	快速，可扩展的；独立于分类器；泛化性能好	分类进度一般，不能完全去冗余
包裹式	简单；与分类器交互	针对特定分类器；易过拟合
嵌入式	与分类器交互；复杂性低，不易过拟合；性能更精确	针对特定分类器
集成式	不易过拟合；对高维数据更加灵活稳定	计算效率低

由于本题的运动数据为多样本数据，基于表 (6.1) 的考量，在定义数据特征的基础上，我们最终建立了集成特征筛选模型，以期望不同的模型能够互补劣势。

6.2.1 基于随机森林的特征筛选模型

随机森林（Random forest，RF）作为一种新兴起、高度灵活的机器学习算法，其拥有广泛的应用前景，不仅在问题 2 中作为集成分类器在大分类以及回归问题中具有极好的准确率，而且自带特征筛选机制，即随机森林能够评估各个特征在相应问题上的重要性。

在问题 3 中，针对分子描述符变量数量多、高度非线性、耦合度高的特点，选择嵌入式特征选择方法随机森林进行数学建模，以期望得到前 10 个对运动状态最具有显著影响的分子描述符变量。

随机森林算法可以在分类的基础上进行回归分析, 通过将样本分类的结果进行一定的运算可以获得各个特征重要性特征的重要性表示特征对预测结果影响程度, 某一特征重要性越大, 表明该特征对预测结果的影响越大, 重要性越小, 表明该特征对预测结果越小。随机森林算法中某一特征的重要性, 是该特征在内部所有决策树重要性的平均值, 而在决策树中, 计算某一个节点 k 的重要性:

$$n_k = \omega_k * G_k - \omega_{\text{left}} * G_{\text{left}} - \omega_{\text{right}} * G_{\text{right}} \tag{6.2}$$

其中，$\omega_k, \omega_{\text{left}}, \omega_{\text{right}}$ 分别为节点 k 以及其左右节点中训练样本与总训练样本数目的比例, $G_k, G_{\text{left}}, G_{\text{right}}$ 分别为节点 k 以及其左右子节点的不纯度。知道每一个节点的重要性之后, 即通过公式的得出某一特征的重要性。

$$f = \frac{\sum_{j\in \text{ nodes split on feature } i} n_j}{\sum_{k\in \text{ all nodes}} n_k} \tag{6.3}$$

6.2.2 基于 Pearson 相关系数的特征筛选模型

传统的相关系数常用来衡量两个数据集合是否在一条线上面，即衡量定距变量间的线性关系，这里，为了衡量变量与目标变量的线性关系，我们使用上节中的 Spearman 相关系数计算，特征变量的重要性由该特征与因变量间的相关系数反应，相关系数越大，则该特征的重要性越高。由于本题中各变量间可能存在非线性关系，传统相关性分析并不足够，为此我们选择距离相关系数（Distance Correlation）作为衡量变量间非线性相关性的指标。Szekely 等人 [7] 定义了距离相关系数的概念, 克服了传统的 Pearson 相关系数的缺点, 可以衡量非线性相关变量之间的关系。在某些情况下, 即便 Pearson 相关系数为 θ , 也无法将两个变量判定为线性无关（有可能非线性相关），但是，如果距离相关系数为 0 , 则可将两个变量判定为互相独立。两个随机向量 $x \in R^P, y \in R^q$ 为观察到的随机样本, x, y 间的距离相关系数 (dCor) 可以定义为:

$$R^2(x,y) = \frac{v^2(x,y)}{\sqrt{v^2(x,x)v^2(y,y)}} \tag{6.4}$$

其中:

$$
\begin{aligned}
v^2(x,y) &= \tfrac{1}{n^2}\sum_{i,j=1}^{n} A_{i,j}B_{i,j} \\
A_{i,j} &= \|x_i - x_j\|_2 - \tfrac{1}{n}\sum_{k=1}^{n}\|x_k - x_j\|_2 - \tfrac{1}{n}\sum_{l=1}^{n}\|x_i - x_l\|_2 + \tfrac{1}{n^2}\sum_{ik,l=1}^{n}\|x_k - x_l\|_2 \\
B_{i,j} &= \|y_i - y_j\|_2 - \tfrac{1}{n}\sum_{k=1}^{n}\|y_k - y_j\|_2 - \tfrac{1}{n}\sum_{l=1}^{n}\|y_i - y_l\|_2 + \tfrac{1}{n^2}\sum_{k,l=1}^{n}\|y_k - y_l\|_2
\end{aligned}
\tag{6.5}
$$

距离相关系数越大，表示两者相关性越强。在统计学中，对相关性强弱有如下约定，如表 (6.2)：

表 6.2　相关性程度度量表

相关性	相关系数
极强相关	0.8 1.0
强相关	0.6 0.8
中相关	0.4 0.6
弱相关	0.2 0.4
极弱或无相关	0 0.2

6.2.3 特征重要性分析

本文利用 data 数据求出的特征性指标，进行一步处理，有 5 种特征重要性分析：

(1) 决策树模型通过特征分裂过程来评估特征的重要性。可以使用 Decision Tree Classifier 或 Decision Tree Regressor 来获得特征的重要性评分；

(2) Random Forest 是集成学习模型，它可以提供更为稳健的特征重要性评分；

(3) Pearson 相关系数可以衡量特征之间的线性关系；

(4) 互信息衡量的是两个变量之间的不确定性减少程度；

(5) 使用 GBDT（Gradient Boosting Decision Tree），GBDT 可以提供各个特征在模型中的分裂度。

综上求解出前 10 个的特征，如表 (6.3)–(6.5)。

gyro_x_dps__mean 在多个方法（决策树、随机森林、互信息、GBDT）中都被认为是最重要的特征，这表明该特征在不同模型中的表现较为一致。pitch_range 在 Pearson 相关系数分析中得分最高，显示了其与分类结果之间较强的线性关系。acc_x_g__range 在互信息分析中得分最高，表明它在减少不确定性方面贡献最大。

6.2.4 特征值主成分分析

将标准化特征数据，使每个特征具有均值为 0 和标准差为 1，使用 PCA 算法对数据进行降维，提取最重要的主成分，展示每个主成分解释的方差比例，已经将标准化后的

表 6.3　特征符号说明 (决策树，随机森林)

决策树		随机森林	
特征	重要性	特征	重要性
gyro_x_dps__mean	0.14230086	gyro_x_dps__mean	0.039652412
gyro_z_dps__std	0.090909091	gyro_z_dps__fft_std	0.017817609
acc_x_g__std	0.080230044	gyro_y_dps__fft_skewness	0.017782602
yaw_fft_kurtosis	0.062467785	acc_x_g__range	0.017497448
gyro_x_dps__fft_max	0.058571337	gyro_y_dps__fft_mean	0.016778261
acc_x_g__above_mean	0.05750437	acc_x_g__std	0.016719651
acc_z_g__max	0.055840198	gyro_z_dps__fft_max	0.016118118
acc_z_g__mode	0.044785888	acc_x_g__max	0.015354499
gyro_y_dps__fft_skewness	0.030052942	gyro_z_dps__fft_mean	0.015278014
yaw_std	0.029853446	gyro_y_dps__fft_max	0.015246494

表 6.4　特征符号说明 (Pearson 相关分析)

