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）。

我们参赛选择的题号是（从组委会提供的赛题中选择一项填写）：A

我们的参赛编号（请填写完整参赛编号）：202418001014

所属学校和学院：国防科技大学系统工程学院

参赛队员（打印后签名）：1.

2.

3.

指导教师或指导教师组负责人：罗志浩

日期： 2024 年 7 月 12 日

湖南省研究生第九届数学建模竞赛论文

基于机器学习的活动状态分类识别及人员画像

摘要

随着智能手机的广泛使用，许多手机都具有评估用户日常活动消耗热量的功能，比如华为的”华为运动健康”软件. 这类软件依赖智能手机内置的运动传感器——加速度计和陀螺仪来记录用户的活动数据. 加速度计负责测量手机在 X、Y、Z 三个轴向上的线性加速度变化，而陀螺仪则负责检测手机围绕这三个轴向转动的角速度. 通过这些传感器收集的数据，手机处理器能够感知手机的姿态和方向变化，并利用算法判断和跟踪用户的活动模式. 针对基于用户时序活动数据的状态分类和人员画像识别问题，本文立足于数据分解思想，通过将用户时序数据进行压缩降维处理，并使用谱聚类分类模型、随机森林模型、双向长短期记忆网络(Bi-LSTM)机器学习模型，成功地实现了活动数据的分类识别和人员画像的任务.

针对问题一：本问题的关键在于判断人员的每组数据所对应的活动状态，考虑到实验数据无标签的特点，将活动状态识别问题转化为对每组数据进行聚类和簇中心归类的分类问题. 首先结合数据的时序特征，采用小波分解对数据进行压缩，得到每组数据 48 维的特征点；然后使用谱聚类算法对数据点进行聚类；最后综合考虑运动轨迹和时序数据的变化曲线，对簇中心进行活动状态的识别，将簇中心的活动状态作为整个簇的活动状态. 识别结果如下:

- Person1: 第一类为[7, 33, 38, 42, 58]，第二类为[14, 17, 28, 41, 47]，第三类为[5, 8, 46, 56, 59]，第四类为[21, 25, 29, 48, 53]，第五类为[27, 31, 37, 49, 52]，第六类为[19, 26, 43, 44, 50]，第七类为[1, 2, 15, 23, 36]，第八类为[13, 24, 20, 30, 39]，第九类为[9, 10, 12, 18, 57]，第十类为[22, 32, 34, 35, 40]，第十一类为[3, 6, 11, 16, 55]，第十二类为[4, 45, 51, 54, 60].
- Person2: 第一类为[7, 10, 19, 23, 30]，第二类为[16, 21, 29, 40, 43]，第三类为[3, 20, 38, 51, 60]，第四类为[12, 24, 33, 44, 57]，第五类为[2, 26, 37, 46, 47]，第六类为[1, 4, 48, 49, 50]，第七类为[13, 27, 28, 34, 42]，第八类为[5, 11, 22, 54, 56]，第九类为[6, 15, 25, 35, 58]，第十类为[31, 32, 39, 52, 59]，第十一类为[8, 9, 36, 41, 55]，第十二类为[14, 17, 18, 45, 53].
- Person3: 第一类为[11, 24, 25, 31, 44]，第二类为[9, 14, 30, 37, 43]，第三类为[8, 23, 39, 42, 52]，第四类为[13, 27, 29, 51, 56]，第五类为[19, 35, 36, 49, 60]，第六类为[10, 41, 57, 58, 59]，第七类为[4, 5, 18, 22, 53]，第八类为[17, 21, 33, 34, 38]，第九类为[2, 6, 12, 47, 48]，第十类为[3, 32, 50, 54, 55]，第十一类为[1, 7, 26, 40, 45]，第十二类为[15, 16, 20, 28, 46].

针对问题二：本问题分为两个子问题：子问题 1 和子问题 2. 经分析可知，解决两个子问题的前提是，完成问题二判别模型的构建，并且对所有实验人员的活动状态进行判别. 由于数据具有固定的标签，因此该问题实际上是一个监督学习问题，其核心在于 600 个数据集的处理以及判别模型的建立. 基于活动状态多分类思想，本文建立了包含多个决策树模型的随机森林模型. 首先通过小波分解将原始数据转化为 600 个具有标签

的表征向量, 以此作为判别模型的数据集. 随后通过提取活动状态的典型特征, 构建了随机森林模型, 将 80%的数据集作为训练集, 20%的数据集作为验证集. 针对子问题 1, 使用问题一的谱聚类分类模型对验证集数据进行分类, 并通过混淆矩阵与随机森林判别结果进行比较, 此外还进一步分析了分类模型的准确度. 针对子问题 2 要求判别 30 次未知标签活动数据的活动状态, 本文将 30 组数据作为模型的测试集, 直接输入随机森林模型进行状态判别. 结果如下所示:

- 子问题 1: 由于问题一的分类模型为无监督分类模型, 而问题二的判别模型为监督学习模型, 因此分类模型的结果与判别模型相比相对较差, 这一结果同样体现在每个类别的分类精确率上.
- 子问题 2: SY1 5 SY11 9 SY21 6

  SY2 1 SY12 7 SY22 2
  SY3 7 SY13 4 SY23 8
  SY4 12 SY14 3 SY24 5
  SY5 7 SY15 4 SY25 2
  SY6 10 SY16 1 SY26 8
  SY7 2 SY17 4 SY27 8
  SY8 6 SY18 5 SY28 5
  SY9 7 SY19 8 SY29 6
  SY10 10 SY20 8 SY30 5

针对问题三: 本问题可以分成两部分: 子问题 1 和子问题 2. 子问题 1 为分析同一活动状态下不同人员之间的差异, 子问题 2 要求建立传感器数据与人员身份信息的关联. 对于子问题 1, 同样采用小波分解对数据进行压缩, 并使用压缩后的数据作为输入, 构建不同人员之间的皮尔逊相关系数矩阵, 以分析每个人员之间的差异. 对于子问题 2, 考虑到传感器数据的复杂性, 本文构建了双向长短期记忆网络(Bi-LSTM)深度学习模型, 将每组数据作为输入, 并以每组数据的人员标签为输出. 测试未知人员类别时, 将 12 组数据出现次数最多的类别标签作为该人员的识别结果. 结果如下:

- 子问题 1: 不同年龄、身高、体重的个体在同一活动状态下, 传感器数据确实具有显著差异. 说明同一活动状态下, 不同人员由于受到诸如身高等因素的影响, 确实存在活动状态的特征差异.
- 子问题 2: 人员 Unknow1 被识别为 Person10, 人员 Unknow2 被识别为 Person7, 人员 Unknow3 被识别为 Person6, 人员 Unknow4 被识别为 Person9, 人员 Unknow5 被识别为 Person13.

最后, 对提出的模型进行全面评价: 本文的模型贴合实际, 能较好的解决提出的问题, 且具有实用性强, 算法效率高等特点, 该模型在模式识别, 目标分类, 时序处理等方面同样具有良好的扩展性.

关键词: 小波分解 谱聚类 随机森林 相关性分析 Bi-LSTM 人员画像

目录

1 问题重述

1.1 问题背景

随着物联网技术的成熟与应用, 许多智能手机具备评估用户日常活动消耗热量的功能, 例如华为的”华为运动健康”软件. 此类软件依赖智能手机内置的运动传感器——加速度计和陀螺仪记录用户的活动数据. 加速度计测量手机在三个轴向(X, Y, Z)上的线性加速度变化, 而陀螺仪则测量手机围绕三个轴向转动的角速度. 通过这些传感器的数据, 手机处理器能够感知手机的姿态和方向变化, 并使用算法判断和跟踪用户的活动模式.

当前, 众多学者通过多种方法对用户活动状态的分类识别问题进行了研究. 主流的研究方法主要可分为两种: 一是基于传感器数据绘制用户活动轨迹, 但这种方法所需数据量较大, 计算时间较长, 难以应用到实际场景中; 二是采用深度学习模型, 但由于网络模型具有较高复杂性, 且计算资源消耗极大, 因此难以有效学习到用户的活动特征. 针对以上现有方法的不足, 本文旨在探索更优的方法, 以较少的数据量以及较小的模型完成人员活动状态的分类识别和人员画像构建.

1.2 问题概括

问题一对附件 1 中 3 名实验人员的活动数据进行分类, 确定活动状态, 并将结果填入表 1. 其中每名实验人员有 60 组加速度计和陀螺仪的数据, 但未记录活动状态.

问题二利用附件 2 中 10 名实验人员的已标注活动数据, 提取活动状态的典型特征, 建立判别模型, 并进行以下验证: 使用问题 1 中建立的分类模型对附件 2 的数据进行分类, 比较两种模型的分类准确度. 使用判别模型对附件 3 中某实验人员的 30 次活动数据进行分类, 并将结果填入表 2.

问题三分析不同实验人员在同一活动状态下的数据差异, 研究活动状态数据与年龄、身高、体重的关系, 探索使用活动传感器数据进行人员画像, 并识别附件 5 中 5 名实验人员的活动数据来源. 其中附件 4 包含 13 名实验人员的年龄、身高、体重等数据; 附件 5 包含 10 名实验人员中的 5 人的某次活动数据.

1.3 资料条件

附件中邀请了 10 余名实验人员携带活动状态传感器进行活动, 并收集他们的日常活动状态的数据. 规定他们需要完成”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, 每种活动记录了 5 组实验数据, 每组数据记录其数秒的线加速度和角加速度数据, 活动状态编号按 1-12 编号.

2 问题分析

2.1 问题一的分析

问题一的核心是判断人员的每组数据所对应的活动状态. 根据题意, 每位人员的 12 个活动状态均进行了 5 次实验, 而附件 1 包含的是三位人员各自的 60 组数据, 因此可以推导的信息是: 附件 1 的 60 组数据对应的活动状态是均匀的, 即每个活动状态有且只有 5 组数据相对应. 具体到每组数据, 表中包含来自加速度计的三轴加速度和来自陀螺仪的三轴角速度, 是典型的时频(时序)数据. 从技术方法的选择来看, 采用聚类方法更适合本题信息.

2.2 问题二的分析

问题二基于附件 2、3 的活动状态数据, 以及问题一的分类模型和问题二要求构建的判别模型, 主要包含两个子问题.

子问题 1 要求使用问题一的分类模型对附件 2 提供的数据进行状态分类, 并且与建立的问题二判别模型进行结果比较, 然后分析分类模型的性能.

这一问题的核心在于, 如何根据附件 2 带标签的数据提取出活动状态的典型特征, 以及选择哪种监督学习 建立活动状态的判别模型, 并对提供的数据进行判别. 采用哪种方式更加直观的比较两个模型的结果, 以及使用什么标准判断分类模型的准确度.

子问题 2 要求使用判别模型对某人员 30 次活动进行状态判别, 这一问题只需要将所有状态数据输入判别模型中, 即可得到该人员的活动状态.

2.3 问题三的分析

问题三包含两个子问题.

子问题 1 是判断同一活动状态下不同人员之间的差异, 其中存在以下关键问题:

1)人员之间差异的特征是什么? 题干信息未仔细说明不同人员之间差异的特征, 但是说明了"在同一活动状态下"这一前置条件, 因此结合附件 1 和附件 2 给出的数据, 可以推断子问题 1 要求说明在同一活动状态下, 不同人员的传感器数据是否存在差异, 即不同人员测量得出的三轴加速度和三轴角速度是否存在差异.

2)数据类型多、数据量大的情况下如何处理数据? 每位人员的每次活动实验都存在六类数据, 且每类数据均有上千条目. 同一活动状态下, 不同实验次数和不同人员之间的数据量均不一致. 因此, 直接在六类数据的基础上开展差异分析的难度较大, 有必要对这些数据进行适当处理, 例如统一数据量、统一数据维数等.

3)差异的评价标准是什么? 对数据进行处理后, 需要明确如何评价不同人员之间数据的差异.

4)问题 1 三名人员的活动状态未知, 如何处理? 附件 1 给出的三名人员数据中, 不包含相关活动状态, 无法直接参与差异分析. 由于问题 1 要求判断三名人员的活动状态, 因此可以根据问题 1 的分类结果将三名人员添加到差异分析之中, 或者考虑分类结果的不稳定性, 排除这三名人员, 不参与差异分析.

子问题 2 是判断传感器的活动数据和人员的身体数据的联系, 最终根据传感器数据判别人员最可能是哪一位. 活动数据规模大、特征多, 而身体数据规模小、特征少, 活动数据与身体数据可能存在复杂的联系. 该子问题的关键是建立活动数据与人员身份的联系, 可以将人员活动数据作为桥梁, 也可以选择直接拟合二者的关联关系. 从技术方法上看, 需要使用机器学习之类的模型建立复杂关系.

3 问题一方法与求解

3.1 问题求解思路

问题一是一个无监督、无标签的情形, 即数据样本的标签未知, 需要采取措施判别数据样本的标签. 其次, 对于同一个人员, 同一活动状态的实验数据呈现相似的特征, 不同类型姿态之间的实验数据很可能呈现不同的特征. 最后, 题目给出的信息是三名人员均进行了 12 个类型的活动, 每个类型均进行了 5 组实验, 这与数据资料中每名人员包含 60 份表格数据相吻合. 综上所述, 将"聚类-归类"作为求解本题的基本思想. 第一, 针对每名人员, 采用聚类方法将该人员的 60 份表格数据归纳为 12 类, 每类包含 5 份数据,

代表某一类型的活动. 第二, 根据数据资料中的相关特征, 识别不同活动之间的关键差异化特征, 以此确定归纳的类别所代表的具体活动状态.

问题一的技术路线如图 1 所示, 总体思路是先将人员的 60 组数据聚类成带有簇中心的 12 簇, 再推导出这 12 类簇所对应的活动状态.

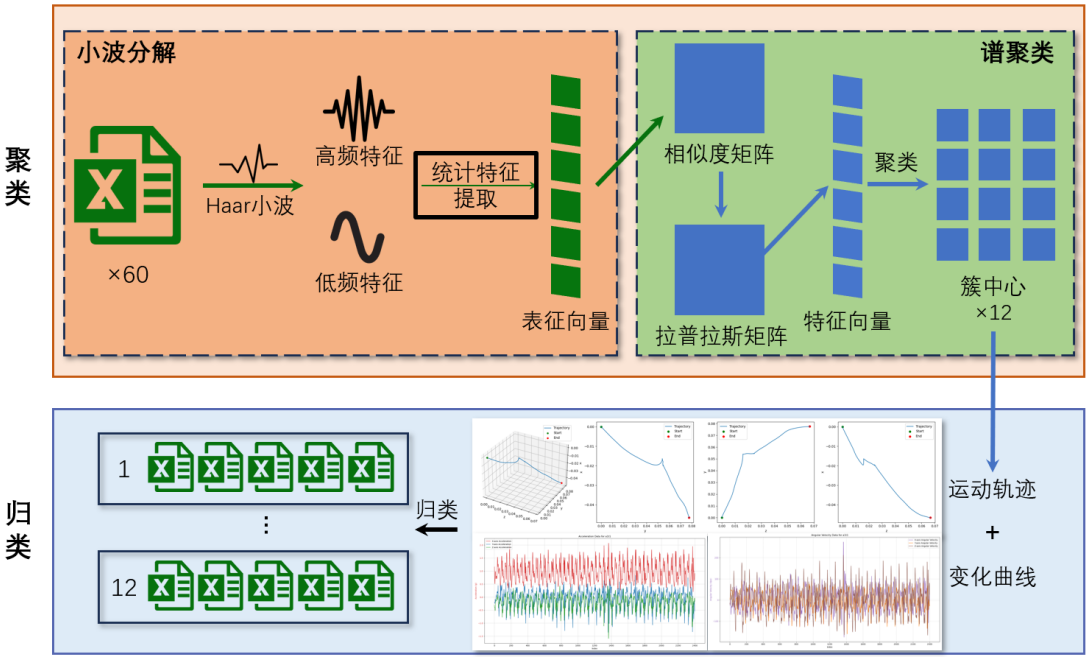

图1 问题一的技术路线

首先, 使用小波分解对每组数据中的数据进行压缩. 小波分解(Wavelet Decomposition)是一种强大的信号处理和分析工具, 具有局部化和稀疏表示等特性. 小波分解的时序局部化特性使得其能够高效捕捉信号的瞬时变化和局部特征, 适合处理非平稳信号. 其稀疏表示特性使得信号在小波基下的表示是稀疏的, 适合数据压缩、降噪和特征提取, 并且降低计算复杂度.

