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）：202418001007

所属学校（请填写完整的全名）：国防科技大学

参赛队员（打印后签名）：1. 徐心雨

2. 李俊杰

3. 王昭茜

指导教师或指导教师组负责人（打印后签名）：谢徐超

日期：2024 年 7 月 12 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。）

基于运动传感器的人体活动状态识别

摘要

随着科技的迅猛发展，人体动作识别技术成为了人工智能和人机交互领域的研究热点。本工作研究基于传感器的人体动作识别系统，通过加速度计和陀螺仪收集的传感数据，感知人体的姿态、角度和方向的变化，进而设计识别算法，进行人体动作识别。在本项工作中，给出了 13 名实验人员携带运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。根据上述数据建立模型，进行人体动作识别，主要工作分为以下三个部分：

第一部分：针对问题一，我们设计了一个聚类模型将每组实验数据中的 60 个样本进行分类。分析表明，12 种动作状态可分为静止和运动两大类。静止状态的加速度曲线平缓，而运动状态则波动较大。我们选用了 K-means 聚类算法，它适用于固定数量的簇划分，无需预设簇数。我们对数据进行预处理，得到合成加速度、加速度与加速度在 YOZ 平面上投影的夹角、加速度在 YOZ 平面上投影与 Z 轴的夹角，特征向量包括方差、平均值、中位数、5% 和 95% 分位点以及它们的差值，这些特征有助于区分动作状态。聚类中心通过迭代更新，直至中心位置稳定或达到最大迭代次数 1200 次。聚类结果应用于三组测试数据，每组包含 12 个动作状态的各 5 个样本。模型能准确分类样本，实现对动作状态的有效识别。

第二部分：针对问题二的多分类模型建立，我们首先需对原始数据进行标准化预处理，这有助于平衡各特征量级，避免不同量级对模型训练产生影响。关键特征提取阶段，我们将基于三个轴方向的加速度和角速度数据，提取反映信号波动和剧烈程度的六个特征值，包括方差、平均值、中位数以及 5% 与 95% 分位点差值。随后，将数据集按照 8:2 比例划分为训练集和测试集，确保数据分布的一致性和代表性，最终训练准确率达到了 85%。使用训练好的模型对附件 3 中的 30 次活动状态进行预测，以验证其泛化能力。

第三部分：问题三通过分析实验人员在相同活动状态下的差异，并探讨其与年龄、身高和体重的关系，以主成分分析（PCA）和皮尔逊相关系数为工具。首先，从附件中提取并预处理了 13 位实验人员的生理和活动状态数据，进行标准化以消除量纲影响。利用 PCA 降维，提取活动状态的一维特征，然后计算并分析了这些一维特征与个人生理特征之间的皮尔逊相关系数，以判断活动状态与个体生理特征的相关性。进一步地，通过附件 5 中的数据，我们构建了 10 位实验人员在 12 类活动状态下的一维特征画像。结合问题 2 中建立的判别模型，使用 KNN 算法对实验人员进行分类判断，评估并选择分类准确度最高的模型。

关键词： 人体动作识别；K-means；随机森林；PCA

目录

1. 问题综述

1.1 问题背景

人体动作识别指的是使用各种设备采集人的日常行为活动信息，通过数据预处理、特征提取和搭建分类算法等步骤，得到的计算机模型能够准确识别人体动作。目前，人体动作识别系统主要分为基于视频的系统和基于传感器的系统。基于视频的系统利用红外光谱仪、摄像机等设备，通过拍摄视频或图像来辨析人体动作。然而，这种方法受限于图像模糊、人体遮挡等问题，影响了识别的准确性。此外，视频数据的存储和计算成本非常高，并且采集设备位置固定，使用场景受限，涉及用户隐私问题。这些问题使得基于视频的动作识别在实际应用中存在较大挑战。相比之下，基于传感器的系统采集时间序列数据进行人体动作识别，不仅成本低、操作便捷，还能有效保护用户隐私，因而逐渐成为研究热点。早期，研究者利用加速度计和陀螺仪收集传感数据，并采用两阶段连续隐马尔可夫模型进行人体动作识别。尽管初期传感器体积大、精准度低，限制了其推广应用，但近年来传感技术的飞速发展，使得传感器向微型化、精准化方向发展，制造成本降低，并能嵌入智能手机、智能手表等便携设备中。因此，基于传感器系统的人体动作识别技术越来越普遍，逐渐应用于日常生活场景中。