Pearson 相关分析	
特征	重要性
pitch_range	0.757278189
yaw_range	0.751011468
yaw_min	0.726649135
yaw_mode	0.726649135
row_std	0.720895799
row_range	0.718745648
yaw_max	0.71722866
row_min	0.701051664
row_mode	0.701051664
pitch_max	0.683787699

运动数据特征数据进行了 PCA 降维，并提取了所有主成分，前五个的主要成分的比例为，PC1: 39.26%，PC2: 18.82%，PC3: 8.19%，PC4: 4.27%，PC5: 3.22%。运动状态数据集 PCA 降维后聚类情况如图 (6.5) 所示，其中的两个轴分别是主成分 1（PC1）和主成分 2（PC2），这两个主成分是从高维数据中提取的，尽量保留了数据的方差。

以下是前五个主成分（PC1 到 PC5）中特征的重要性，图 (6.6) 展示了每个主成分

表 6.5　特征符号说明 (互信息，GBDT)

互信息		GBDT	
特征	重要性	特征	重要性
gyro_x_dps__mean	0.147343209	gyro_x_dps__mean	0.14230086
acc_z_g__range	0.061600016	gyro_z_dps__std	0.090909091
acc_x_g__mean	0.056758416	acc_x_g__std	0.080230044
acc_x_g__std	0.04251098	yaw_fft_kurtosis	0.062467785
acc_z_g__min	0.040332406	gyro_x_dps__fft_max	0.058571337
acc_z_g__mode	0.0364393	acc_x_g__above_mean	0.05750437
acc_x_g__min	0.031845108	acc_z_g__max	0.055840198
pitch_max	0.030797293	acc_z_g__mode	0.044785888
gyro_z_dps__fft_max	0.02874349	gyro_y_dps__fft_skewness	0.030052942
gyro_y_dps__fft_skewness	0.025284285	yaw_std	0.029853446

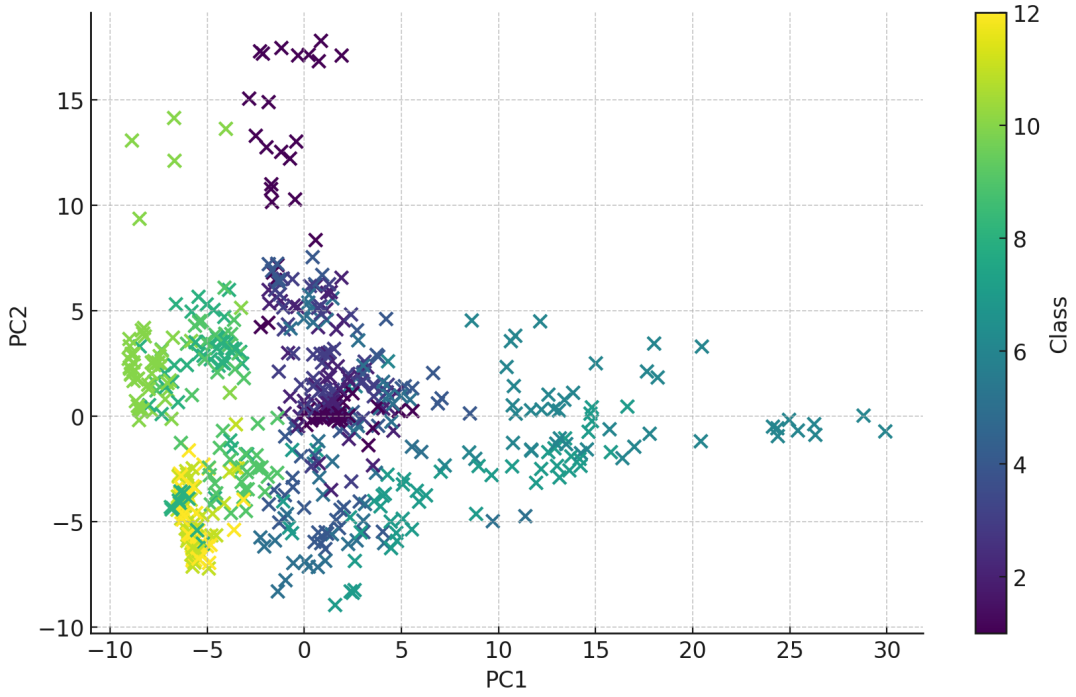


图 6.5　运动状态数据集 PCA 降维后聚类情况散点图

中贡献最大的特征及其绝对载荷值，如下：

PC1 中的主要特征是陀螺仪和加速度计的统计特性，包括标准差、范围、最小值和均值。陀螺仪的 y 轴和 x 轴的标准差和范围对 PC1 的贡献较大，表明陀螺仪在这些轴上的波动对主成分有显著影响。加速度计的 y 轴范围和模式，以及 x 轴范围也对 PC1 有重

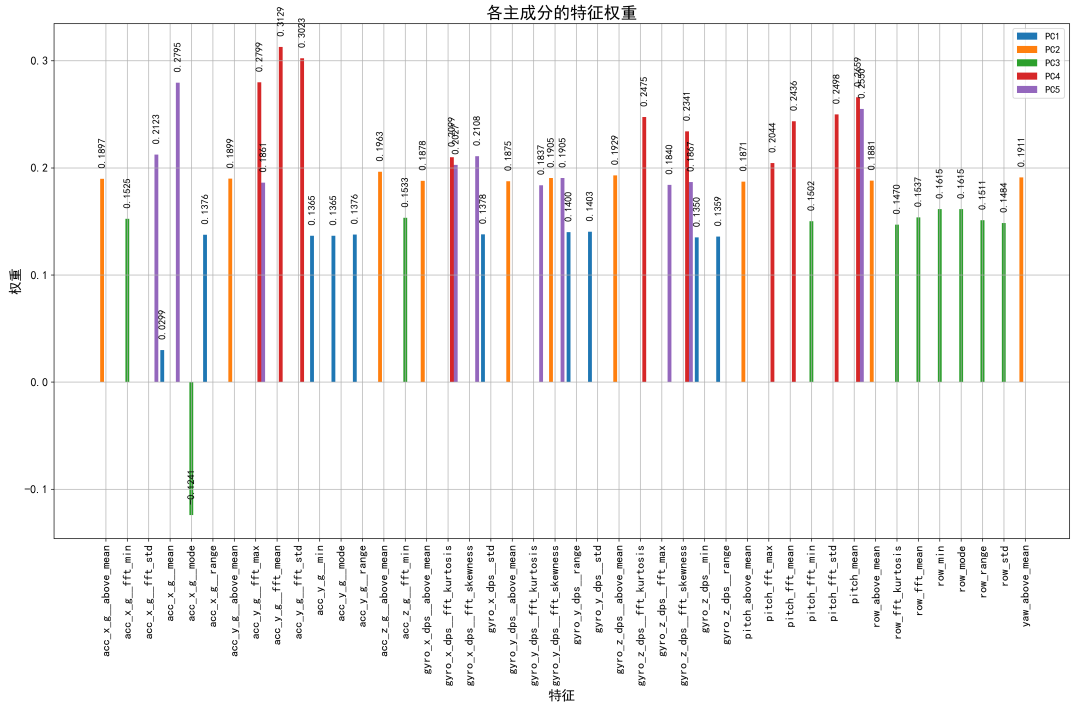


图 6.6　各主成分的特征权重

要贡献。PC2 中的主要特征是各个轴上的加速度计和陀螺仪值超过平均值的次数。加速度计 z 轴和陀螺仪 z 轴的值超过平均值的频率对 PC2 的贡献最大。航向、俯仰和横滚角度超过平均值的频率也对 PC2 有显著影响。