其次, 使用谱聚类对压缩后的数据进行聚类. 谱聚类(Spectral Clustering)是一种基于图论的聚类方法, 具有广泛适用性和稳定性等特性. 谱聚类的广泛适用性在于不仅能处理凸和非凸的数据集, 还可以处理复杂结构和非线性分布的数据. 谱聚类的稳定性在处理高维数据时表现优异.

最后, 使用运动轨迹法和变化曲线图分析法对簇中心进行归类. 每组数据给出的三轴加速度和三轴角速度可以通过物理公式输出运动轨迹, 从而直观判断活动状态; 同时, 三轴加速度和三轴角速度随时间的变化曲线结合物理知识, 也能对判断活动状态提供帮助. 因此, 采取运动轨迹和变化曲线结合的方法对活动状态进行判别.

3.2 同一动作类别识别

3.2.1 小波分解

首先对每名人员的每组数据 $SY_j(j\in\{1,2,\ldots 60\})$ 的六维数据进行压缩, 采用小波分解方法将每组数据数据压缩成一个表征向量. 具体过程如下.

1. 读取人员的每组数据. 读取表中的三轴加速度 $acc\_i, i\in\{x,y,z\}$ 、三轴角速度 $gyro\_i, i\in\{x,y,z\}$. 将以上数据转换为向量形式, 得到六个信号向量, 分别为:

$$\boldsymbol{acc}_i = acc_i(t) \tag{1}$$

$$\boldsymbol{gyro}_i = gyro_i(t) \tag{2}$$

其中 $i \in \{x, y, z\}$，$t \in \{1, 2, 3, \ldots, N\}$，$\boldsymbol{acc}_i$、$\boldsymbol{gyro}_i$ 分别表示加速度向量和角速度向量，N 为每组数据中采样的数据量，代表向量的维数.

2. 选择小波. 小波(Wavelet)是一类特殊数学函数，通过在不同的尺度和位置上对信号进行缩放和平移，达到分析信号局部特性的目的. 小波函数的主要构成为：母小波(Wavelet Function，$\psi(t)$)、尺度函数(Scaling Function，$\phi(t)$)、小波滤波器(High-pass Filter，g_k)、尺度滤波器(Low-pass Filer，h_k). 其中，母小波用于表示信号的高频部分，尺度函数用于表示信号的低频部分，小波滤波器用于提取信号的低频部分，尺度滤波器用于提取信号的高频部分. 它们满足以下递归关系：

$$\phi(t) = \sqrt{2}\sum_k h_k \phi(2t - k) \tag{3}$$

$$\psi(t) = \sqrt{2}\sum_k g_k \psi(2t - k) \tag{4}$$

$$g(t) = (-1)^t h(1 - t) \tag{5}$$

按照特性和应用场景分，常见的小波函数有：Haar 小波、Daubechies 小波、Symlets 小波、Coiflets 小波等等. 其中，Haar 小波由于其简单、易算的特点受到广泛应用，因此选择 Haar 小波作为小波分解的小波函数. Haar 小波的母小波、尺度函数、小波滤波器、尺度滤波器分别定义为：

$$\psi(t) = \begin{cases} 1, \text{if } 0 \le t < \dfrac{1}{2} \\ -1, \text{if } \dfrac{1}{2} \le t < 1 \\ 0, \text{otherwise} \end{cases} \tag{6}$$

$$\phi(t) = \begin{cases} 1, \text{if } 0 \le t < 1 \\ 0, \text{otherwise} \end{cases} \tag{7}$$

$$g(t) = \frac{1}{\sqrt{2}}[1, -1] \tag{8}$$

$$h(t) = \frac{1}{\sqrt{2}}[1, 1] \tag{9}$$

3. 选择小波分解的级数. 小波分解的级数是指信号在小波变换过程中被分解的次数. 在每一分解的层级中，信号都会通过尺度滤波器和小波滤波器被分解为一个低频成分和一个高频成分. 按照递归的原则，每一级的低频成分会被进一步分解，直至达到设定的分解级数或信号的最小分辨率. 为降低计算复杂度，将小波分解的级数设置为 1，即将读取的数据表向量分解一次，得到低频成分和高频成分.

4. 卷积与下采样. 首先，将第一步中的六个信号向量分别与尺度滤波器和小波滤波器进行卷积操作，以加速度向量 $\boldsymbol{acc}_i$ 为例，卷积结果分别为：

$$(acc_i * h)(t) = \sum_k acc_i(k) h(t - k) \tag{10}$$

$$(acc_i * g)(t) = \sum_k acc_i(k) g(t - k) \tag{11}$$

得到信号向量与滤波器的卷积结果后，需对其进行下采样，即每隔 j 个样本取值. 以 $j=2$ 为例，选择 Haar 小波函数，卷积与下采样后的低频信号向量和高频信号向量分别为:

$$\boldsymbol{A}_{acc_i}=\{A(t)\}, A(t)=\frac{1}{\sqrt{2}}(acc_i(2t)+acc_i(2t+1)) \tag{12}$$

$$\boldsymbol{D}_{acc_i}=\{D(t)\}, D(t)=\frac{1}{\sqrt{2}}(acc_i(2t)-acc_i(2t+1)) \tag{13}$$

5. 提取统计特征. 每个信号向量经过前述操作后输出为低频信号向量 $\boldsymbol{A}$ 和高频信号向量 $\boldsymbol{D}$，分别对 $\boldsymbol{A}$、$\boldsymbol{D}$ 进行四类统计特征的提取，分别是:

$$mean(\boldsymbol{A})=\frac{1}{N}\sum_{t=1}^{N}A(t) \tag{14}$$

$$std(\boldsymbol{A})=\sqrt{\frac{1}{N}\sum_{t=1}^{N}(A(t)-mean(\boldsymbol{A}))^2} \tag{15}$$

$$\max(\boldsymbol{A})=\max\{A(1),\ldots,A(t)\} \tag{16}$$

$$\min(\boldsymbol{A})=\min\{A(1),\ldots,A(t)\} \tag{17}$$

由此，一个信号向量可以由 $\boldsymbol{A}$ 和 $\boldsymbol{D}$ 的四类统计特征所表征，将这八个统计特征合为一个向量，即为信号向量的表征向量. $\boldsymbol{acc}_i$ 的表征向量记为:

$$\boldsymbol{sta}(\boldsymbol{acc}_i)=$$
$$[mean(\boldsymbol{A}_{acc_i}), std(\boldsymbol{A}_{acc_i}), \max(\boldsymbol{A}_{acc_i}), \min(\boldsymbol{A}_{acc_i}), mean(\boldsymbol{D}_{acc_i}), std(\boldsymbol{D}_{acc_i}), \max(\boldsymbol{D}_{acc_i}), \min(\boldsymbol{D}_{acc_i})] \tag{18}$$

$\boldsymbol{sta}(\boldsymbol{acc}_i)$ 的维数是 8. 将六个信号向量的特征向量合为一个向量，即为该组数据的表征向量. 1 号人员的数据表 $SY_j(j\in\{1,2,\ldots 60\})$ 的表征向量记为:

$$\boldsymbol{STA}_j=[\boldsymbol{sta}(\boldsymbol{acc}_x), \boldsymbol{sta}(\boldsymbol{acc}_y), \boldsymbol{sta}(\boldsymbol{acc}_z), \boldsymbol{sta}(\boldsymbol{gyro}_x), \boldsymbol{sta}(\boldsymbol{gyro}_y), \boldsymbol{sta}(\boldsymbol{gyro}_z)] \tag{19}$$

$\boldsymbol{STA}_j$ 的维数是 $8\times 6=48$. 将人员的所有组数据的表征向量合为一个 60×48 的矩阵，该矩阵是该人员活动状态的表征矩阵，记为:

$$\boldsymbol{S}_k=\begin{bmatrix}\boldsymbol{STA}_1\\ \vdots\\ \boldsymbol{STA}_{60}\end{bmatrix}(k\in\{1,2,3\}) \tag{20}$$

下一部分的求解内容将以表征向量 $\boldsymbol{STA}_j$ 和人员的表征矩阵 $\boldsymbol{S}_k$ 为基础.

3.2.2 谱聚类

针对小波分解得到的人员的 60 组数据表征向量 $\boldsymbol{STA}_j(j\in\{1,2,\ldots 60\})$，采用谱聚类的方法对这 60 个表征向量进行 12 簇的聚类，最终得到 12 个类别，对应于 12 个活动状态，每个类别含有 5 个表征向量，对应于同一活动状态下的 5 组实验.

谱聚类的基本思想是把数据点视为网络图中的节点，按照某种相似度度量规则构建相似度矩阵，将其作为网络图的边权重，再通过图拉普拉斯矩阵的特征向量将数据点映射到一个低维空间，最后在该空间内对数据点进行聚类. 具体过程如下.

1. 构建相似度矩阵. 记相似度矩阵为$\boldsymbol{W}_{60\times60}$，其元素$W_{ij}$表示表征向量$\boldsymbol{STA}_i$和$\boldsymbol{STA}_j$之间的相似度. 相似度的度量方法通常分为高斯核法与 k-近邻法. 高斯核法的W_{ij}计算方式为:

$$W_{ij}=\exp(-\frac{\left\|\boldsymbol{STA}_i-\boldsymbol{STA}_j\right\|^2}{2\sigma^2}) \tag{21}$$

式中, $\|\bullet\|$为向量的范数, σ为高斯核的带宽参数. k-近邻法的相似度矩阵构建过程为: 首先构建距离矩阵$\boldsymbol{D}_{60\times60}$，其元素的计算方式为:

$$D_{ij}=\sqrt{\sum_{d=1}^{48}(STA_i(d)-STA_j(d))^2} \tag{22}$$

然后对于每个数据点$\boldsymbol{STA}_i$，寻找距离最近的 k 个样本(k 取目标聚类簇数 12), 最后根据数据点之间的 k 近邻关系构建相似度矩阵, 计算方式为:

$$W_{ij}=\begin{cases}1,\ \boldsymbol{STA}_i\text{和}\boldsymbol{STA}_j\text{互为k}-\text{近邻}\\0,\ \text{otherwise}\end{cases} \tag{23}$$

高斯核法适用于处理边界复杂、非线性、高维的数据, 高斯核法的降维能够揭示数据的内在结构; k-近邻法适用于边界简单、数据量小、特征少的场景. 在本题中, 表征向量的数量为 60, 特征维数为 48, 规模较小, 因此采用 k-近邻法构建相似度矩阵.

2. 构建拉普拉斯矩阵. 首先根据相似度矩阵计算度矩阵 $\boldsymbol{Degree}$，该矩阵存储所有节点的度数, 是一个对角矩阵, 其计算方式为:

$$Degree_{ii}=\sum_j W_{ij} \tag{24}$$

获取度矩阵后, 依据相似度矩阵和度矩阵构建拉普拉斯矩阵, 存在以下三种计算方式:

1)未归一化的拉普拉斯矩阵:

$$\boldsymbol{L}=\boldsymbol{Degree}-\boldsymbol{W} \tag{25}$$

2)对称归一化的拉普拉斯矩阵:

$$\boldsymbol{L}_{sym}=\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{\frac{1}{2}} \tag{26}$$

其中$\boldsymbol{I}$是单位矩阵.

3)随机游走归一化的拉普拉斯矩阵:

$$\boldsymbol{L}_{rw}=\boldsymbol{I}-\boldsymbol{D}^{-1}\boldsymbol{W} \tag{27}$$

本题采用未归一化的拉普拉斯矩阵.

3. 特征分解. 对拉普拉斯矩阵进行特征分解, 获取前 12 个最小的非零特征值对应的特征向量$u_1,u_2,\ldots u_{12}$，将这些特征向量合并为60×12的特征矩阵$\boldsymbol{U}$.

4. 数据再表示. 特征矩阵$\boldsymbol{U}$共有 60 个行向量, 每个行向量的维数为 12, 因此, 人员的每组数据数据被表示为表征向量$\boldsymbol{STA}_j$后, 现在被再表示为特征矩阵的行向量, 记为$\boldsymbol{x}_j$. 接下来以$\boldsymbol{x}_j$为聚类过程的输入数据.

5. 聚类. 采用 k-means 聚类方法对获取的 60 个行向量$\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_j,\ldots,\boldsymbol{x}_{60}\}$进行聚类, 其中聚类簇数量 k 取 12. 聚类过程如下.

1)初始化中心点. 选择 12 个初始簇中心$\{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\ldots,\boldsymbol{\mu}_{12}\}$.

2)分配数据点. 将每个数据点$\boldsymbol{x}_j$分配到最近的簇中心$\boldsymbol{\mu}_k$, 记c_j为数据点$\boldsymbol{x}_j$所属的簇, 计算方式为:

$$c_j = \arg\min_k \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_k \right\|^2 \tag{28}$$

3)更新簇中心. 将每个数据点聚类到各初始簇中心的类别后, 需要重新计算每个簇的中心, 计算方式为:

$$\boldsymbol{\mu}_k' = \frac{1}{|C_k|} \sum_{\boldsymbol{x}_j \in C_k} \boldsymbol{x}_j \tag{29}$$

式中, C_k是第k簇的所有数据点集合, $|C_k|$是第k簇数据点的数量.

4)最小化目标函数. 通过重复前述两步骤, 最小化目标函数:

$$J = \sum_{k=1}^{12} \sum_{\boldsymbol{x}_j \in C_k} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_k \right\|^2 \tag{30}$$

此目标函数是簇内数据点与簇中心的平方误差和. 当簇中心不再发生变化后, 聚类停止. 记最终的聚类簇中心为$\{\boldsymbol{\mu}_1^*,\boldsymbol{\mu}_2^*,\ldots,\boldsymbol{\mu}_{12}^*\}$, 对应的簇中心数据表为$\{SY_1^*,SY_2^*,\ldots,SY_{12}^*\}$.

3.3 簇中心归类

前一节采用小波分解和谱聚类的方法将人员的 60 个数据聚类为 12 类簇, 但未明确此 12 个簇的具体含义, 即无法得知 12 个簇分别对应的活动状态. 因此, 本节采用运动轨迹法和曲线图分析法, 明确这些类别对应的活动状态. 采取的思路为: 根据簇中心数据表的运动轨迹和曲线图判断所属的活动状态, 并以簇中心数据表的活动状态作为整个簇的类型, 簇内部的其余四个数据表的类型与簇中心数据表一致.

3.3.1 运动轨迹法

人员右下腹的加速度计记录了三轴加速度, 陀螺仪记录了三轴角速度. 根据这六类数据, 结合物理学的运动规律, 可以推导出人员的运动方程, 结合可视化技术, 则可以呈现人员进行某一活动的运动轨迹图, 从而判断活动状态.

常见的运动方程描述方法有: 四元数法、方向余弦矩阵法、欧拉角法. 其中, 四元数法具有计算效率高、避免奇点等优势, 且与六类活动数据的契合程度高, 因此采用四元数法对人员活动的运动方程进行推导. 四元数指的是由一个实数和一个三维向量构成的数, 表示为

$$\boldsymbol{q} = w + x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k} \tag{31}$$

或者简记为

$$\boldsymbol{q} = (w, x, y, z) \tag{32}$$

其中, w,x,y,z是实数, $\boldsymbol{i},\boldsymbol{j},\boldsymbol{k}$是三个相互垂直的单位向量. 此外, 定义四元数乘法规则:

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 =$$

$$(w_1w_2 - x_1x_2 - y_1y_2 - z_1z_2) + (w_1w_2 + x_1x_2 + y_1y_2 - z_1z_2)\boldsymbol{i} +$$

$$(w_1w_2 - x_1x_2 + y_1y_2 + z_1z_2)\boldsymbol{j} + (w_1w_2 + x_1x_2 - y_1y_2 + z_1z_2)\boldsymbol{k} \tag{33}$$

定义四元数的共轭为

$$\boldsymbol{q}^* = w - x\boldsymbol{i} - y\boldsymbol{j} - z\boldsymbol{k} \tag{34}$$

定义四元数的模为

$$\|\boldsymbol{q}\| = \sqrt{w^2 + x^2 + y^2 + z^2} \tag{35}$$

定义四元数的逆为

$$\boldsymbol{q}^{-1} = \frac{\boldsymbol{q}^*}{\|\boldsymbol{q}\|^2} \tag{36}$$

运动轨迹的具体推导过程如下.