6.2.5 集成特征筛选模型

根据上文对五种方法结果的分析，不同的特征选择方法因为模型方法的不同，所选择出的特征变量也有所不同，这里作出五种方法特征选择结果的相关性热力图，如图 xx 所示，

可以发现，除了两种使用相关性筛选方法的相关性较强以外，大部分具有较弱相关性，利于模型的融合与集成。因此，为了综合这些方法对线性、非线性、高耦合特征的建模能力，基于本题的多样本高维数据，本文建立了集成特征筛选模型，期望获得更加灵活，过拟合风险更低的筛选模型，为了综合五种方法的特征重要性，首先对特征重要性进行归一化，得到特征权值，其反映了变量的重要程度占比，每一个分子描述符的特征权值，表示为该特征的重要性与全体特征重要性之和的比值，其公式如下：

$$y = \frac{y_0 - y_{min}}{y_{max} - y_{min}} \tag{6.6}$$

归一化后，对每种方法的特征权值进行加权融合，即可得到集成的自变量的重要性向量：

$$Y = \sum_{i=1}^{4} \lambda_i y_i \tag{6.7}$$

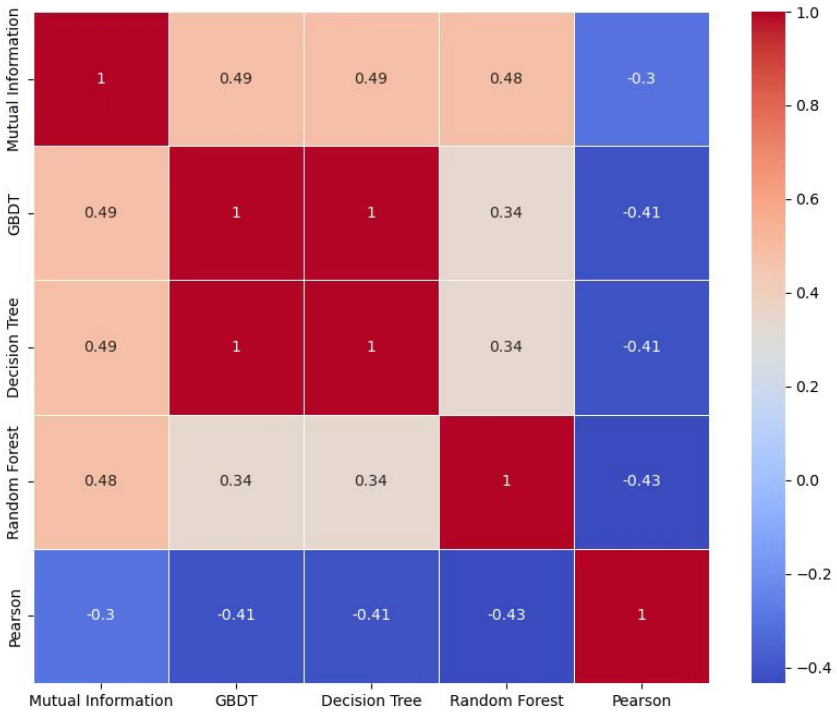


图 6.7　5 种特征选择方法的相关性热力图

其中, Y 为集成四个结果的特征重要性向量, λ_i 为第 i 个方法在融合中所占的权重, 并满足权重和为 1, y_i 为第 i 个方法归一化后的特征权值。

对建立的集成特征筛选模型进行求解，前 10 个特征判断最具有显著影响的运动状态的判断，其信息如表 (6.6)：

当主要特征两两之间相关性均较弱时，某种意义上任何主要变量均可独立的描述因变量的某方面性质此时主要变量具有一定的独立性。我们对提出的分步集成筛选模型的结果: 前 10 个显著特征进行了相关性热力图可视化，见下图。可见，变量之间的独立性较好，满足特征变量独立性的要求。

6.3 案例推理

基于案例的推理 (Case-Based Reasoning，CBR) 是类比推理的一种, 是一种基于经验的智能学习和问题求解方法。案例推理是通过对目标案例与以往历史案例的对比分析和解决方案调整来实现问题解决的，运用案例推理解决问题的一般步骤共包含四个阶段：案例检索（Retrieve）、案例重用（Reuse）、案例修正（Revise）和案例保存（Retain），又被称为“4R”生命周期模型。在遇到一个新问题时，案例推理系统首先将该问题转化为案例表示格式生成目标案例，然后检索案例库中与目标案例相似的源案例。在检索到源

表 6.6　最显著的 10 个特征及其权重

特征	权重
gyro_x_dps__mean	0.749236
acc_x_g__std	0.554963
acc_x_g__max	0.463417
acc_x_g__range	0.441654
acc_z_g__range	0.436993
gyro_z_dps__fft_mean	0.422951
acc_z_g__max	0.411692
gyro_x_dps__fft_max	0.407268
gyro_z_dps__fft_max	0.406632
acc_z_g__min	0.40411

案例后，若源案例解决方案可以解决目标案例问题，则问题得到解决；若源案例解决方案不能解决目标案例问题，那么就要对源案例解决方案进行调整，得到目标案例的解决方案。问题解决完毕后，将本次的目标案例存入案例库中，将案例库动态更新以便下一次问题解决。一个通用的案例推理解决问题的流程图如下图 (6.9)：

考虑 Person 相关系数法以及 DT，RF，GBDT 法等 5 种分析方法，构建集成式特征筛选模型选出重要性排名前 10 名的特征数据，对选出的显著变量使用 Spearman 系数进行独立性验证，得出最佳的特征权重分配方案。利用 Stacking 集成学习算法作为相似案例检索的依据，当输入一个新的运动特征数据时，Stacking 集成学习算法随机对输入数据进行判别，最后通过集成效果来确定最终的分类结果，在第二问的基础上，增加人员的年龄，身高，体重，相关性程度等特征，将人员编号作为目标，利用 Stacking 集成学习进行历史数据的相似度匹配，实现基于案例推理思想的任务刻画。

通过上述的案例推理结果，可以判定活动状态数据与实验人员的年龄、身高、体重具有强相关性。对此，将实验人员的活动状态作为特征输入，以实验人员编号为输出，利用附件 2 中提供的数据对 Stacking 组合模型进行重新训练，最后对附件 5 中的数据进行预测，得到活动状态数据对应人员编号和活动状态编号判别结果如表 (6.7)，问题 3 综合判别结果如表 (6.8)：