1. 姿态估计. 假设姿态四元数为$\boldsymbol{q}_t$，时间步长为Δt. 首先将三轴角速度转换为四元数表示:

$$\boldsymbol{\omega}_t = (0, gyro_x(t), gyro_y(t), gyro_z(t)) \tag{37}$$

其次, 进行姿态四元数的迭代, 迭代过程为:

$$\boldsymbol{q}_{t+1} = \boldsymbol{q}_t + \frac{1}{2}\Delta t \cdot \boldsymbol{q}_t \otimes \boldsymbol{\omega}_t \tag{38}$$

最后, 进行姿态四元数的归一化, 计算方式为:

$$\boldsymbol{q}_{t+1} = \frac{\boldsymbol{q}_{t+1}}{\|\boldsymbol{q}_{t+1}\|} \tag{39}$$

2. 转换坐标系. 该步骤的目的是将三轴加速度(acc_x, acc_y, acc_z)转换到世界坐标系. 首先将三轴加速度表示为加速度四元数:

$$\boldsymbol{a}_t^{local} = (0, acc_z(t), acc_y(t), acc_x(t)) \tag{40}$$

式中, 根据题目对坐标系的定义, 需要将定义后的坐标系转换为默认坐标系. 然后使用上一步的姿态四元数, 对加速度四元数进行转换, 得到世界坐标系下的加速度, 转换方式为:

$$\boldsymbol{a}_t^{world} = \boldsymbol{q}_t \otimes \boldsymbol{a}_t^{local} \otimes \boldsymbol{q}_t^{-1} \tag{41}$$

3. 重力补偿. 由于加速度计测量的重力方向加速度包含重力, 因此需要从世界坐标系下的加速度中减去重力加速度:

$$\boldsymbol{a}_t^{motion} = \boldsymbol{a}_t^{world} - \boldsymbol{g} \tag{42}$$

4. 速度计算. 对加速度进行积分, 得到速度为:

$$v_{t+1} = v_t + \boldsymbol{a}_t^{motion} \cdot \Delta t \tag{43}$$

5. 位置计算. 对速度进行积分, 得到位置为:

$$p_{t+1} = p_t + v_t \cdot \Delta t \tag{44}$$

在实际描绘运动轨迹的过程中, 加速度计和陀螺仪的零偏漂移(bias drift)和噪声在积分运算中会被放大, 从而导致轨迹偏移, 因此需要引进相关的改进方法. 本文在描绘运动轨迹时, 采取以下方法:

1. 高通滤波器. 高通滤波器允许频率高于某个截止频率的信号通过, 而对低于截止频率的信号进行衰减. 具体来说, 对于一个输入信号 $x(t)$, 高通滤波器输出信号 $y(t)$ 可以表示为:

$$Y(f) = H(f) \cdot X(f) \tag{45}$$

其中 $H(f)$ 是滤波器的频率响应函数, 设计成在高频时接近 1, 在低频时接近 0. 常用的 Butterworth 低通滤波器的传递函数为:

$$H(s) = \frac{1}{\sqrt{1 + (s/\omega_c)^{2n}}} \tag{46}$$

式中, ω_c 为截止角频率, n 为滤波器的阶数. 通过频率反转, 将低通滤波器的传递函数转换为高通滤波器的传递函数:

$$H(s)_{high} = H(\omega_c / s) \tag{47}$$

此高通滤波器可以有效去除加速度信号中的低频成分(重力和慢速变化), 保留高频成分(运动的加速度变化).

2. 卡尔曼滤波器. 卡尔曼滤波器是一种适用于时间序列数据平滑和去噪的递归算法, 可以在噪声环境中对系统状态进行最优评估. 具体过程如下.

1)线性动态系统的状态向量 x_t 受控于状态方程和观测方程, 分别为:

$$x_t = F_t x_{t-1} + B_t u_t + w_t \tag{48}$$

$$z_t = H_t x_t + v_t \tag{49}$$

状态方程中, F_t 是状态转移矩阵, B_t 是控制输入矩阵, u_t 是控制向量, w_t 是过程噪声. 观测方程中, z_t 是观测向量, H_t 是观测矩阵, v_t 是观测噪声.

2)预测. 在该步骤中, 利用之前的状态估计来预测当前时刻的状态和误差协方差. 状态预测方程为:

$$\hat{x}_t^- = F_t \hat{x}_{t-1} + B_t u_t \tag{50}$$

误差协方差预测为:

$$P_t^- = F_t P_{t-1} F_t^T + Q_t \tag{51}$$

3)更新. 在该步骤中, 利用新的观测数据来修正预测的状态估计和误差协方差. 首先计算卡尔曼增益:

$$K_t = P_t^- H_t^T (H_t P_t^- H_t^T + R_t)^{-1} \tag{52}$$

然后更新状态估计:

$$\hat{x}_t = \hat{x}_t^- + K_t(z_t - H_t \hat{x}_t^-) \tag{53}$$

最后更新误差协方差:

$$P_t = (I - K_t H_t) P_t^- \tag{54}$$

3. 零速更新法. 零速更新法的主要目的是对运动物体的某些静止或低速时刻进行速度的修正, 从而减小累积误差. 具体过程如下.

(1)零速检测. 当加速度和角速度信号满足以下条件时, 认为达到零速标准:

$$\|a(t) - g\| < \epsilon_a \tag{55}$$

且:

$$\|\omega(t)\| < \epsilon_{\omega} \tag{56}$$

其中，$a(t)$ 是加速度测量值，g 是重力加速度，$\omega(t)$ 是角速度测量值，ϵ_a 和 ϵ_{ω} 是检测阈值.

(2)卡尔曼滤波器状态修正. 检测到零速时刻 t_k 时，更新阶段进行状态修正:

$$K_k = P_{k|k-1}H_k^T(H_k P_{k|k-1} H_k^T + R_k)^{-1} \tag{57}$$

$$x_{k|k} = x_{k|k-1} + K_k(z_k - H_k x_{k|k-1}) \tag{58}$$

$$P_{k|k} = (I - K_k H_k)P_{k|k-1} \tag{59}$$

式中，K_k 表示卡尔曼增益，H_k 表示测量矩阵，R_k 表示测量噪声协方差矩阵，z_k 表示测量值.

由此，根据簇中心数据表的三轴加速度和三轴角速度，描绘该活动的运动轨迹，期间采用高通滤波器和卡尔曼滤波器对信号进行平滑和去噪处理，以初步达到识别活动状态的目的.

3.3.2 变化曲线图分析法

众所周知，同一人员的不同活动，会带来不同的加速度和角速度的变化，其加速度和角速度随时间将呈现不同的发展趋势. 因此，活动之间的加速度和角速度发展趋势的差异，是判别活动状态的重要参考. 本部分基于聚类得到的簇中心数据表，描绘不同簇中心数据表的三轴加速度和角速度随时间的变化曲线，以此分析差异，判断活动状态. 具体过程如下.

1. 抽取簇中心数据表. 根据聚类结果，12 个最终的聚类簇中心 $\{\boldsymbol{\mu}_1^*, \boldsymbol{\mu}_2^*, \ldots, \boldsymbol{\mu}_{12}^*\}$ 对应的数据表为 $\{SY_1^*, SY_2^*, \ldots, SY_{12}^*\}$. 由此，选取这 12 个数据表，作为判别活动状态的代表.

2. 曲线图绘制. 选定数据表 SY_i^*，分别读取三轴加速度和三轴角速度，分别绘制三轴加速度和三轴角速度随时间步长的变化曲线图. 对于选定的人员，总计绘制 24 幅加速度和角速度随时间步长的变化曲线图.

3. 判别活动状态. 选定数据表 SY_i^*，根据绘制的运动轨迹和加速度、角速度的变化曲线图，综合判断该数据表的活动状态，将此类型作为整个聚类簇的活动状态，最终即可获得该名人员 60 组数据所对应的活动状态. 具体结果在下一节中展示.

3.4 问题结果

3.4.1 小波分解

使用 Haar 小波将人员数据表的信号转换成高频特征和低频特征，再对两个特征分贝进行 4 个统计特征的提取. 按照此小波分解过程，人员的一组数据最终被提取为 1×48 的表征向量 $\boldsymbol{STA}_j$，将 60 组数据提取成 60 个表征向量，再保存到同一份文件中，表示该人员的表征矩阵. 以人员 Person1 为例，其局部表征矩阵如图 2 所示.

	A	B	C	D	E	F	G	H	I
1	1.385635	2.019339	6.56359	-1.01728	-0.00099	0.209095	0.56487	-0.96777	0.093291
2	1.401618	1.961959	6.95121	-0.97555	-0.00135	0.234938	1.425472	-1.27953	0.13698
3	1.384364	0.04822	1.478499	1.031455	-0.0001	0.009426	0.135745	-0.15351	0.254143
4	1.346503	0.043011	1.482789	1.232637	#######	0.004768	0.028779	-0.07264	0.387044
5	1.385503	0.506556	3.359186	0.269487	0.000196	0.079201	0.473315	-0.41187	0.050925
6	1.373482	0.059323	1.627756	0.939021	#######	0.014021	0.130441	-0.32378	0.300775
7	1.358174	0.572017	3.28445	0.140158	0.000763	0.081669	0.344448	-0.36396	0.068945
8	1.387773	0.5438	3.287238	0.031582	0.000505	0.092639	0.636361	-0.49301	0.054091
9	1.354456	0.035757	1.596827	1.165329	-0.0002	0.00769	0.070519	-0.07305	0.191771
10	1.373251	0.041298	1.674739	1.167496	9.07E-07	0.007057	0.070784	-0.0333	0.196623
11	1.359928	0.051312	1.622121	1.046456	-0.00032	0.018616	0.114359	-0.47864	0.380544
12	1.366929	0.043449	1.61465	1.101226	0.000418	0.010318	0.093912	-0.06504	0.208655
13	1.34721	0.008201	1.382807	1.303	4.39E-05	0.002129	0.018369	-0.01409	0.451635
14	1.394118	0.556129	3.317102	0.169838	-0.0002	0.105038	0.407897	-0.505	0.033806

图2 Person1 的表征矩阵(局部)

在图 2 中，每行代表一组数据的表征向量，例如第一行表示 Person1 的 SY1 数据表的表征向量. 在表征向量中，每 8 列代表一个信号向量的高频统计特征和低频统计特征，共计六个信号向量的 48 个统计特征. 例如，第一行的第 1 至 8 列表示 Person1 的 SY1 数据表中 acc_x(x 轴加速度)经过小波分解后，得到的四个高频统计特征和四个低频统计特征. 在接下来的谱聚类中，每个表征向量(行数据)将作为一个数据点，经过一系列操作和变换，被聚类成 12 个簇.

3.4.2 谱聚类

给定人员，经过小波分解得到的表征向量需通过谱聚类方法被均匀地分成 12 簇，以初步达到判别人员活动状态的目的. 得到的最终聚类簇中心 $\{\mu_1^*, \mu_2^*, \ldots, \mu_{12}^*\}$ 由于维数过高，难以进行可视化，因此有必要采取针对性的处理措施.

主成分分析(Principal Component Analysis, PCA)是一种用于降维和特征提取的统计技术. PCA 的目的是通过线性变换将原始数据转换到新的坐标系中，使得数据在新坐标系中的投影能够最大化方差，广泛应用于高维特征的降维以达到可视化的目的. 将 PCA 的目标维数设置为 2，即对每个数据点 c_j，寻找并保留其方差最大的两个主成分，以这两个主成分作为二维平面图的两个坐标轴，由此，数据点能够以二维的形式可视化呈现. Person1 的 60 个数据点在经过 PCA 降维后的谱聚类结果如图 3 所示.

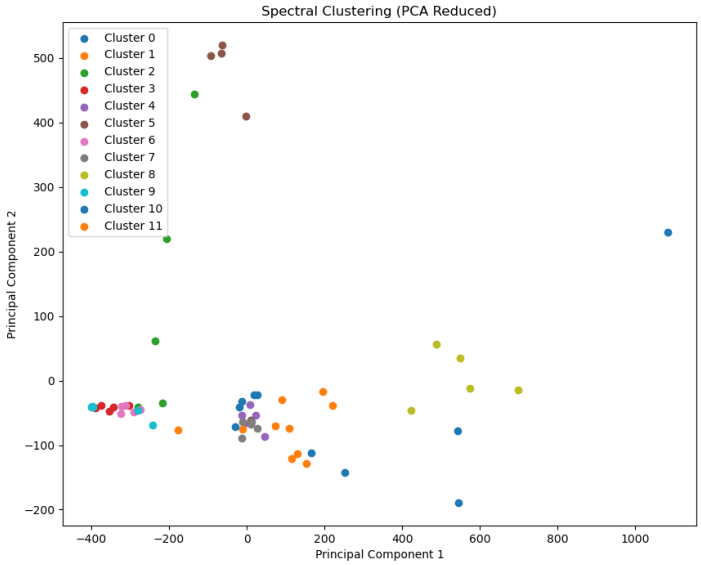

图3 Person1 的谱聚类结果

谱聚类仅能将 60 个数据点按照一定规律均匀分为 12 簇，但簇所属的活动状态仍然未知，因此需要采取一定方法对 12 个簇进行归类.

以 Person1 为例, 谱聚类的结果如表 1 所示.

表1 Person1 的谱聚类结果

簇序号	簇内包含的数据表号	簇中心数据表号
1	[7, 33, 38, 42, 58]	58
2	[1, 2, 15, 23, 36]	23
3	[21, 25, 29, 48, 53]	29
4	[13, 24, 20, 30, 39]	24
5	[5, 8, 46, 56, 59]	59
6	[3, 6, 11, 16, 55]	16
7	[9, 10, 12, 18, 57]	9
8	[14, 17, 28, 41, 47]	14
9	[19, 26, 43, 44, 50]	44
10	[22, 32, 34, 35, 40]	34
11	[27, 31, 37, 49, 52]	37
12	[4, 45, 51, 54, 60]	51

3.4.3 运动轨迹

谱聚类得到的 12 个簇分别存在 12 个簇中心及其对应的数据表, 为判断 12 个簇分别被所属的活动状态, 需要对这些簇中心进行活动状态的判断. 结合数据表的三轴加速度、三轴角速度和运功轨迹的推导方法以及可视化技术, 能够在一定程度上判断簇中心的活动状态.

得到聚类结果及其簇中心数据表号后, 读取 12 个簇中心数据表的数据, 描绘各自的运动轨迹, 结果如图 4 到图 15 所示.

运动轨迹图中, x 轴正方向为重力方向(竖直向下), y 轴正方向为人员前进方向(水平向前), z 轴正方向为垂直于 xy 平面指向身体一侧(水平向左).

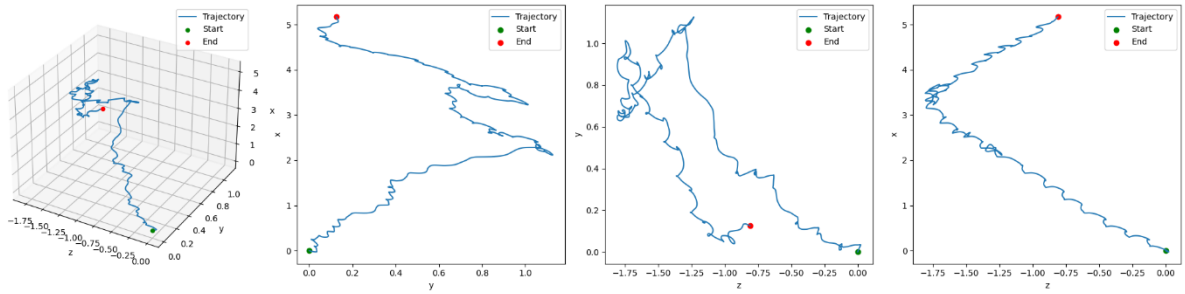

图4 1 号簇中心 SY58 的运动轨迹

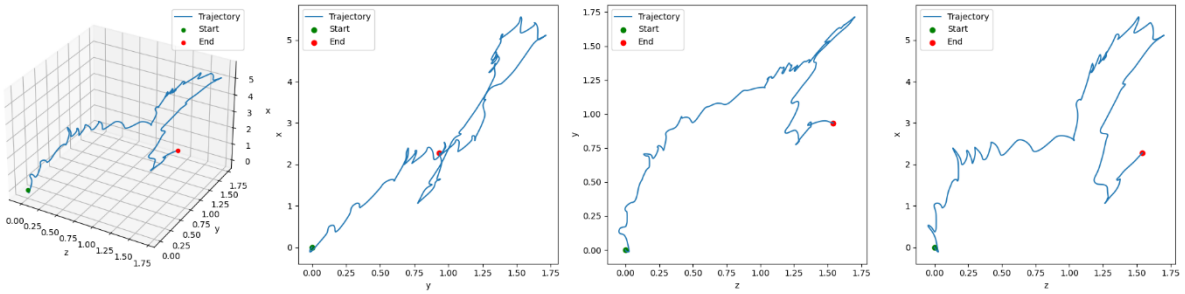

图5 2 号簇中心 SY23 的运动轨迹

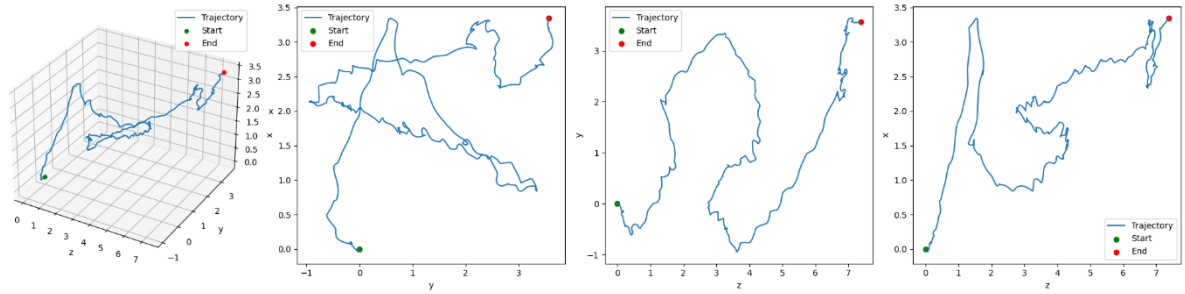

图6 3 号簇中心 SY29 的运动轨迹

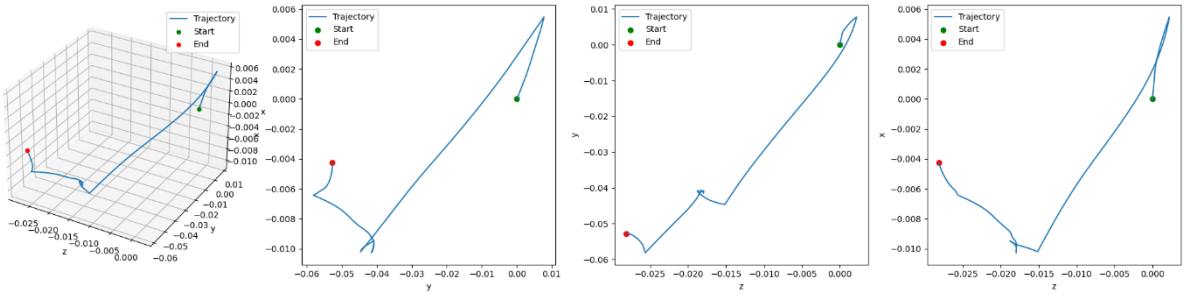

图7 4 号簇中心 SY24 的运动轨迹

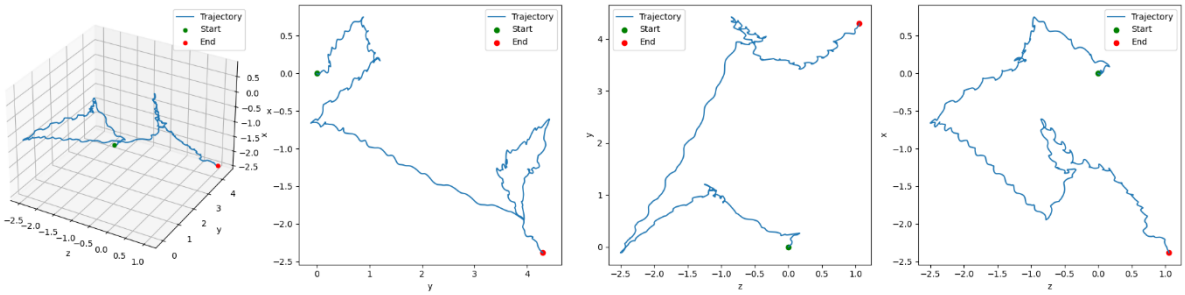

图8 5 号簇中心 SY59 的运动轨迹

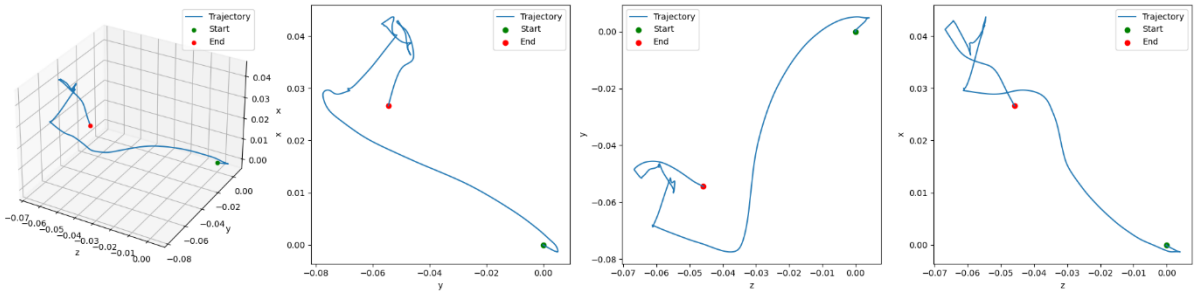

图9 6 号簇中心 SY16 的运动轨迹

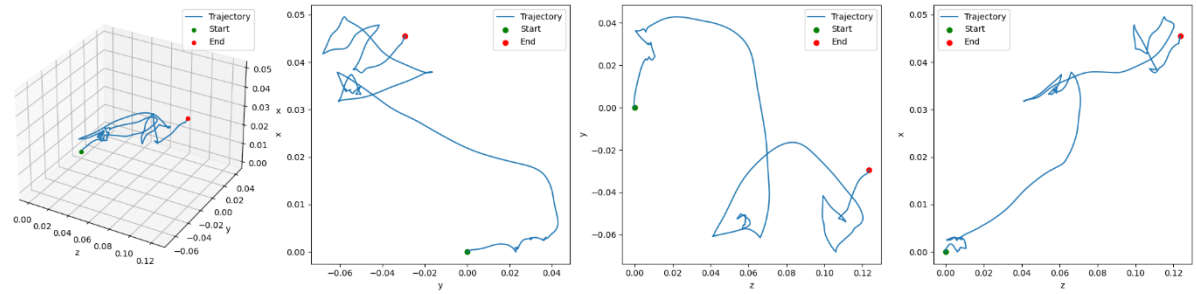

图10 7 号簇中心 SY9 的运动轨迹

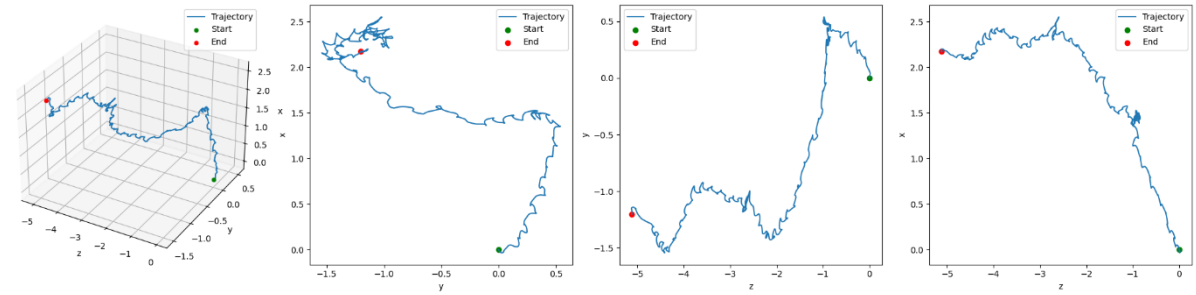

图11 8 号簇中心 SY14 的运动轨迹

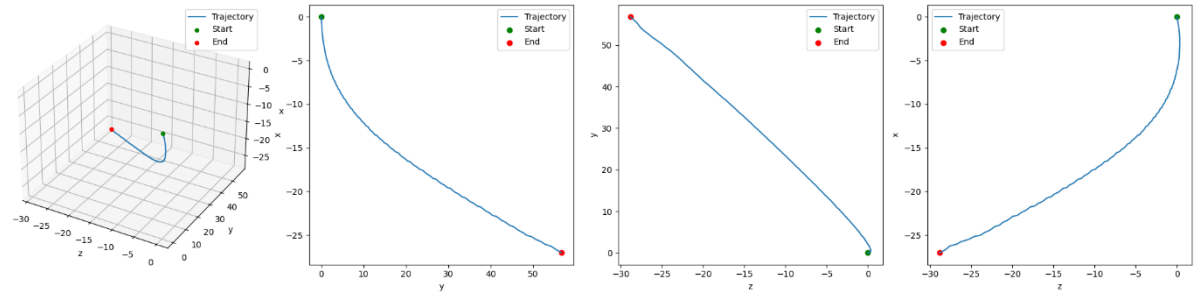

图12 9 号簇中心 SY44 的运动轨迹

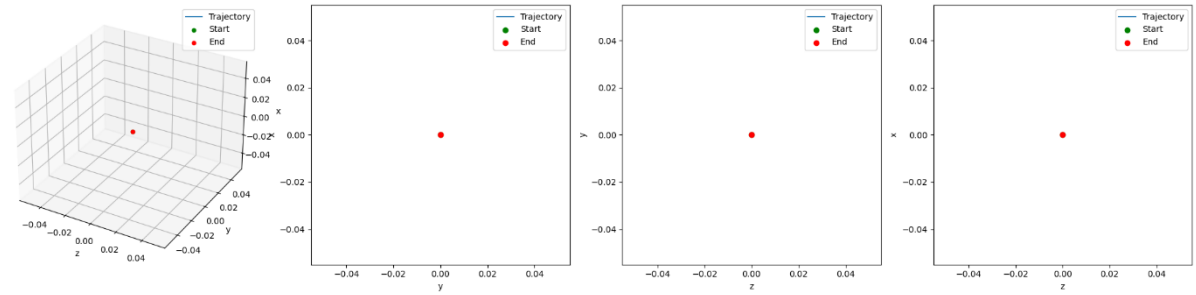

图13 10 号簇中心 SY34 的运动轨迹

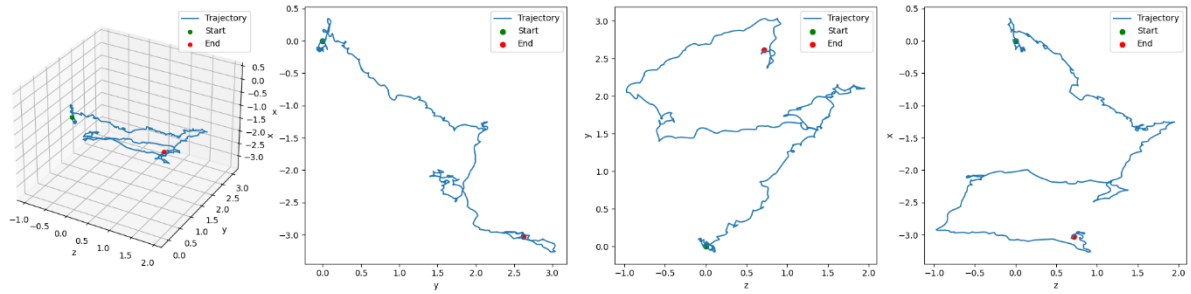

图14 11 号簇中心 SY27 的运动轨迹

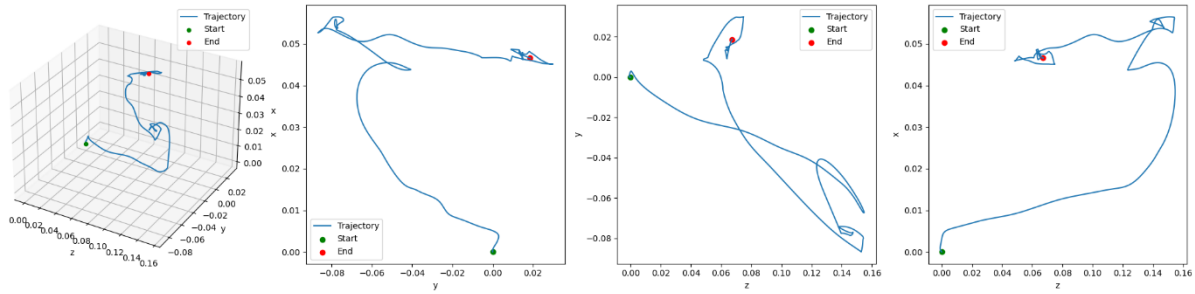

图15 12 号簇中心 SY51 的运动轨迹

3.4.4 变化曲线图

分析数据表的三轴加速度、三轴角速度随时间的变化曲线，能够在一定程度上判断簇中心的活动状态. 得到聚类结果及其簇中心数据表号后，读取 12 个簇中心数据表的数据，描绘表中三轴加速度和三轴角速度随时间的变化曲线，以 Person1 为例，结果如图 16 所示，图中上方的子图是三轴加速度的变化曲线，下方的子图是三轴角速度的变化曲线.