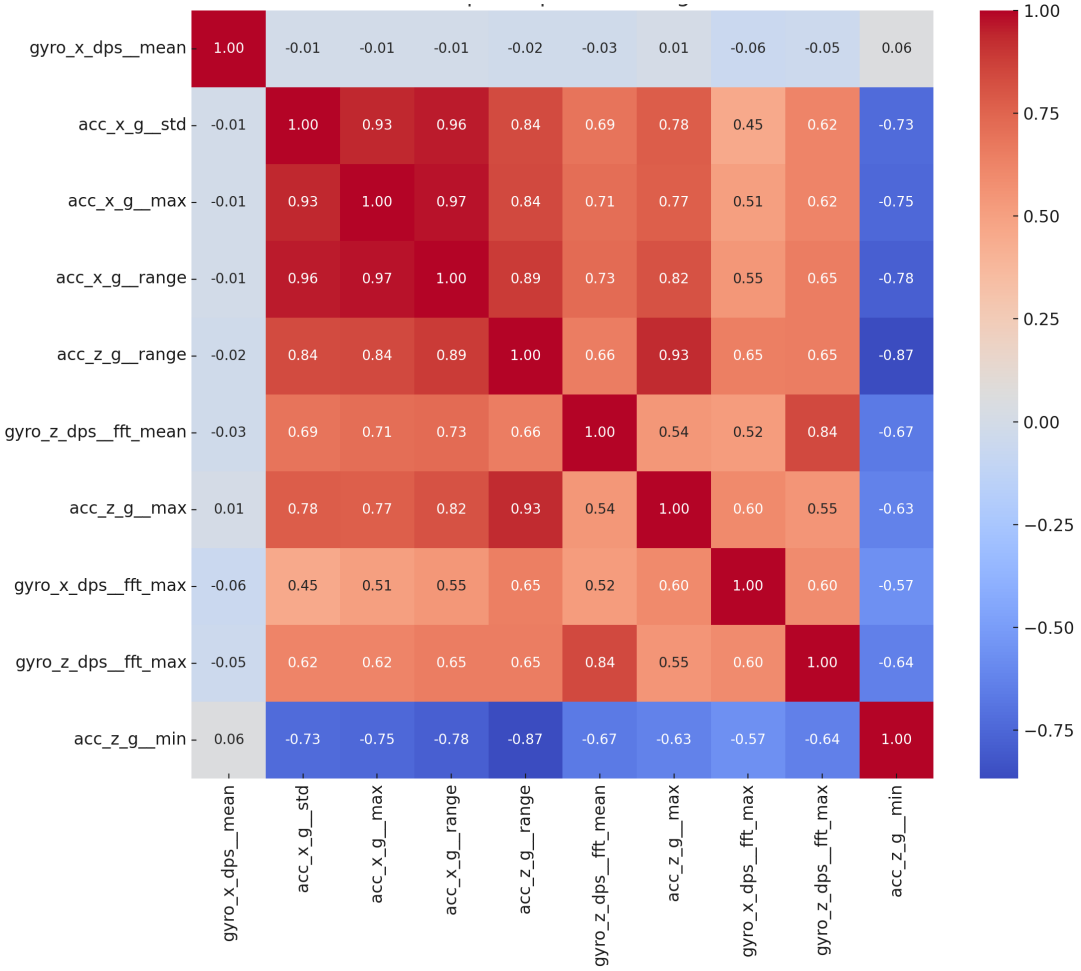


图 6.8　10 类最重要特征相关性热力图

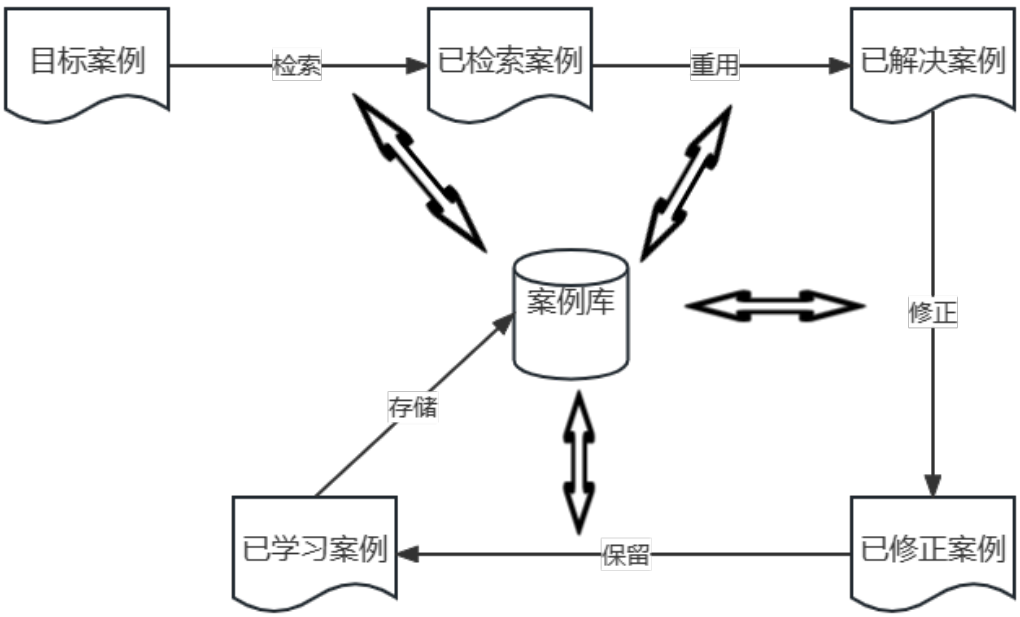


图 6.9　案例推理流程图

表 6.7　问题 3 活动状态数据对应人员编号和活动状态编号判别结果

数据编号	预测人员编号	预测活动状态编号	数据编号	预测人员编号	预测活动状态编号
1-01	10	10	3-07	6	4
1-02	10	11	3-08	6	5
1-03	10	12	3-09	6	6
1-04	10	1	3-10	6	7
1-05	10	2	3-11	6	8
1-06	10	3	3-12	6	9
1-07	10	4	4-01	9	10
1-08	10	5	4-02	9	11
1-09	10	6	4-03	9	12
1-10	10	7	4-04	9	1
1-11	10	8	4-05	9	2
1-12	10	9	4-06	9	3
2-01	7	10	4-07	9	4
2-02	5	11	4-08	9	5
2-03	7	12	4-09	9	6
2-04	7	1	4-10	9	7
2-05	7	2	4-11	9	8
2-06	7	3	4-12	9	9
2-07	7	4	5-01	12	10
2-08	7	5	5-02	13	11
2-09	7	6	5-03	8	12
2-10	6	7	5-04	13	1
2-11	7	8	5-05	13	2
2-12	7	9	5-06	13	3
3-01	9	10	5-07	13	4
3-02	7	11	5-08	13	5
3-03	4	12	5-09	13	6
3-04	6	1	5-10	13	7
3-05	6	2	5-11	12	8
3-06	6	3	5-12	13	9

表 6.8　问题 3 结果

活动类型	判别状态
Unknow1	10
Unknow2	7
Unknow3	6
Unknow4	9
Unknow5	13

7 模型的评价与推广

7.1 模型优点

（1）本文综合运用了 Mahony 算法、互信息、K-means 算法和 Stacking 集成学习等多种方法，展示了在处理移动终端数据信息时的方法多样性和整合能力，这种综合方法能够有效地提高模型的复杂度和准确性。

（2）在问题二中，建立了基于 Stacking 集成学习的活动状态判别模型，并通过详细的实验分析和验证，展示了该模型在测试集上达到了 92.67% 的高准确率，这表明模型在实际应用中具有良好的分类效果和稳定性。

（3）文章创新地利用 Mahony 算法进行姿态解算，增加了姿态角数据，并应用互信息进行特征选择，从而提升了数据处理的精度和模型的解释能力。

7.2 模型缺点

（1）对于问题一中的基于互信息特征选择的 K-means 聚类模型，其聚类效果与实际差距较大，还需进一步提升。

（2）基于 Stacking 集成学习的活动状态判别模型，其内部决策过程难以解释，同时在新数据集或不同环境中的泛化能力需要进一步验证和调整。

7.3 模型推广

基于互信息与 Stacking 集成学习的人体活动状态判别模型具有广泛的应用潜力。可以应用于健康监测与运动分析，如识别步态、姿势分类，支持个体健康管理和运动训练；在安全与安防领域，用于人员行为识别和异常检测；在智能设备与智能家居中，实现根据用户活动状态的智能控制；同时，也适用于工业生产与操作监控，优化工作人员的工作状态和操作行为，提升生产效率与安全性。

参考文献

[1] Bouten, Carlijin, V. C , et al. A triaxial accelerometer and portable data processing unit for the assessment of daily physical. [J]. IEEE Transactions on Biomedical Engineering, 1997.