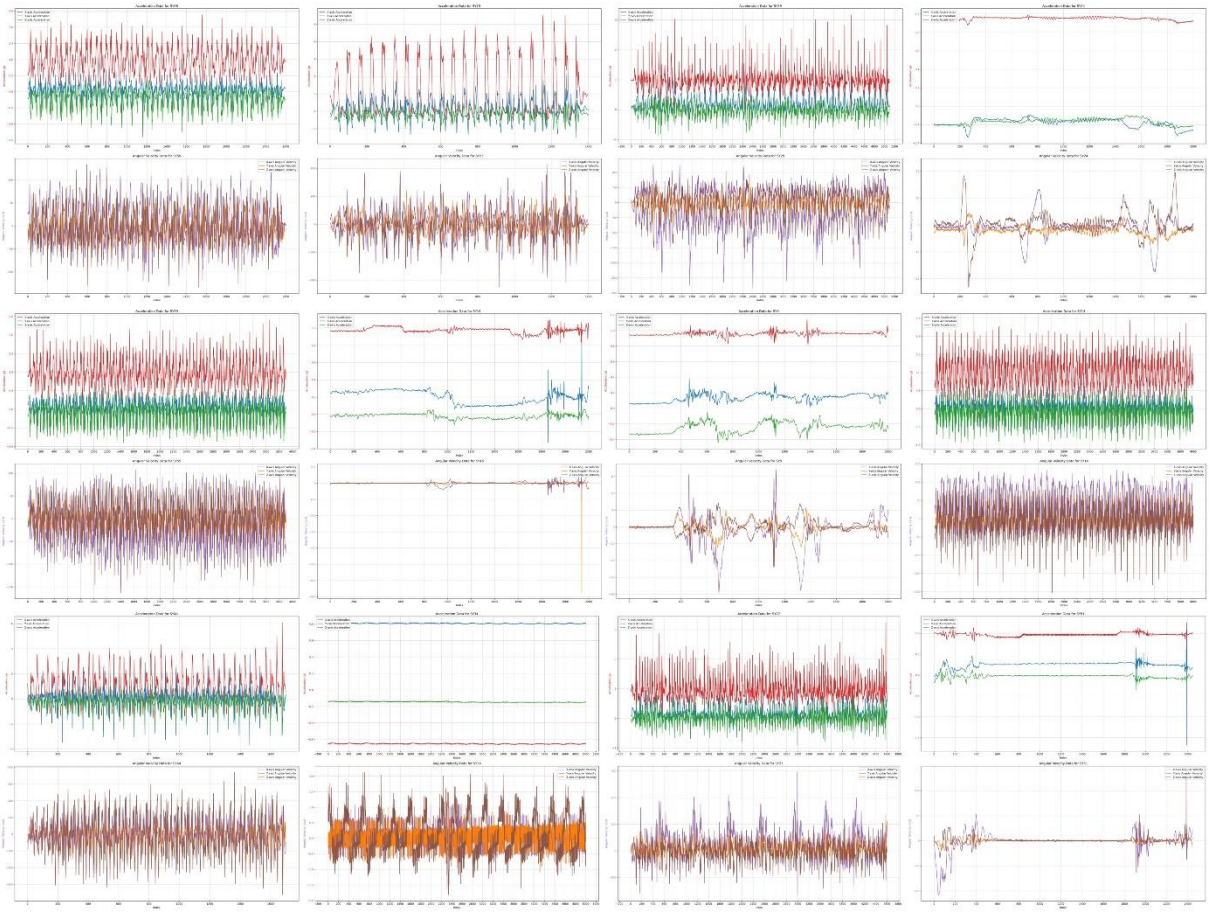

图16 Person1 的 12 个簇中心的变化曲线

3.4.5 问题一的求解结果

综合数据表的运动轨迹和变化曲线，得出三名人员各自的 60 组数据表对应的活动类别结果，如表 2 所示.

表2 三名人员的活动类别结果

活动编号	活动含义	Person1	Person2	Person3
1	向前走	[7, 33, 38, 42, 58]	[7, 10, 19, 23, 30]	[11, 24, 25, 31, 44]
2	向左走	[14, 17, 28, 41, 47]	[16, 21, 29, 40, 43]	[9, 14, 30, 37, 43]
3	向右走	[5, 8, 46, 56, 59]	[3, 20, 38, 51, 60]	[8, 23, 39, 42, 52]
4	步行上楼	[21, 25, 29, 48, 53]	[12, 24, 33, 44, 57]	[13, 27, 29, 51, 56]
5	步行下楼	[27, 31, 37, 49, 52]	[2, 26, 37, 46, 47]	[19, 35, 36, 49, 60]
6	向前跑	[19, 26, 43, 44, 50]	[1, 4, 48, 49, 50]	[10, 41, 57, 58, 59]
7	跳跃	[1, 2, 15, 23, 36]	[13, 27, 28, 34, 42]	[4, 5, 18, 22, 53]
8	坐下	[13, 24, 20, 30, 39]	[5, 11, 22, 54, 56]	[17, 21, 33, 34, 38]
9	站立	[9, 10, 12, 18, 57]	[6, 15, 25, 35, 58]	[2, 6, 12, 47, 48]
10	躺下	[22, 32, 34, 35, 40]	[31, 32, 39, 52, 59]	[3, 32, 50, 54, 55]
11	乘坐电梯向上移动	[3, 6, 11, 16, 55]	[8, 9, 36, 41, 55]	[1, 7, 26, 40, 45]
12	乘坐电梯向下移动	[4, 45, 51, 54, 60]	[14, 17, 18, 45, 53]	[15, 16, 20, 28, 46]

4 问题二的模型建立与求解

4.1 问题求解思路

问题二可使用的附件 2 数据共有 10×60=600 组，且每组数据具有固定标签(活动状态). 要求提取出 12 种活动状态的典型特征，并建立活动状态的判别模型. 显然，该问题是一个带标签的监督学习问题，首先确定 12 类活动状态的典型特征，并建立活动状态的分类模型，然后对 600 组数据进行活动状态的分类判断.

随机森林是一种基于统计分析的预测模型，属于监督学习范畴，是一个通过重采样技术的由多个决策树组成的组合分类器，决策树之间没有关联，其形成采用随机的方法. 每个决策树具有相同独立分布，采用随机方法选择输入变量，可以在减小相关性的同时保持强度不变，单棵决策树的分类能力很小，但由随机产生的决策树组成森林后具有较高的预测精度.

因此，本文拟采用随机森林方法完成 600 组数据的活动状态分类任务. 首先利用问题一求解方法中的小波分解步骤处理 600 组数据，得到每名人员的 60 组表征向量，提取出活动状态的典型特征并确定特征的重要性排序. 然后，构建包含多个决策树的随机森林，对 600 组表征向量进行分类判断. 随后，比较随机森林判别模型与问题 1 谱聚类分类模型的结果差异，并分析分类模型的性能. 最后利用随机森林判别模型完成附件 3 中 30 组数据的活动状态判定结果.

问题二研究框架如图 17 所示.

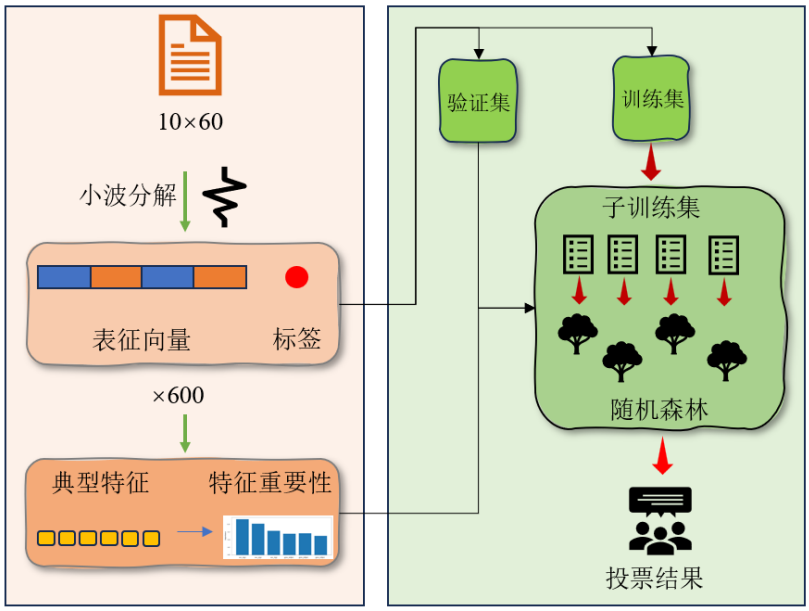

图17 问题二解决框架示意图

4.2 随机森林

本节首先利用小波分解方法得到 600 组数据的表征向量以及对应的标签，共同组成可输入随机森林的数据形式；然后介绍决策树方法的原理、节点分割过程、常用的不纯度函数以及决策树训练流程；最后介绍随机森林的基本思想、bootstrap 抽样方法、特征选择过程以及随机森林的训练流程.

4.2.1 小波分解

采用小波分解方法对每名实验人员 60 组数据 $a_i t_j (i \in \{1,2,\ldots,12\}, j \in \{1,2,\ldots,5\})$ 的六维数据进行压缩, 将一组数据压缩成一个表征向量, 具体过程参考 3. 2. 1 节.

首先以 $\boldsymbol{acc}_x$ 列信号为例, 得到其表征向量记为:

$$\begin{aligned}\boldsymbol{sat}(\boldsymbol{acc}_x) = [&mean(\boldsymbol{A}_{acc_x}), std(\boldsymbol{A}_{acc_x}), max(\boldsymbol{A}_{acc_x}), min(\boldsymbol{A}_{acc_x}), mean(\boldsymbol{D}_{acc_x}),\\ &std(\boldsymbol{D}_{acc_x}), max(\boldsymbol{D}_{acc_x}), min(\boldsymbol{D}_{acc_x})]\end{aligned} \tag{60}$$

向量维数为 8.

随后, 将 $\boldsymbol{acc}_x, \boldsymbol{acc}_y, ..., \boldsymbol{gyro}_z$ 六个信号的特征向量合并为一个向量, 记为 $\boldsymbol{S}_p\boldsymbol{A}_i\boldsymbol{T}_j (p \in \{4,...,13\}, i \in \{1,...,12\}, j \in \{1,...,5\})$, 表示第 p 个实验人员第 i 个活动状态的第 j 个表的表征向量. 例如, 11 号实验人员数据表 $a_4 t_1$ 的表征向量记为:

$$\boldsymbol{S}_{11}\boldsymbol{A}_4\boldsymbol{T}_1 = [\boldsymbol{sta}(\boldsymbol{acc}_x), \boldsymbol{sta}(\boldsymbol{acc}_y), \boldsymbol{sta}(\boldsymbol{acc}_z), \boldsymbol{sta}(\boldsymbol{gyro}_x), \boldsymbol{sta}(\boldsymbol{gyro}_y), \boldsymbol{sta}(\boldsymbol{gyro}_z)] \tag{61}$$

该向量包含的特征维度为 $8 \times 6 = 48$.

执行以上方法后, 得到 600 个数据表对应的 600 个表征向量, 所有表征向量的维度均为 48. 同时, 每个表征向量对应一个标签数据(活动状态) $label_i (i \in \{1,...,12\})$, 两者共同组成数据形式如下所示:

$$\boldsymbol{Data} = \begin{bmatrix} \boldsymbol{S_4}\boldsymbol{A}_1\boldsymbol{T}_1 & label_1 \\ \boldsymbol{S_4}\boldsymbol{A}_1\boldsymbol{T}_2 & label_1 \\ ... & ... \\ \boldsymbol{S}_{13}\boldsymbol{A}_{12}\boldsymbol{T}_4 & label_{12} \\ \boldsymbol{S}_{13}\boldsymbol{A}_{12}\boldsymbol{T}_5 & label_{12} \end{bmatrix}_{600\times(48+1)} \tag{62}$$

将该数据作为随机森林的数据集.

4.2.2 分类决策树

决策树是机器学习中的一个预测模型, 其中的非叶子节点代表对象的一个属性特征, 分支路径代表属性测试的一个输出, 叶节点代表对象的一个分类结果. 因变量若为分类变量, 则构建的决策树为分类树, 因变量若为连续变量, 则构建的决策树为回归树. 决策树算法不需要任何先验假设, 不假设类和其他属性服从一定的概率分布. 由于决策树分割依据的是观测值的顺序而不是数值的具体大小, 故即使数据中出现异常值, 对结果也不会造成太大的影响, 因此决策树算法对于噪声的干扰具有相当好的稳健性.

1. 分类决策树原理. 分类是通过统计学习得到一个目标函数(也称为分类器或分类模型) C, 每个属性 X 映射到一个事先定义的因变量 Y (Y 为离散变量), 通过数学符号表示为:

$$C : Dom(X_1) \times \cdots \quad (X_m) \to Dom(Y) \tag{63}$$

并令

$$\Omega = Dom(X_1) \times \cdots \quad (X_m) \to Dom(Y) \tag{64}$$

对于一个给定的分类模型C和一个给定空间Ω的概率测度P，计算满足误分率函数

$$R_p(C)=P[C(X_1,...,Y)] \tag{65}$$

其中C是取值最小的函数.

分类决策树可按递归方式定义如下:

$$c(X_1,...,T)=\begin{cases} label(T) & T\text{为叶子节点} \\ c(X_1,...,X_m) & T\text{为非叶子节点，}\ q(T,T_j)(x_i)=True \end{cases} \tag{66}$$

其中, 取值最小的函数C进行分类, $q(T,T_j)(x_i)$为从节点T到节点T_j包含分割变量x_i的分割谓项.

最终得到分类决策树:

$$D_{T_r}(x_1,...,x_m)=c[x_1,...,x_m,Root(T_r)] \tag{67}$$

其中$Root(T_r)$为决策树T_r的根节点.

2. 分类决策树分割. 构造分类树最重要的问题是选择最佳分割, 用分割前后记录的类分布来度量, 适用的度量方法是不纯度.

为确定测试条件的效果, 引入增益的概念, 即父节点和子节点间不纯度的差. 用t_p,t_L,t_R分别表示父节点、左子节点和右子节点. 用不纯度函数$i(t)$度量最大化子节点的同质性.

因为对于任何可能的分割$x_j\le x_j^R, j=1,...,M$，父节点t_p的不纯度都是常数, 因此最大化左、右子节点的同质性等价于最大化如下的不纯度增益函数$\Delta i(t)$:

$$\Delta i(t)=t_p-E[i(t_c)] \tag{68}$$

其中t_c表示t_p的子节点. 假设P_L,P_R分别是左右子节点的概率(对于样本来说是记录比例), 且有$P_L+P_R=1$, 则可以得到$\Delta i(t)=i(t)-P_Li(t_L)-P_Ri(t_R)$.

因此, 对于每个节点, CART(classification and regression tree)方法是使得下式达到最大化的问题:

$$\underset{x_j\le x_j^R,j=1,...,M}{argmax}\ [i(t_p)-P_Li(t_L)-P_Ri(t_R)] \tag{69}$$

上式表明, CART 通过搜索所有自变量所有可能的值使不纯度增益$\Delta i(t)$最大化以获得最优分割. 决策树的分类实际上就是选择最大化增益的测试条件, 也等价于最小化不纯性度量的加权平均.

3. 不纯度函数的定义方法. 对于不纯度函数$i(t)$的定义, 可主要采用 Gini 指数法、信息增益法和增益比率法等.

1)Gini 指数法. Gini 指数属于使用最广泛的一种分割规则. Gini 的不纯度函数$i(t)$可表示如下:

$$i(t)=1-\sum_{j=1}^{J}\{p(j\mid t)\}^2 \tag{70}$$

其中，$p(j|t)$表示在节点t中第j类的重要条件概率，对于样本来说，表示属于第j类记录的所占比例，J为Y的分类总数.

可以得到如下不纯度增益$\Delta i(t)$:

$$\Delta i(t) = -\sum_{j=1}^{J} p^2(j|t_p) + P_L \sum_{j=1}^{J} p^2(j|t_p) + P_R \sum_{j=1}^{J} p^2(j|t_p) \tag{71}$$

因此, Gini 指数最优分割就是寻找使得下式达到最大的分割:

$$\arg\max_{x_j \le x_j^R, j=1,\cdots,M} \left[-\sum_{j=1}^{J} p^2(j|t_p) + P_L \sum_{j=1}^{J} p^2(j|t_p) + P_R \sum_{j=1}^{J} p^2(j|t_p) \right] \tag{72}$$

2)信息增益法. 条件熵$H(Y|X)$表示在已知随机变量X的条件下随机变量Y的不确定性. 随机变量X给定的条件下随机变量Y的条件熵$H(Y|X)$，定义为X给定条件下Y的条件概率分布的熵对X的数学期望:

$$H(Y|X) = \sum_{i=1}^{n} p_i H(Y|X = x_i),\ p_i = P(X = x_i), i = 1,...,n \tag{73}$$

信息增益表示得知特征X的信息而使得类Y的信息的不确定性减少的程度.

特征A对训练数据集D的信息增益$G(D,A)$，定义为集合D的经验熵$H(D)$与特征A给定条件下D的经验条件熵$H(D|A)$之差，即:

$$G(D,A) = H(D) - H(D|A) \tag{74}$$

3)增益比率. 为了抵消因较大定义域的分类变量引起的偏差, Quinlan(1986)提出了信息增益的修正方法——增益比率法:

$$G_R(D,A) = G(D,A) / H_A(D) \tag{75}$$

即信息增益$G(D,A)$与训练数据集D关于特征A值的熵$H_A(D)$的比值.

本文算法中使用 Gini 指数法，从所有可能划分中找出 Gini 指数最小的划分，使用此特征对训练样本集进行划分以构建活动状态分类决策树.