[2] J. B. J. Bussmann, W. L. J. Martens, J. H. M. Tulen, F. C. Schasfoort, H. J. G. van den Berg-Emons and H. J. Stam. Measuring daily behavior using ambulatory accelerometry: the Activity Monitor[J]. Behavior Research Methods, Instruments, & Computers, 2001, 33(3): 349-356.

[3] M. J. Mathie,B. G. Celler,Dr N. H. Lovell,A. C. F. Coster.Classification of basic daily movements using a triaxial accelerometer[J].Medical & biological engineering & computing,2004, 42(5):679 687.

[4] Tognetti A , Lorussi F , Tesconi M , et al. Wearable kinesthetic systems for capturing and classifying body posture and gesture[C]// International Conference of the Engineering in Medicine & Biology Society. Conf Proc IEEE Eng Med Biol Soc, 2005.

[5] Jeong D U, Kim S J, Chung W Y. Classification of posture and movement using a 3-axis accelerometer[C]//2007 International Conference on Convergence Information Technology (ICCIT 2007). IEEE, 2007: 837-844.

[6] He Z, Jin L. Activity recognition from acceleration data based on discrete consine transform and SVM[C]//2009 IEEE International Conference on Systems, Man and Cybernetics. IEEE, 2009: 5041-5044.

[7] Putchana W , Chivapreecha S , Limpiti T . Wireless intelligent fall detection and movement classification using fuzzy logic[C]// Biomedical Engineering International Conference (BMEiCON), 2012. IEEE, 2013.

[8] Li Q, Stankovic J A, Hanson M A, et al. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information[C]//2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks. IEEE, 2009: 138-143.

[9] Yeoh W S, Pek I, Yong Y H, et al. Ambulatory monitoring of human posture and walking speed using wearable accelerometer sensors[C]//2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2008: 5184-5187.

[10] Verma A , Merchant R A , Seetharaman S , et al. An intelligent technique for posture and fall detection using multiscale entropy analysis and fuzzy logic[C]// Region 10 Conference. IEEE, 2016.

[11] 张洁. 基于加速度传感器的人体运动行为识别研究 [J]. 自动化与仪器仪表, 2016 (3): 228-229.

[12] 徐川龙, 顾勤龙, 姚明海. 一种基于三维加速度传感器的人体行为识别方法 [J]. 计算机系统应用, 2013, 22(6): 132-135.

[13] 李路. 基于多传感器的人体运动模式识别研究 [D]. 山东大学, 2013.

附录

附录 1

介绍：该代码是 Matlab 语言编写的，作用是进行姿态角解算

```
function q = Mahony(q, ax, ay, az, gx, gy, gz)

    % 定义
    sampleFreq = 1000.0; % 采样频率 (Hz)
    twoKp = 2.0 * 0.5;   % 2 * 比例增益
    twoKi = 2.0 * 0.0;   % 2 * 积分增益

    % 变量定义
    persistent integralFBx integralFBy integralFBz

    if isempty(integralFBx)
        integralFBx = 0.0;
        integralFBy = 0.0;
        integralFBz = 0.0;
    end

    % 只在加速度计数据有效时进行运算
    if ~(ax == 0.0 && ay == 0.0 && az == 0.0)
        % 将加速度计得到的实际重力加速度向量单位化
        recipNorm = invSqrt(ax * ax + ay * ay + az * az);
        ax = ax * recipNorm;
        ay = ay * recipNorm;
        az = az * recipNorm;

        % 通过四元数得到理论重力加速度向量
        halfvx = q(2) * q(4) - q(1) * q(3);
        halfvy = q(1) * q(2) + q(3) * q(4);
        halfvz = q(1) * q(1) - 0.5 + q(4) * q(4);

        % 对实际重力加速度向量与理论重力加速度向量做外积
        halfex = (ay * halfvz - az * halfvy);
        halfey = (az * halfvx - ax * halfvz);
        halfez = (ax * halfvy - ay * halfvx);

        % 在 PI 补偿器中积分项使能情况下计算并应用积分项
        if twoKi > 0.0
            % 积分过程
            integralFBx = integralFBx + twoKi * halfex * (1.0 / sampleFreq);
            integralFBy = integralFBy + twoKi * halfey * (1.0 / sampleFreq);
            integralFBz = integralFBz + twoKi * halfez * (1.0 / sampleFreq);
            % 应用误差补偿中的积分项
            gx = gx + integralFBx;
            gy = gy + integralFBy;
            gz = gz + integralFBz;
        else
            % 避免为负值的 Ki 时积分异常饱和
            integralFBx = 0.0;
            integralFBy = 0.0;
            integralFBz = 0.0;
        end

        % 应用误差补偿中的比例项
        gx = gx + twoKp * halfex;
        gy = gy + twoKp * halfey;
        gz = gz + twoKp * halfez;
    end
```

```
    % 微分方程迭代求解
    gx = gx * (0.5 * (1.0 / sampleFreq));
    gy = gy * (0.5 * (1.0 / sampleFreq));
    gz = gz * (0.5 * (1.0 / sampleFreq));
    qa = q(1);
    qb = q(2);
    qc = q(3);
    q(1) = q(1) + (-qb * gx - qc * gy - q(4) * gz);
    q(2) = q(2) + (qa * gx + qc * gz - q(4) * gy);
    q(3) = q(3) + (qa * gy - qb * gz + q(4) * gx);
    q(4) = q(4) + (qa * gz + qb * gy - qc * gx);

    % 单位化四元数，保证四元数在迭代过程中保持单位性质
    recipNorm = invSqrt(q(1) * q(1) + q(2) * q(2) + q(3) * q(3) + q(4) * q(4));
    q(1) = q(1) * recipNorm;
    q(2) = q(2) * recipNorm;
    q(3) = q(3) * recipNorm;
    q(4) = q(4) * recipNorm;

end
function euler_angles = quaternionToEulerAngles(q)
    % 将四元数 q 反解为欧拉角 (yaw, pitch, roll)
    % q 是一个四元数，格式为 [w, x, y, z]

    % 计算旋转矩阵的元素
    R = [
        1 - 2*q(3)^2 - 2*q(4)^2,     2*q(2)*q(3) - 2*q(4)*q(1),     2*q(2)*q(4) + 2*q(3)*q(1);
        2*q(2)*q(3) + 2*q(4)*q(1),   1 - 2*q(2)^2 - 2*q(4)^2,       2*q(3)*q(4) - 2*q(2)*q(1);
        2*q(2)*q(4) - 2*q(3)*q(1),   2*q(3)*q(4) + 2*q(2)*q(1),     1 - 2*q(2)^2 - 2*q(3)^2
    ];

    % 计算欧拉角 (yaw, pitch, roll)
    % yaw (航向角) = atan2(R(2,1), R(1,1))
    % pitch (俯仰角) = -asin(R(3,1))
    % roll (滚转角) = atan2(R(3,2), R(3,3))
    euler_angles = [
        -asin(R(3,1));              % pitch (俯仰角)
        atan2(R(2,1), R(1,1));      % yaw (航向角)
        atan2(R(3,2), R(3,3))       % roll (滚转角)
    ];

    % 将弧度转换为角度
    euler_angles = rad2deg(euler_angles);
end
```