4. 分类决策树算法流程. 分类决策树的构建方法可描述为：给定训练集$D = \{\omega_1,...,\omega_N\}$，其中$\omega_i (i = 1,...,N)$，记$v$为属性分裂方法，对独立同分布的$N$个样本，查找误分率函数$R_P(D_{T_r})$最小的一个分类树$T_r$.

分类决策树算法流程如下所示:

在节点T用v寻找属性X

1) if T 分裂

2) 把数据集D分割为$D_1, D_2,...,$，对T标记分裂变量X

3) 构建子节点T的节点$T_1, T_2,...$，并记$edge(T, T_i)$，对应预测值记为$q(T, T_i)$

4) for each $i \in \{1,...,N\}$

5) 构建树(T_i, D_i, V)

6) End for each

7) Else

8) T标记数据集D的多数分类标签

---

9) End if

4.2.3 随机森林判别模型

决策树算法虽然是一种很好的分类模型, 但其分类规则复杂, 容易产生过拟合现象; 其次由于决策树算法沿路径递归分割, 在叶子节点处的记录越来越少, 对于叶子节点最终分类判断较难, 很难得到高质量的结果.

为了解决决策树存在的问题, Leo Breiman 提出了随机森林理论, 很好的解决了过拟合现象, 不仅能保证精度而且能并行计算, 因此非常适合需要精确分类的活动状态识别领域.

1. 随机森林基本原理. 随机森林(Random Forest, RF)是以决策树为个体学习器构建的集成学习方法. 决策树作为随机森林的基本分类单元, 以类似于流程图的树形结构构建模型, 并从根节点、非叶子结点和叶子节点的顺序对样本进行分类. 决策树采用自顶向下的递归方式, 从树的根节点开始进行样本属性比较测试, 根据选取的属性值确定分支, 形成非叶子节点, 每个非叶子节点进行属性值的比较测试, 然后根据测试给定的属性值确定对应分支, 依次递归, 直到在决策树的叶子节点得到样本的分类类别.

随机森林的基本思想是利用 bootstrap 抽样方法从训练样本中抽取 K 个子训练集, 对每个子训练集建立一个决策树模型, 每个待测试样本经过所有决策树模型会产生 K 种分类结果, 最后所有决策树进行投票, 根据投票结果确定样本类型. 需要注意的是, 传统决策树在选择划分属性时, 选择的是一个最优的属性, 而随机森林在构建决策树时, 是从所有属性特征中随机抽取出部分属性特征, 然后从中选取一个最优的划分属性特征.

2. Bootstrap 抽样方法. Bootstrap 抽样方法是一种自助采样法, 即在有 n 个样本的训练集中有放回的抽取 n 个样本作为一个采样集. 因此, 对连续 n 次抽样, 一个样本始终没有被抽中的概率为 $(1-1/n)^n$, 由于样本集中各个样本是相互独立的, 由下式可知训练集中约有 36. 8%的样本未出现在采样集中.

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n \approx 0.368 \tag{76}$$

3. 随机特征选取. 随机特征选取是指随机森林为了提高预测准确度, 引入随机性来减小相关系数, 同时保持强度不变. 随机性的使用有随机选择输入变量和随机组合输入变量.

对每一个节点在特征变量集合中随机选取一组变量进行分割, 依据随机选取的特征变量而非所有的特征变量, 采用 CART 方法生成决策树. 决策树构建之后, 使用随机森林分类的众数, 共同表决判断出预测分类结果, 此时构建随机森林过程中随机选择的是固定的分类变量, 为增加决策树的随机性, 用 bootstrap 方法产生样本. 另外, 决策树的强度和相关性也受到特征变量数量的影响, 特征变量数据减小, 决策树的相关性较弱, 但强度增加. 由于决策树的每个节点用到的仅仅是特征变量集合的一个子集, 故能大幅度地减少计算成本.

本文输入变量数选取为 $\text{int}(\log_2 M+1)$, 其中 M 表示总的输入变量个数, int() 表示取小于或等于 $\log_2 M+1$ 的最大整数.

4. 随机森林算法流程. 设样本数量为 n, 每个样本的属性个数为 M, m 为大于零且小于 M 的整数并且有 $m=\text{int}(\log_2 M+1)$, 随机森林的具体算法流程如下所示.

输入: 训练集 S, 测试集 X, 其他参数

输出: 测试集 X 的样本所属的类别

1) 采用 bootstrap 抽样方法对训练集 S 进行随机采样, 产生 T 个子训练集 $S_1, S_2, \cdots, S_T$

2) 利用 T 个子训练集, 生成相应的决策树 $C_1, C_2, \cdots, C_T$

3) 在每个非叶子节点上选择属性前, 从 M 个属性中随机选取 $m = \text{int}(\log_2 M + 1)$ 个属性作为当前节点的分裂属性集, 并将具有最大熵增益的属性作为当前分裂点, 对该节点进行分裂, 直至生成决策树

4) 依次构建 T 个决策树, 构成随机森林

5) 对于测试集样本 X, 利用每个决策树模型进行测试, 得到对应的分类类别 $C_1(X), C_2(X), \cdots, C_T(X)$

6) T 个决策树进行投票, 依据投票准则确定测试样本 X 所属类别.

随机森林算法的训练流程示意图如图 18 所示.

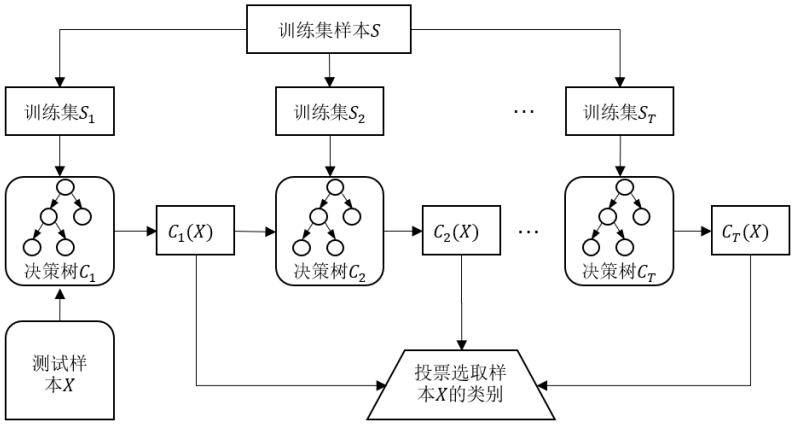

图18 随机森林算法示意图

将随机森林作为问题二人员活动状态的判别模型, 对附件 2 中 10 名实验人员的活动状态进行判定, 数据集中共包括 600 组带标签的样本, 按照 8: 2 的比例将数据集拆分为训练集(480 组数据)和验证集(120 组数据)对模型进行训练和验证.

4.3 问题结果

4.3.1 典型特征提取

经过对 12 种活动状态的特征分析, 提取到 6 种典型特征, 即三轴上的线加速度和角速度, 对这些特征的重要性进行计算, 结果如图 19 所示.

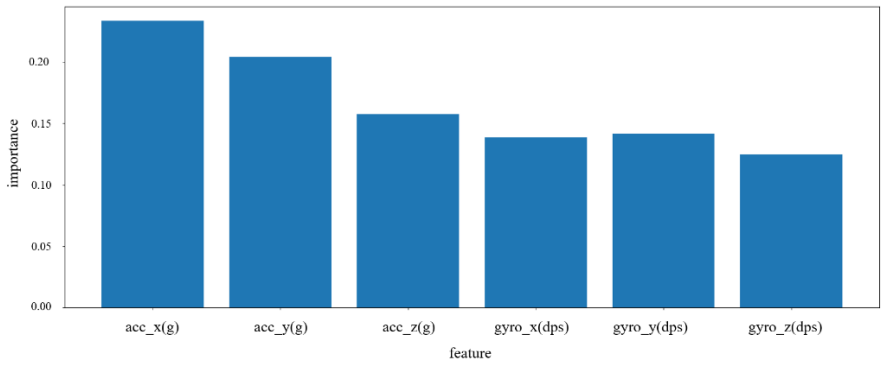

图19 特征重要性排序

其中$acc_x(g)$特征的重要度最高，而该特征的重要性由其 8 维元素(高、低频特征及均值、方差、最大、最小值)共同决定. 因此，在决策树每个节点分裂前的随机特征选取过程中，特征子集的构建将有较大概率从 48 个特征中选中$acc_x(g)$包含的 8 个特征，并组成包含$m=\text{int}(\log_2 M+1)=\text{int}(\log_2 48+1)=\text{int}(5.59+1)=6$个特征的集合，然后从这个子集中选择最佳的分裂特征. 这种方法有助于减少模型的过拟合，并增加模型的多样性.

为理解决策树的分支过程，随机森林中的两个决策树分支示意图如图 20 所示.

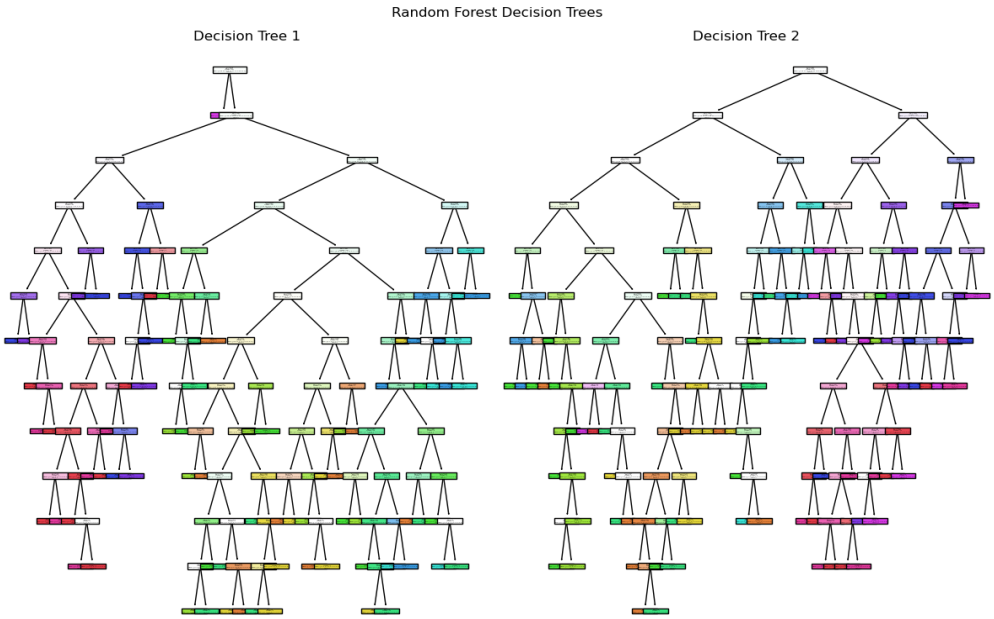

图20 决策树分支示意图

第二个决策树中，根节点表示 4 <= -0. 18，左、右分支分别为 4 <= 19. 21、4 > 19. 21，左分支节点的左右子节点为 4 <= 2. 98 与 4 > 2. 98，依次类推进行分支直到满足分类终止条件为止.

4.3.2 结果验证、比较与分析

利用问题 1 谱聚类分类模型对附件 2 中的 10 名实验人员数据进行分类，得到每名实验人员的分类结果如图 21 所示.

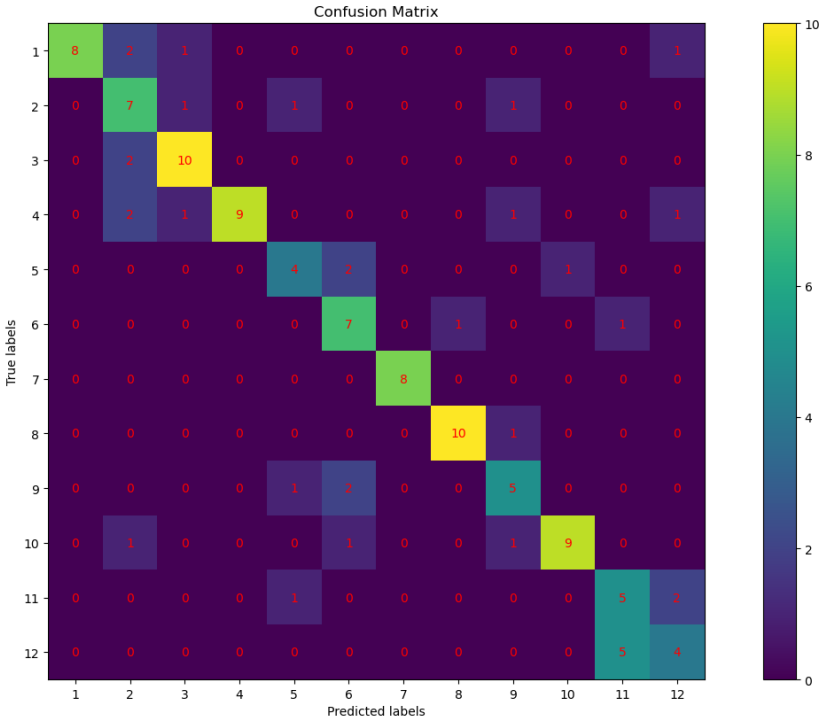

图21 谱聚类分类模型混淆矩阵

为比较问题 2 随机森林判别模型和问题 1 谱聚类分类模型的结果，以及分析分类模型的准确度，给出两个模型在验证集 120 组数据上的混淆矩阵，如图 22 所示.

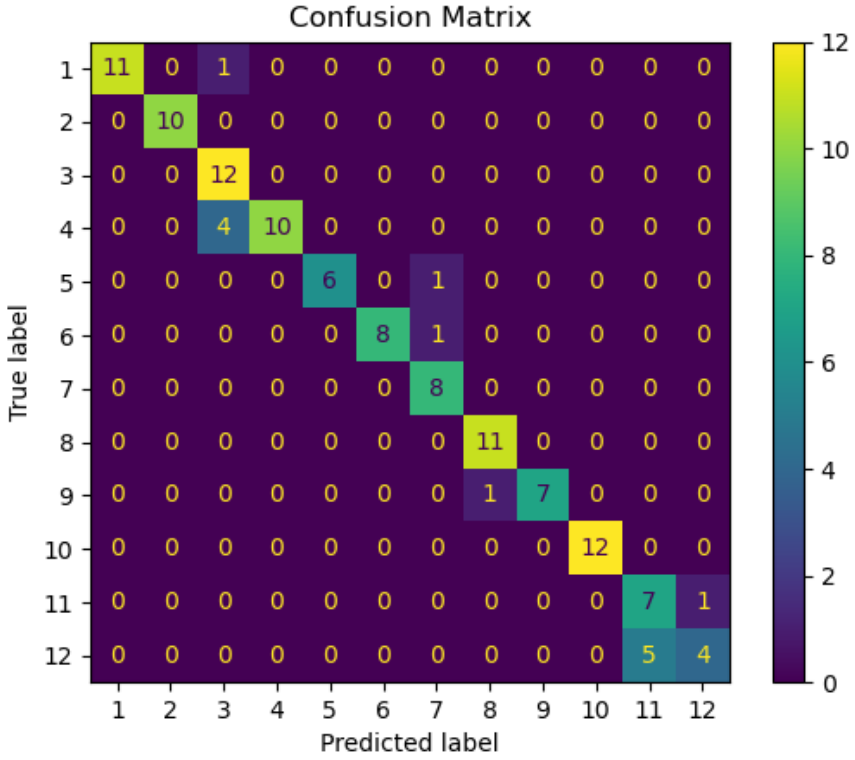

图22 随机森林判别模型混淆矩阵

根据两个混淆矩阵和准确率定义:

$$Accuracy = \frac{\text{识别正确的个数}}{\text{样本总个数}} \tag{77}$$

可分别计算谱聚类模型和随机森林模型对不同活动类型分类时的准确率:

$$Accuracy_{\text{谱聚类}} = \frac{86}{120} = 71.67\%$$

$$Accuracy_{\text{随机森林}} = \frac{106}{120} = 88.33\%$$

根据精确率定义，计算两个模型每个类别的精确率如图 23 所示.