附录 2

介绍：该代码是 Matlab 语言编写的，作用是进行数据处理

```
clc;
clear;
% 列出文件夹对应内容
% file_read=dir('C:\Users\Lenovo\Desktop\2024 年湖南省研究生数学建模竞赛赛题及附件\2024 年湖南省研究生数学建模竞赛赛题\A 题\附件 2\Person13\*.xlsx');
file_read=dir('C:\Users\Lenovo\Desktop\2024 年湖南省研究生数学建模竞赛赛题及附件\2024 年湖南省研究生数学建模竞赛赛题\A 题\附件 3\*.xlsx');
filenames={file_read.name}';
file_length=length(file_read);

for k = 1:file_length
    % 读取
```

```
%   currentFname = strcat('C:\Users\Lenovo\Desktop\2024 年湖南省研究生数学建模竞赛赛题及附件\2024 年湖南省研究生数学建模竞赛赛题\A 题\附件 2\Person13\',filenames{k});
    currentFname = strcat('C:\Users\Lenovo\Desktop\2024 年湖南省研究生数学建模竞赛赛题及附件\2024 年湖南省研究生数学建模竞赛赛题\A 题\附件 3\',filenames{k});
    data = readtable(currentFname,'VariableNamingRule','preserve');

    % 定义初始四元数 [w, x, y, z]
    q = [1.0, 0.0, 0.0, 0.0];

    [m,n]=size(data);
    % 初始化存储姿态角的数组
    euler_angles = zeros(m, 3);

    % 循环更新四元数，并计算姿态角
    for i = 1:m
        q = Mahony(q,  data.("acc_x(g)")(i), data.("acc_y(g)")(i),data.("acc_z(g)")(i),data.("gyro_x(dps)")(i), data.("gyro_y(dps)")(i),data.("gyro_z(dps)")(i));
        euler_angles(i, :) = quaternionToEulerAngles(q);
    end


    T=array2table(euler_angles,'VariableNames',{'pitch','yaw','row'});

    % 垂直合并表格
    t_cat = horzcat(data,T);

    %写入数据
%     repetname = strcat('C:\Users\Lenovo\Desktop\Data\附件 2\Person13\',filenames{k});
    repetname = strcat('C:\Users\Lenovo\Desktop\Data\附件 3\',filenames{k});
    writetable(t_cat,repetname);

    disp(repetname)

end
```

附录 3

介绍：该代码是 Matlab 语言编写的，作用是提取数据时域与频域特征

```
clc;
clear;
% 列出文件夹对应内容
% file_read=dir('C:\Users\Lenovo\Desktop\Data\附件 1\Person3\*.xlsx');
file_read=dir('C:\Users\Lenovo\Desktop\Data\附件 3\*.xlsx');
filenames={file_read.name}';
file_length=length(file_read);

T=table();

for k = 1:file  length
    % 读取
%     currentFname = strcat('C:\Users\Lenovo\Desktop\Data\附件 1\Person3\',filenames{k});
    currentFname = strcat('C:\Users\Lenovo\Desktop\Data\附件 3\',filenames{k});
    data = readtable(currentFname,'VariableNamingRule','preserve');

    [m,n]=size(data);
    colname=data.Properties.VariableNames;

    for i=1:9
        da=data.(colname{i});

        [m,n]=size(da);

        %傅里叶变换
```

```
df=fftshift(fft(ifftshift(da)));
%% 时域特征
%平均值
t=strcat(colname{i},"_mean");
T.(t)(k)=mean(da);

%标准差
t=strcat(colname{i},"_std");
T.(t)(k)=std(da);

%众数
t=strcat(colname{i},"_mode");
T.(t)(k)=mode(da);

%最大值
t=strcat(colname{i},"_max");
T.(t)(k)=max(da);

%最小值
t=strcat(colname{i},"_min");
T.(t)(k)=min(da);

%极差
t=strcat(colname{i},"_range");
T.(t)(k)=abs(max(da)-min(da));

%过均值点个数
sAboveMean = da > mean(da);
t=strcat(colname{i},"_above_mean");
T.(t)(k)=sum(sAboveMean);

%% 频域特征

% 采样频率
Fs = 100; % 采样频率设置为 100 Hz

% 计算功率谱密度
N = length(da); % 数据点数
[Pxx, f] = pwelch(da, [], [], [], Fs); % 使用 pwelch 计算功率谱密度

% 计算幅度统计特征
% 平均值
t=strcat(colname{i},"_fft_mean");
T.(t)(k)=mean(Pxx);

%标准差
t=strcat(colname{i},"_fft_std");
T.(t)(k)=std(Pxx);

%最大值
t=strcat(colname{i},"_fft_max");
T.(t)(k)=max(Pxx);

%最小值
t=strcat(colname{i},"_fft_min");
T.(t)(k)=min(Pxx);

% 形状统计特征（偏度和峰度）
% 偏度
t=strcat(colname{i},"_fft_skewness");
T.(t)(k) = skewness(Pxx);
% 峰度
t=strcat(colname{i},"_fft_kurtosis");
T.(t)(k) = kurtosis(Pxx);
```

```
    end
end

%% 写入数据
% writetable(T,'C:\Users\Lenovo\Desktop\Data\附件 1\result\Person3_oupt.xlsx');
writetable(T,'C:\Users\Lenovo\Desktop\Data\附件 3\oupt_data.xlsx');
```

附录 4

介绍：该代码是 Python 语言编写的，作用是进行聚类

```
import pandas as pd
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import SelectKBest, mutual_info_classif
 读取数据
# data = pd.read_excel('C:/Users/Lenovo/Desktop/Data/附件 2/data_k.xlsx')
data = pd.read_excel('C:/Users/Lenovo/Desktop/Data/附件 2/data_k.xlsx')
# 数据清理和预处理
data = data.dropna()
X = data
# 数据标准化
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# K-means 聚类
n_clusters = 12  # 设置聚类的数量
kmeans = KMeans(n_clusters=n_clusters, random_state=0)
kmeans_labels = kmeans.fit_predict(X_scaled)
print("K-means 聚类结果:")
print(kmeans_labels)
# 将结果添加到原始数据中
data['K-means'] = kmeans_labels
# data['Hierarchical'] = hierarchical_labels


# 输出每个样本的聚类标签
print("\n 完整的聚类结果（包括原始数据和聚类标签）:")
print(data)
# 保存结果到新的 Excel 文件
# output_file_path = 'C:/Users/Lenovo/Desktop/Data/附件 2/clustered_data.xlsx'
output_file_path = 'C:/Users/Lenovo/Desktop/Data/附件 2/No2_clustered_data.xlsx'
data.to_excel(output_file_path, index=False)
```