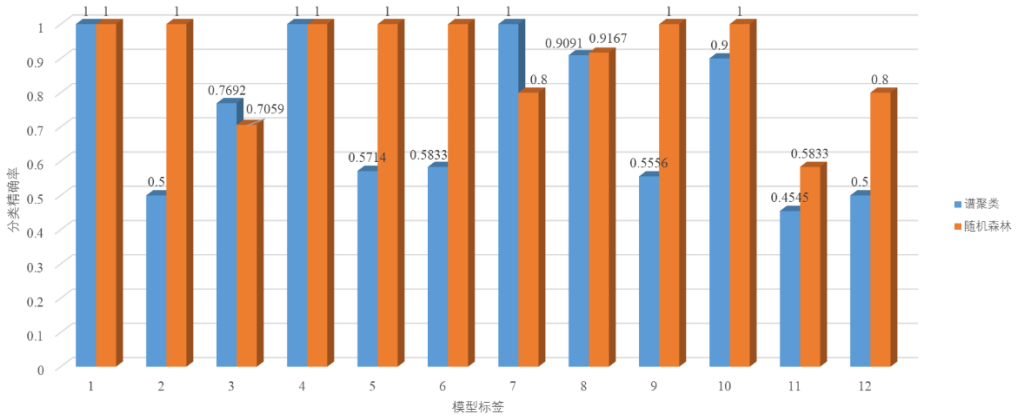

图23 各活动类别精确率对比示意图

由计算结果对比可知，使用随机森林模型可以得到更加准确的判别结果，此外可知，使用谱聚类模型对活动状态分类时，分类的准确度可达 71. 67%，表明本文提出的谱聚类模型具有较好的分类效果.

4.3.3 人员活动状态判定

利用随机森林判别模型，对附件 3 中 30 组数据进行状态判断，结果如表 3 所示.

表3 某人员 30 次活动判断结果

活动状态	判别状态	活动状态	判别状态	活动状态	判别状态
SY1	5	SY11	9	SY21	6
SY2	1	SY12	7	SY22	2
SY3	7	SY13	4	SY23	8
SY4	12	SY14	3	SY24	5
SY5	7	SY15	4	SY25	2
SY6	10	SY16	1	SY26	8
SY7	2	SY17	4	SY27	8
SY8	6	SY18	5	SY28	5
SY9	7	SY19	8	SY29	6
SY10	10	SY20	8	SY30	5

5 问题三的模型建立与求解

5.1 问题求解思路

问题三的子问题 1 是对同一活动状态下, 不同人员之间的差异进行分析. 在求解问题一时, 采用了小波分解的方法对人员的每组数据进行了压缩, 使得不同数据量的数据表最后都输出为相同维度的表征向量, 其维度为 1×48. 因此子问题 1 可以采用相同的思想, 采用小波分解将附件 2 中的 10 名人员的数据表数据压缩为 1×48 的表征向量, 以此作为差异分析的输入数据. 而后进行差异程度的评价, 使用统计领域常用的皮尔逊相关系数作为评价指标, 构建不同人员之间的相关性矩阵, 并使用可视化工具展示人员之间的差异程度. 此外, 考虑到附件 1 人员活动数据不含相关活动状态, 且问题一的分类结果存在不准确性, 因此在差异分析的过程中, 只考虑附件 2 的 10 名人员, 不纳入附件 1 的 3 名人员.

问题三的子问题 2 是分析传感器数据和人员的关系, 需要判断传感器数据的人员来源. 然而, 传感器数据的类型多样、规模庞大, 人员的身份信息和运动特征隐藏在这大规模活动数据之中. 因此, 为了建立传感器数据和人员身份的之间复杂的关联关系, 有必要采用深度学习模型对人员的运动特征进行学习. 而在深度学习模型中, LSTM 模型广泛应用于处理时序数据. 由此, 选择使用 LSTM 深度学习模型对该子问题进行学习, 最后对未知人员的身份进行判别.

5.2 差异分析

首先按照 5. 3. 1 的小波分解方法, 对附件 2 的 10 名人员的每组数据进行小波分解, 得到的表征向量记为 $\boldsymbol{STA}_{ijk}$, 其中 $i \in [4,13]$ 表示表示人员的编号, $j \in [1,12]$ 表示活动状态的编号, $k \in [1,5]$ 表示同一类型的实验次数编号. 由于人员对同一活动状态进行了 5 次实验, 为综合考虑这 5 次实验的结果, 将这 5 个表征向量 $\boldsymbol{STA}_{ij1}$ 至 $\boldsymbol{STA}_{ij5}$ 合并为一个新的表征向量, 记为:

$$\boldsymbol{STA}_{ij} = [\boldsymbol{STA}_{ij1}, \boldsymbol{STA}_{ij2}, \boldsymbol{STA}_{ij3}, \boldsymbol{STA}_{ij4}, \boldsymbol{STA}_{ij5}] \tag{78}$$

该表征向量的维数为 48×5=240.

其次, 针对某一活动状态 j, 计算附件 2 中两名人员 m、n 之间的皮尔逊相关系数, 计算方式为:

$$r_{jm,n} = \frac{\sum_{i=1}^{n}(STA_{mj}(i) - \frac{1}{n}\sum_{k=1}^{n} STA_{mj}(k))(STA_{nj}(i) - \frac{1}{n}\sum_{k=1}^{n} STA_{nj}(k))}{\sqrt{\sum_{i=1}^{n}(STA_{mj}(i) - \frac{1}{n}\sum_{k=1}^{n} STA_{mj}(k))^2 \sum_{i=1}^{n}(STA_{nj}(i) - \frac{1}{n}\sum_{k=1}^{n} STA_{nj}(k))^2}} \tag{79}$$

式中, n 为表征向量的维数, 相关系数越大, 说明两名人员数据的相似程度越高. 由此可得活动状态 j 的相关性矩阵:

$$\boldsymbol{R}_j = \begin{bmatrix} r_{j1,1} & \cdots & r_{j1,10} \\ \vdots & r_{jm,n} & \vdots \\ r_{j10,1} & \cdots & r_{j10,10} \end{bmatrix} \tag{80}$$

最后, 根据活动状态 j 的相关性矩阵, 绘制该类型下的相关性矩阵热力图, 用于显示两名人员在同一活动状态下的差异程度.

5.3 特征学习

LSTM 模型处理时序数据的特点是: LSTM 按照逐个时间步长处理输入数据, 通过内部结构中的门控机制(遗忘门、输入门、输出门)来捕捉时间序列中的动态特征和长期依赖关系. 经过转换, LSTM 将输出高维特征向量, 此特征向量表示区分不同人员的运动特征, 然后高维特征向量经过全连接层和 Softmax 层, 输出人员的身份信息.

在实际训练过程中, 对时间序列进行前后两个方向的学习能够提升模型的训练准确度, 因此选择 Bi-LSTM 模型进行运动特征的学习. 具体来说, Bi-LSTM 具有以下特点:

1)双向信息处理. Bi-LSTM 可以同时从序列的前后两个方向进行信息处理. 它由两个 LSTM 层组成, 一个处理正向序列, 另一个处理反向序列, 然后将两个 LSTM 层的输出拼接在一起. 这使得 Bi-LSTM 能够捕捉到序列中每个时间步的前后文信息, 比单向 LSTM 更能全面地理解整个序列的上下文关系.

2)提高对复杂模式的识别能力. 活动状态数据通常包含复杂的时间依赖模式, 比如速度、加速度等. 这些模式可能不仅依赖于过去的状态, 也依赖于未来的状态. Bi-LSTM 能够利用双向的信息来更好地识别和理解这些复杂模式.

本文设计的 Bi-LSTM 模型架构如图 24 所示.

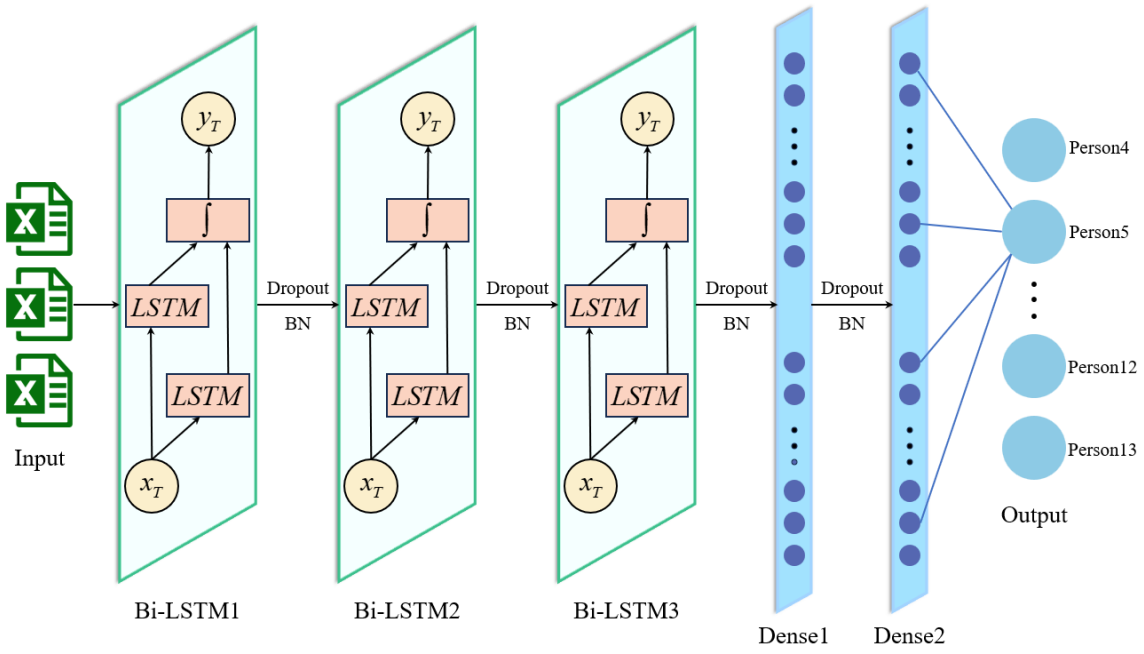

图24 模型架构

模型具体过程如下.

1. 超参数设置. 如表 4 所示.

表4 模型超参数

超参数	含义	取值
Noise_factor	数据增强, 添加噪声	0. 005
Bi-LSTM1	第一层 Bi-LSTM 单元数量	128
Bi-LSTM2	第一层 Bi-LSTM 单元数量	64
Bi-LSTM3	第一层 Bi-LSTM 单元数量	32

超参数	含义	取值
Dense1	第一层 Dense 神经元数量	64
Dense2	第一层 Dense 神经元数量	10
Learning_rate	学习率	0. 001
Dropout_rate	神经元参数丢弃比例	0. 3
Batch_size	批次大小	32
Epochs	训练轮数	50

2. 数据增强. 训练之前对附件 2 的十位人员的 600 组数据数据进行数据增强, 具体分为添加噪声和随机裁剪, 增强后的数据规模为原始数据规模的三倍. 按照 4: 1 的比例将数据集划分为训练集和验证集.

3. 训练模型. 对设定好的 Bi-LSTM 模型进行训练, 期间记录每个轮次的训练集准确率和验证集准确率, 以及训练集损失值和验证集损失值.

4. 输出结果. Bi-LSTM 模型训练完毕后, 将附件 5 的未知人员信息数据作为输入, 模型的输出是每组数据的人员编号, 每一名人员存在 12 个编号. 对于其中的某一人员, 将 12 个编号出现次数最多的编号作为该人员的身份信息.

5.4 问题结果

5.4.1 差异分析

12 种活动状态下的相关性矩阵热力图如图 25 所示, 其中颜色越深, 表示两名人员在此活动状态下的皮尔逊相关系数越高.

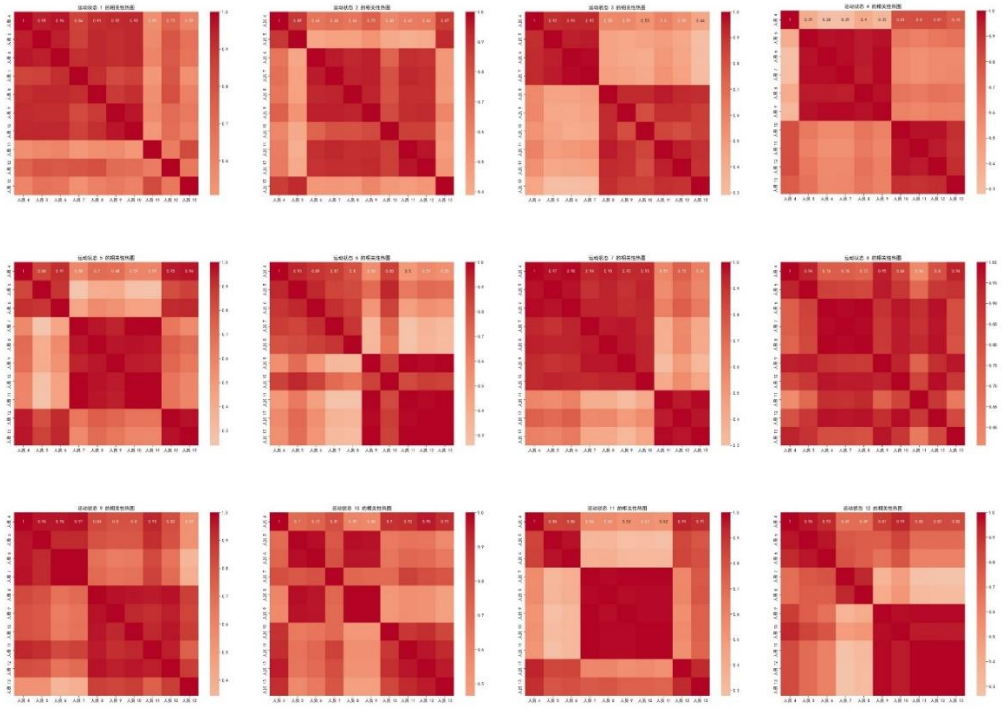

图25 各活动状态下的相关性矩阵热力图

从图中可以看到, 每个活动状态下的热力图均存在深色区域和浅色区域, 即使是较难看出差别的”坐下”、”站立”、”躺下”等静止活动也不例外, 且浅色区域的分布各不相同, 因此能以基于小波分解表征向量的皮尔逊相关系数为评价标准, 认为同一活动状态

下不同人员之间存在差异. 状态 2 下, 年龄相似、身高相似但体重差异较大的人员 5 和人员 11 之间的相关性仅为 0. 39, 存在较大差异. 状态 11 下, 年龄相似, 体重接近但身高差异较大的人员 6 和人员 8 之间的相关性仅为 0. 39, 存在较大差异. 状态 2 下, 身高相似、体重相似但年龄差异较大的人员 5 和人员 12 之间的相关性仅为 0. 41, 存在较大差异. 这说明同一活动状态下不同人员由于诸如身高等宏观因素, 确实会存在活动状态的特征差异. 因此, 我们希望 Bi-LSTM 通过学习数据表数据, 从而得到复杂的特征向量, 最终完成人员画像的分类任务.

5.4.2 特征学习

训练完毕的 Bi-LSTM 模型精度和损失随轮次的变化如图 26 所示. 其训练集和验证集准确率均稳步上升, 其训练集和验证集损失均稳步下降. 模型取得了良好的训练效果. 同时明显的上升趋势说明模型性能还有进一步提升空间，增加训练轮数会有更好的表现.

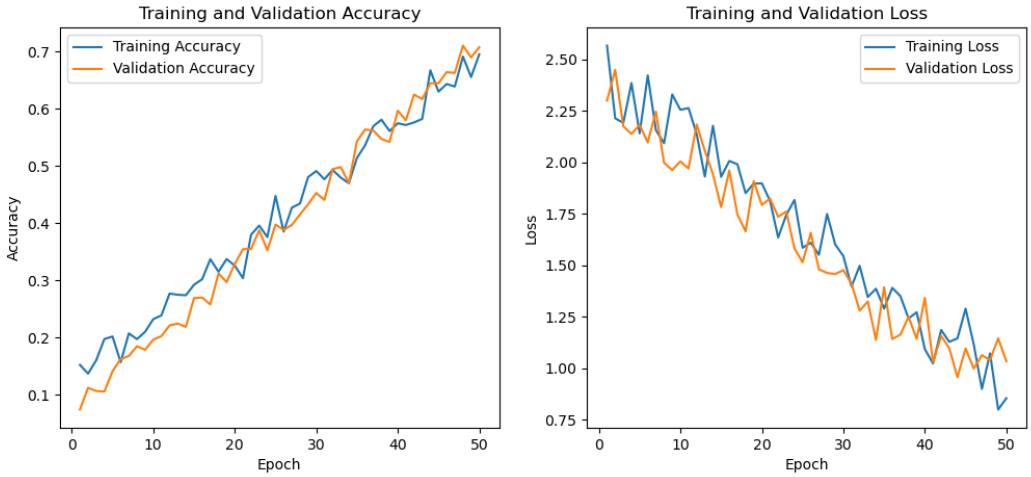

图26 模型精度和损失随轮次的变化

将附件 5 的 5 个未知人员的数据表作为测试集, 输入到训练完毕的 Bi-LSTM 模型中, 输出为每张数据表的预测人员标签, 将出现次数最多的预测标签作为最终的未知人员判别结果, 结果如表 5 所示.

表5 模型预测结果

人员	a1 t1	a2 t1	a3 t1	a4 t1	a5 t1	a6 t1	a7 t1	a8 t1	a9 t1	a10 t1	a11 t1	a12 t1	判别结果
U1	10	10	10	7	10	5	10	10	10	10	10	4	**P10**
U2	7	8	7	7	7	7	9	7	7	7	12	4	**P7**
U3	8	6	6	8	6	6	6	5	3	6	12	6	**P6**
U4	9	9	9	9	8	5	9	9	4	9	13	9	**P9**
U5	13	8	13	13	9	13	9	10	13	8	13	4	**P13**

根据以上判别结果，问题三的判别结果如表 6 所示.

表6 问题三结果

人员	判别结果
Unknow1	**Person10**
Unknow2	**Person7**
Unknow3	**Person6**
Unknow4	**Person9**
Unknow5	**Person13**

6 模型评价与推广

6.1 模型的评价

6.1.1 模型的优点

(1) 模型充分结合实际，简化具体活动参数条件，考虑了诸多重要因素得到合理的模型，如：人员差异，时空干扰等. 这样得到的模型贴合实际，具有较高的应用价值，可以推广到人员活动状态分类识别;
(2) 模型运用系统思想，抓住影响活动状态分类问题的重要因素，将复杂的分类问题转化为简单的模式识别问题，合理设置参数，模型的输出结果符合题目要求，能解决实际问题;
(3) 本文使用的机器学习算法具有效率高，分类效果好，鲁棒性强等优点，对于求解不同情景下的分类模型非常适用;
(4) 本文得到的活动状态识别和人员画像具有效率高、输出稳定等特点，基本不存在低效率等问题，在现有条件下能有效提高分类效率.

6.1.2 模型的不足

(5) 实际应用中，复杂的运动轨迹和人员动作的连续性可能也是重要的因素，但本文未能考虑到这些因素的影响，一定程度上影响了模型的准确性;
(6) 本文提出的模型对于现有条件使用效果较好，由于时间问题没有对其他情况进行检验. 对于其他情形(如：扩大样本特征维度、增加测试人员)，可能无法达到较好的效果.
(7) 本文应该适当扩充一下对数据噪声的思考，实际上只在运动轨迹计算的滤波过程中用了去噪的方法，但是噪声可能影响之前进行谱聚类的结果，还会导致运动轨迹绘制有些不合理地方.

6.2 模型的推广

在研究活动状态与人员特征的关系方面，可以进行人员画像和数据来源识别. 具体应用实践，比如手机运动软件设置方面，可以深入了解不同人员的活动状态及其特征，验证模型的准确性，并探索传感器数据在个体识别和分类中的应用.

参考文献

[1] 张亚杰. 基于小波分解的 AVOA-DELM 月径流时间序列预测模型及应用[J]. Pearl River, 2022, 43(7).

[2] Wang S, Gao C, Zhang Q, et al. Research and experiment of radar signal support vector clustering sorting based on feature extraction and feature selection[J]. IEEE Access, 2020, 8: 93322-93334.

[3] 向涛, 李涛, 赵雪专, 等. 基于随机森林的精确目标检测方法[J]. Application Research of Computers, 2016, 33(9).

[4] Kpienbaareh D, Wang J, Luginaah I, et al. A Geospatial Approach to Assessing the Impact of Agroecological Knowledge and Practice on Crop Health in a Smallholder Agricultural Context[J]. The Professional Geographer, 2023, 75(4): 618-635.

[5] Chiew K L, Tan C L, Wong K S, et al. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system[J]. Information Sciences, 2019, 484: 153-166.

[6] TU J, LIU Q, LIN Y, et al. Coupling Relationship between Tree Growth and Water Consumption of Pinus elliottii in Subtropical Red Earth Area[J]. Journal of Natural Resources, 2008, 23(4): 685-693.

[7] 倪祥龙, 石长安, 麻曰亮, 等. 基于 Bi-LSTM 的电子装备故障预测方法研究[J]. Aero Weaponry, 2022, 29(6).

[8] 娄洋歌, 邹锦涛, 周学平. 基于隐马尔科夫模型的动向目标运动趋势预测[J]. 舰船电子工程, 2021, 41(8): 30-33.

[9] Breiman L. Random forests[J]. Machine learning, 2001, 45: 5-32.

附　录

附录 A:　主要数学计算程序/关键代码

问题 1　小波分解、谱聚类、运动轨迹

小波分解

```
# 读取 Excel 数据
data = pd. read_excel('filepath')

# 初始化一个字典，用于存储压缩后的特征
compressed_features = {} # 对每一列进行小波分解和特征提取
for column in data. columns:
    # 进行 1 级小波分解（可以根据需要调整分解级别)
    coeffs = pywt. wavedec(data[column], 'db1', level=1)

    # 提取每个系数的统计特征
    features = []
    for coeff in coeffs:
        mean_value = coeff. mean()
        std_value = coeff. std()
        max_value = coeff. max()
        min_value = coeff. min()
        features. extend([mean_value, std_value, max_value, min_value])
```

谱聚类

```
# 读取 Excel 文件
file_path = 'filepath'
df = pd. read_excel(file_path)

# 定义读取块大小
chunk_size = 60
num_clusters = 12
points_per_cluster = 5

# 存储所有块的列表
all_chunks = []

for start_row in range(0, len(df), chunk_size):
    # 读取每块数据
    chunk = df. iloc[start_row: start_row + chunk_size, 1: ]. values
    # 存储块数据
    all_chunks. append(chunk)
```

```
    # 初始化一个列表来存储每个块的聚类结果
    cluster_results = []

    for block_index, data in enumerate(all_chunks):
        # 初始谱聚类
        spectral_clustering = SpectralClustering(n_clusters=num_clusters, affinity='nearest_neighbors', random_state=42)
        initial_labels = spectral_clustering. fit_predict(data)

        # 调整聚类结果
        final_labels = np. zeros(len(data), dtype=int) - 1  # 初始化为-1
        remaining_indices = list(range(len(data)))

        # 计算每个点到所有其他点的距离
        distances = euclidean_distances(data, data)

        for cluster_num in range(num_clusters):
            selected_points = []
            if remaining_indices:
                # 选择一个初始点
                initial_point_index = remaining_indices[0]
                selected_points. append(initial_point_index)
                remaining_indices. remove(initial_point_index)

                # 选择距离初始点最近的四个点
                sorted_indices = np. argsort(distances[initial_point_index, remaining_indices])[: points_per_cluster - 1]
                for idx in sorted_indices:
                    selected_points. append(remaining_indices[idx])
                remaining_indices = [idx for idx in remaining_indices if idx not in selected_points]

                # 设置最终标签
                for point_idx in selected_points:
                    final_labels[point_idx] = cluster_num

        # 检查是否有未被分配的点
        for idx in remaining_indices:
            # 找到最近的已分配簇
```

```
            nearest_cluster = np. argmin([np. min(distances[idx, final_labels == cluster]) for cluster in range(num_clusters)])
            final_labels[idx] = nearest_cluster

        # 使用 PCA 将数据降到二维
        pca = PCA(n_components=2)
        data_2d = pca. fit_transform(data)

        # 计算轮廓系数
        silhouette_avg = silhouette_score(data, final_labels)
        print(f'Block {block_index + 1} Silhouette Score:    {silhouette_avg: . 2f}')

        # 输出每个簇的点以及每个簇的中心点对应的序号
        for cluster_num in range(num_clusters):
            cluster_indices = np. where(final_labels == cluster_num)[0]
            cluster_points = data[cluster_indices]
            cluster_point_indices = np. arange(1, 61)[cluster_indices]    # 生成 1 到 60 的序号

            # 计算簇的中心点
            cluster_center = np. mean(cluster_points, axis=0)

            # 找到距离中心点最近的点的序号
            distances_to_center = np. linalg. norm(cluster_points - cluster_center, axis=1)
            nearest_point_index = cluster_indices[np. argmin(distances_to_center)]
            nearest_point_serial = np. arange(1, 61)[nearest_point_index]    # 生成 1 到 60 的序号

            # 输出簇的信息
            print(f'Block {block_index + 1} - Cluster {cluster_num}: ')
            print(f'    Points:    {cluster_point_indices}')
            print(f'    Center Point Serial:    {nearest_point_serial}')

        # 存储每个块的聚类结果
        cluster_results. append((block_index + 1, final_labels, silhouette_avg))
```

运动轨迹

```
# 读取 Excel 文件
df = pd. read_excel('filepath')

# 提取数据并转换单位
```

```
a_x = (df['acc_x(g)']. values - 1) * g  # 转换为 m/s²，并假设静止时 acc_x=g
a_y = df['acc_y(g)']. values * g
a_z = df['acc_z(g)']. values * g

omega_x = np. radians(df['gyro_x(dps)']. values)  # 转换为弧度/秒
omega_y = np. radians(df['gyro_y(dps)']. values)
omega_z = np. radians(df['gyro_z(dps)']. values)

# 高通滤波器去除加速度中的低频漂移
def highpass_filter(data, cutoff, fs, order=4):
    nyquist = 0. 5 * fs
    normal_cutoff = cutoff / nyquist
    b, a = butter(order, normal_cutoff, btype='high', analog=False)
    y = filtfilt(b, a, data)
    return y

cutoff_frequency = 0. 5  # 0. 1 Hz 的高通滤波器
a_x = highpass_filter(a_x, cutoff_frequency, sampling_rate)
a_y = highpass_filter(a_y, cutoff_frequency, sampling_rate)
a_z = highpass_filter(a_z, cutoff_frequency, sampling_rate)

# 使用卡尔曼滤波器对加速度和角速度进行平滑处理
def apply_kalman_filter(data):
    kf = KalmanFilter(initial_state_mean=0, n_dim_obs=1, initial_state_covariance=1,
observation_covariance=1)
    state_means, _ = kf. filter(data)
    return state_means. flatten()

a_x = apply_kalman_filter(a_x)
a_y = apply_kalman_filter(a_y)
a_z = apply_kalman_filter(a_z)

omega_x = apply_kalman_filter(omega_x)
omega_y = apply_kalman_filter(omega_y)
omega_z = apply_kalman_filter(omega_z)

# 积分计算角度
theta_x = np. cumsum(omega_x * dt)
theta_y = np. cumsum(omega_y * dt)
theta_z = np. cumsum(omega_z * dt)
```

```
# 定义旋转矩阵函数
def rotation_matrix(theta_x, theta_y, theta_z):
    Rx = np. array([[1, 0, 0],
                    [0, np. cos(theta_x), -np. sin(theta_x)],
                    [0, np. sin(theta_x), np. cos(theta_x)]])
    Ry = np. array([[np. cos(theta_y), 0, np. sin(theta_y)],
                    [0, 1, 0],
                    [-np. sin(theta_y), 0, np. cos(theta_y)]])
    Rz = np. array([[np. cos(theta_z), -np. sin(theta_z), 0],
                    [np. sin(theta_z), np. cos(theta_z), 0],
                    [0, 0, 1]])
    return Rz @ Ry @ Rx

# 检测零速度更新（简单阈值法)
def detect_zero_velocity(omega_x, omega_y, omega_z, threshold=0. 2):    # 调整阈值
    return np. sqrt(omega_x ** 2 + omega_y ** 2 + omega_z ** 2) < threshold

zupt = detect_zero_velocity(omega_x, omega_y, omega_z)

for i in range(1, len(a_x)):
    # 当前时刻的旋转矩阵
    R = rotation_matrix(theta_x[i], theta_y[i], theta_z[i])

    # 将加速度从物体坐标系转换到惯性坐标系
    a_inertial = R @ np. array([a_x[i], a_y[i], a_z[i]])

    # 分解加速度到惯性坐标系的分量
    a_x_inertial, a_y_inertial, a_z_inertial = a_inertial

    # 积分计算速度
    v_x[i] = v_x[i - 1] + a_x_inertial * dt
    v_y[i] = v_y[i - 1] + a_y_inertial * dt
    v_z[i] = v_z[i - 1] + a_z_inertial * dt

    # 积分计算位移
    s_x[i] = s_x[i - 1] + v_x[i] * dt
    s_y[i] = s_y[i - 1] + v_y[i] * dt
    s_z[i] = s_z[i - 1] + v_z[i] * dt
```

问题 2 随机森林

```
# 读取 Excel 文件
```

```
df = pd. read_excel('filepath')

# 提取特征和标签
X = df. iloc[: , : 6]. values  # 前 6 列是特征
y = df. iloc[: , -1]. values  # 最后一列是标签

# 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=46)

# 创建并训练 Random Forest 模型
rf = RandomForestClassifier(n_estimators=50, random_state=46)
rf. fit(X_train, y_train)
# 保存模型
#import pickle
# 序列化模型到 pickle 文件
#filename = 'random_forest_model. pkl'
#pickle. dump(rf, open(filename, 'wb'))

# 在测试集上评估模型
y_pred = rf. predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy:  {accuracy: . 2f}")

#特征重要性
import matplotlib. pyplot as plt
feature_importances = rf. feature_importances_
feature_names = ['acc_x(g)', 'acc_y(g)', 'acc_z(g)', 'gyro_x(dps)', 'gyro_y(dps)', 'gyro_z(dps)']

#绘制决策树，并且记录剪枝条件
from sklearn. tree import plot_tree, export_text
plt. figure(figsize=(12, 8))
for i in range(1):
    plt. subplot(3, 2, i+1)
    tree = rf. estimators_[i]
    plot_tree(tree, feature_names=feature_names, class_names=[str(c) for c in np. unique(y)], filled=True, max_depth=5)
    plt. title(f'Decision Tree {i}')

#绘制混淆矩阵
from sklearn. metrics import confusion_matrix, ConfusionMatrixDisplay
```

```
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=np. unique(y))
disp. plot()
plt. title('Confusion Matrix')
plt. show()
```

问题 3 相关性分析

```
# 读取原始数据
df = pd. read_excel(file_path)

# 假设每个活动状态有 10 个人，每个人有 5 个数据点
num_people = 10
num_data_points = 5
person_start = 4  # 人员从 4 开始

# 添加 '人员' 和 '活动状态' 列
df['人员'] = (df. index // num_data_points) % num_people + person_start
df['活动状态'] = (df. index // (num_people * num_data_points)) + 1
ode_minus'] = False

# 遍历每个活动状态，计算相关性并绘制热图
for state in df['活动状态']. unique():
    df_state = df[df['活动状态'] == state]

    # 创建一个数据框来存储每个人员的数据
    data_by_person = pd. DataFrame()

    for person in range(person_start, person_start + num_people):
        person_data = df_state[df_state['人员'] == person]. drop(columns=['活动状态',
'人员']). values. flatten()
        data_by_person[f'人员 {person}'] = person_data

    # 计算相关性矩阵
    corr_matrix = data_by_person. corr()

    # 绘制热图
    plt. figure(figsize=(10, 8))
    sns. heatmap(corr_matrix, annot=True, cmap='coolwarm', center=0)
    plt. title(f'活动状态 {state} 的相关性热图')
```