附录 5

介绍：该代码是 Python 语言编写的，作用是单一模型训练与评估

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report, accuracy_score

# 从指定的 Excel 文件加载数据
file_path = 'C:/Users/Lenovo/Desktop/Data/附件 2/Data.xlsx'
df = pd.read_excel(file_path)

# 假设数据包括特征列和目标列（最后一列是目标列）
X = df.iloc[:, :-1]  # 特征列
y = df.iloc[:, -1]  # 目标列

# 数据标准化
scaler = StandardScaler()
X = scaler.fit_transform(X)

# 划分训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 初始化分类器
models = {
    'Random Forest': RandomForestClassifier(),
    'SVM': SVC(),
    'KNN': KNeighborsClassifier(),
    'Gradient Boosting': GradientBoostingClassifier(),
    'Naive Bayes': GaussianNB(),
    'Decision Tree': DecisionTreeClassifier(),
    'Logistic Regression': LogisticRegression(),
    'MLP': MLPClassifier()
}

# 训练和评估每个模型
results = {}
for model_name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # 获取分类报告
    report = classification_report(y_test, y_pred, output_dict=True)

    # 保存评估结果
    results[model_name] = {
        'accuracy': accuracy_score(y_test, y_pred),
        'precision': report['macro avg']['precision'],
        'recall': report['macro avg']['recall'],
        'f1-score': report['macro avg']['f1-score'],
        'classification_report': classification_report(y_test, y_pred, output_dict=False)
    }

# 转换为 DataFrame 并保存到 Excel 文件
df_results = pd.DataFrame(results).T  # 转置以模型名称作为行索引
df_results.to_excel('classification_results.xlsx', index=True)
print("分类评估结果已保存到 classification_results.xlsx 文件中。")
```

附录 6

介绍：该代码是 Python 语言编写的，作用是 Stacking 组合模型训练与预测

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
```

```
from sklearn.ensemble import StackingClassifier, RandomForestClassifier, GradientBoostingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.feature_selection import SelectKBest, mutual_info_classif
import numpy as np

# 1. 加载数据
file_path = 'C:/Users/Lenovo/Desktop/Data/附件 2/Data.xlsx'
data = pd.read_excel(file_path)
test_data_path = 'C:/Users/Lenovo/Desktop/Data/附件 3/oupt_data.xlsx'
test_data = pd.read_excel(test_data_path)

# 2. 数据处理和特征工程
X = data.iloc[:, :-1]  # 特征数据
y = data.iloc[:, -1]  # 标签数据

# 3. 将类别标签从 1-12 改为 0-11
y = y - 1

# 4. 使用互信息选择特征
k = 60  # 选择前 k 个互信息最高的特征
selector = SelectKBest(mutual_info_classif, k=k)
X_selected = selector.fit_transform(X, y)

# 5. 数据打乱和划分训练集测试集
X_train, X_test, y_train, y_test = train_test_split(X_selected, y, test_size=0.2, random_state=42)

# 6. 定义基本分类器
base_estimators = [
    ('rf', RandomForestClassifier(random_state=42)),
    ('knn', KNeighborsClassifier()),
    ('svm', SVC(random_state=42))
]

# 7. 定义元分类器
meta_classifier = GradientBoostingClassifier(random_state=42)

# 8. 定义 Stacking 分类器
stacking_clf = StackingClassifier(
    estimators=base_estimators,
    final_estimator=meta_classifier,
    cv=5  # 交叉验证折数
)

# 9. 定义参数网格
param_grid = {
    'rf__n_estimators': [50],
    'knn__n_neighbors': [3],
    'svm__C': [1],
    'final_estimator__learning_rate': [0.1],
    'final_estimator__max_depth': [3]
}

# 10. 网格搜索
grid_search = GridSearchCV(
    estimator=stacking_clf,
    param_grid=param_grid,
    scoring='accuracy',
    cv=3  # 减少网格搜索的折数
)

# 11. 训练模型并搜索最佳参数
```

```
grid_search.fit(X_train, y_train)

# 12. 输出最佳参数和最佳得分
print("Best Parameters:", grid_search.best_params_)
print("Best CV Score:", grid_search.best_score_)

# 13. 使用最佳模型进行预测
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)

# 14. 输出评估结果
print("Stacking Classifier Classification Report:")
print(classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))

# # 保存训练集预测结果
# results_df = pd.DataFrame({'True Labels': y_test, 'Predicted Labels': y_pred})
# results_df.to_excel('C:/Users/Lenovo/Desktop/Data/附件 2/prediction_results.xlsx', index=False)

# 15. 对 test_data 进行特征选择和预测
X_test_data = test_data.values  # 假设测试数据的最后一列是标签，但不使用
X_test_data_selected = selector.transform(X_test_data)

# 使用最佳模型进行预测
y_test_data_pred = best_model.predict(X_test_data_selected)

# 保存测试集预测结果
test_results_df = pd.DataFrame({'Predicted Labels': y_test_data_pred})
test_results_df.to_excel('C:/Users/Lenovo/Desktop/Data/附件 3/prediction_test_results.xlsx', index=False)
```

附录 7

介绍：该代码是 Python 语言编写的，作用是证据推理下的人物画像刻画

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
plt.rcParams['font.sans-serif'] = ['SimHei']  # 指定默认字体为黑体
plt.rcParams['axes.unicode_minus'] = False    # 解决负号'-'显示为方块的问题

# Load the Excel file
file_path = r'C:\Users\王纪凯\Desktop\a1t1.xlsx'
all_sheets_data = pd.read_excel(file_path, sheet_name=None)

# Define variables and color map
variables = ['acc  x(g)', 'acc  y(g)', 'acc  z(g)', 'gyro  x(dps)', 'gyro  y(dps)', 'gyro  z(dps)']
sheet_names = all_sheets_data.keys()
colors = plt.cm.viridis(np.linspace(0, 1, len(sheet_names)))

# Determine the minimum length across all sheets
min_length = min(len(all_sheets_data[sheet_name]) for sheet_name in sheet_names)

# Plot each variable in a separate figure with distinct colors for each sheet
for i, var in enumerate(variables):
    fig, ax = plt.subplots(figsize=(12, 8))
    for sheet_name, color in zip(sheet_names, colors):
        ax.plot(all_sheets_data[sheet_name][var][:min_length], label=f'Sheet {sheet_name}', color
=color)

    ax.set_title(f'{var} 在所有表格中的变化')
```

```
    ax.set_xlabel('样本点')
    ax.set_ylabel(var)
    ax.legend(loc='upper right', fontsize='small')

    file_path = f'variable_plot_{i}.png'
    fig.savefig(file_path)
    plt.close(fig)



# Plot each variable in a separate figure with distinct colors for each sheet
for i, var in enumerate(variables):
    fig, ax = plt.subplots(figsize=(12, 8))
    for sheet_name in sheet_names:
        if var in all_sheets_data[sheet_name].columns:
            color = next(color_cycle) # 获取颜色
            ax.plot(all_sheets_data[sheet_name][var][:min_length],
                    label=f'{sheet_name}',
                    color=color)

    ax.set_title(f'{var} 在所有表格中的变化')
    ax.set_xlabel('样本点')
    ax.set_ylabel(var)
    ax.legend(loc='upper right', fontsize='small')

    # 调整子图间距，确保坐标轴标签和标题不会重叠
    plt.tight_layout()

    # 保存图表为图片，设置高 dpi 以提高清晰度
    save_path = os.path.join(r'C:\Users\王纪凯\Desktop', f'{var}.png')
    plt.savefig(save_path, dpi=300) # 300dpi 通常提供较好的清晰度
    plt.close(fig) # 关闭图表


# In[78]:


# ...之前的代码...



# Plot each variable in a separate figure with distinct colors for each sheet
for i, var in enumerate(variables):
    fig, ax = plt.subplots(figsize=(12, 8))
    for sheet_name in sheet_names:
        if var in all_sheets_data[sheet_name].columns:
            color = next(color_cycle) # 获取颜色
            ax.plot(all_sheets_data[sheet_name][var][:min_length],
                    label=f'{sheet_name}',
                    color=color)

    ax.set_xlabel('样本点', fontsize=22)
    ax.set_ylabel(var, fontsize=22)
    ax.legend(loc='upper right', fontsize=12)
```

```
    # 调整子图间距，确保坐标轴标签和标题不会重叠
    plt.tight_layout()

    # 保存图表为图片，设置高 dpi 以提高清晰度
    save_path = os.path.join(r'C:\Users\王纪凯\Desktop', f'{var}.png')
    plt.savefig(save_path, dpi=300)  # 300dpi 通常提供较好的清晰度
    plt.close(fig)  # 关闭图表


file_path = r'C:\Users\王纪凯\Desktop\14-a1t1.xlsx'
all_sheets_data = pd.read_excel(file_path, sheet_name=None)

# 定义传感器数据列和颜色
variables = ['acc_x(g)', 'acc_y(g)', 'acc_z(g)', 'gyro_x(dps)', 'gyro_y(dps)', 'gyro_z(dps)']
sheet_names = all_sheets_data.keys()
colors = plt.cm.viridis(np.linspace(0, 1, len(sheet_names)))

# 确定所有表格中最短的长度
min_length = min(len(all_sheets_data[sheet_name]) for sheet_name in sheet_names)

# 计算统计特征并可视化差异
stats = {var: [] for var in variables}
for sheet_name in sheet_names:
    data = all_sheets_data[sheet_name]
    for var in variables:
        mean_val = data[var][:min_length].mean()
        std_val = data[var][:min_length].std()
        stats[var].append({'sheet': sheet_name, 'mean': mean_val, 'std': std_val})

# 可视化差异
for var in variables:
    fig, ax = plt.subplots(figsize=(12, 8))
    means = [item['mean'] for item in stats[var]]
    stds = [item['std'] for item in stats[var]]
    ax.bar(sheet_names, means, yerr=stds, capsize=5, color=colors)
    ax.set_xlabel('人员', fontsize=24)
    ax.set_ylabel(var)
    plt.xticks(rotation=45)

    # 设置横纵坐标标签和图例的字体大小
    ax.set_xlabel('样本点', fontsize=24)  # 增大字体大小
    ax.set_ylabel(var, fontsize=24)  # 增大字体大小
    ax.legend(loc='upper right', fontsize=12)  # 增大图例字体大小
    plt.xticks(fontsize=20)
    plt.yticks(fontsize=20)



    # 保存图表为图片，设置高 dpi 以提高清晰度
    save_path = os.path.join(r'C:\Users\王纪凯\Desktop', f'{var}.png')
    plt.savefig(save_path, dpi=300)  # 300dpi 通常提供较好的清晰度
    plt.close(fig)  # 关闭图表


# In[36]:
```

```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Load the Excel file
file_path = r'C:\Users\王纪凯\Desktop\14-a1t1.xlsx'
all_sheets_data = pd.read_excel(file_path, sheet_name=None)

# 定义传感器数据列
variables = ['acc_x(g)', 'acc_y(g)', 'acc_z(g)', 'gyro_x(dps)', 'gyro_y(dps)', 'gyro_z(dps)']
sheet_names = list(all_sheets_data.keys())

# 确定所有表格中最短的长度
min_length = min(len(all_sheets_data[sheet_name]) for sheet_name in sheet_names)

# 将所有数据合并成一个 DataFrame
all_data = []
labels = []
for sheet_name in sheet_names:
    data = all_sheets_data[sheet_name][variables][:min_length]
    all_data.append(data)
    labels.extend([sheet_name] * min_length)

all_data = pd.concat(all_data).reset_index(drop=True)

# 进行 PCA 降维
pca = PCA(n_components=2)
pca_result = pca.fit_transform(all_data)
pca_df = pd.DataFrame(pca_result, columns=['PCA1', 'PCA2'])
pca_df['label'] = labels

# 进行 K-means 聚类
kmeans = KMeans(n_clusters=13)
kmeans.fit(pca_result)
pca_df['cluster'] = kmeans.labels_

# 可视化 PCA 结果和聚类结果
plt.figure(figsize=(12, 8))
for cluster in range(13):
    cluster_data = pca_df[pca_df['cluster'] == cluster]
    plt.scatter(cluster_data['PCA1'], cluster_data['PCA2'], label=f'Cluster {cluster}')
plt.title('PCA and K-means Clustering of 13 People')
plt.xlabel('PCA1')
plt.ylabel('PCA2')
plt.legend()
plt.show()

# 打印聚类结果
cluster_results = {}
for cluster in range(13):
    cluster_data = pca_df[pca_df['cluster'] == cluster]
    cluster_results[cluster] = cluster_data["label"].unique()
```

```
print(cluster_results)


# 加载实验人员信息
person_info = pd.DataFrame({
    '实验人员编号': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13],
    '年龄': [26, 31, 32, 27, 23, 27, 35, 30, 22, 49, 21, 34, 36],
    '体重': [75, 68, 75, 43, 52, 50, 63, 60, 76, 68, 71, 48, 80],
    '身高': [185, 169, 160, 164, 168, 164, 170, 171, 180, 166, 178, 165, 170]
})

# 加载传感器数据
file_path = r'C:\Users\王纪凯\Desktop\14-a1t1.xlsx'
all_sheets_data = pd.read_excel(file_path, sheet_name=None)
variables = ['acc_x(g)', 'acc_y(g)', 'acc_z(g)', 'gyro_x(dps)', 'gyro_y(dps)', 'gyro_z(dps)']
sheet_names = list(all_sheets_data.keys())

min_length = min(len(all_sheets_data[sheet_name]) for sheet_name in sheet_names)
all_data = []
labels = []
for sheet_name in sheet_names:
    data = all_sheets_data[sheet_name][variables][:min_length]
    all_data.append(data)
    labels.extend([sheet_name] * min_length)
all_data = pd.concat(all_data).reset_index(drop=True)
all_data['label'] = labels

# 关联传感器数据与实验人员信息
person_info['label'] = ['Sheet' + str(i) for i in person_info['实验人员编号']]
merged_data = pd.merge(all_data, person_info, on='label')

# 进行回归分析
X = merged_data[['年龄', '体重', '身高']]
y = merged_data[variables]

# 建立多元线性回归模型
reg = LinearRegression()
reg.fit(X, y)

# 输出回归系数
coefficients = pd.DataFrame(reg.coef_, columns=X.columns, index=variables)
print(coefficients)

# 可视化
plt.figure(figsize=(12, 8))
sns.heatmap(coefficients, annot=True, cmap=sns.diverging_palette(220, 10, as_cmap=True), cba
r_kws={'label': '回归系数'})
plt.xlabel('人员属性', fontsize=14)
plt.ylabel('传感器数据', fontsize=14)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.tight_layout()
plt.show()
# 保存图表为图片，设置高 dpi 以提高清晰度
```

```
save_path = os.path.join(r'C:\Users\王纪凯\Desktop', 'rlitu.png')
plt.savefig(save_path, dpi=300)  # 300dpi 通常提供较好的清晰度
plt.close()  # 关闭图表
```