传统的数学统计方法在人体动作识别中主要依赖于统计特征的提取和分析。这些方法包括线性回归、判别分析、主成分分析（PCA）和独立成分分析（ICA）等。它们通过对数据进行预处理、归一化和特征工程，手工提取出能够代表人体动作的统计特征。这些特征可能包括均值 [?]、方差 [3, 5]、标准差、峰值 [6]、频率分布等。通过这些特征，传统数学统计方法构建分类器或回归模型，以实现对人体动作的识别和分类。然而，这些方法通常依赖于专家的领域知识，需要手工设计特征，且在面对复杂和高维数据时表现有限。相比于传统数学统计方法，一些主流运动轨迹的识别算法比如传统决策树法，动态时间规整 (dynamic time warping，DTW) 技术 [7]，支持向量机 [8] 和隐马尔科夫模型 (Hidden Markov Model，HMM)[9] 等方法在人体动作识别中具有显著优势。首先，传统机器学习方法能够自动从数据中提取和选择特征，减少了对人工干预和专家知识的依赖，提升了模型的适用性和灵活性。其次，这些方法在处理复杂和高维数据时表现更为优越，能够通过优化算法和调参技术提高模型的准确性和稳定性。而传统统计方法在面对复杂数据模式时往往显得力不从心。此外，传统机器学习方法具有更好的鲁棒性，能够有效应对数据中的噪声和异常值，提高模型的泛化能力，这使得其在实际应用中更具优势。尽管深度学习技术在近年来取得了显著的进展 [10, 11, 12, 13]，并在人体动作识别中展现出强大的性能，但传统机器学习方法依然具有不可忽视的优势，特别是在解释性和透明性方面。深度学习模型通常被视为“黑箱”，其内部决策过程难以理解和解释，而传统机器学习方法，如 KNN 和随机森林，通过特征重要性、决策树结构等方式，可

以更清晰地展示模型的决策依据，使得结果更具可解释性和可信度。此外，传统机器学习方法对数据量和计算资源的需求相对较低，适合数据量较小或计算资源有限的场景。而深度学习方法通常需要大量标注数据和高性能计算资源进行训练，传统机器学习方法在数据不足的情况下仍能取得较好的性能，并且训练和推理速度较快，适用于实时性要求高的应用场景。

1.2 问题提出

对于人体动作识别问题，比较重要的几个要素分别是特征提取、模型训练和模型评估。为了有效识别和分类人体的各种活动状态，我们需要考虑如下问题：

在满足下列两点约束下：

- 特征提取的有效性：设计高效的算法，从多维时间序列数据中提取能够显著区分不同动作的特征。
- 模型训练的鲁棒性和性能：在有限的有标签数据和大量无标签数据的条件下，训练出准确率高、泛化能力强的模型。

从特征提取、模型训练和系统实现角度考虑，解决以下三个问题：

- 附件 1 中有 3 名实验人员的运动数据，包含每名实验人员每种活动状态的 5 组加速度计和陀螺仪数据，但实验时未记录数据所代表的活动状态。请根据附件 1 提供的活动数据（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，在论文中将分类结果（编号）填入表 1。
- 附件 2 中有 10 名实验人员的活动数据，包含每位实验人员每种活动状态的 5 组加速度计和陀螺仪数据，但实验时记录了每组数据所代表的实验人员的活动状态。请根据附件 2 提供的活动数据（每人 60 组数据）提取 12 类人员活动状态的典型特征，建立人员活动状态的判别模型，并利用提出的模型开展以下验证工作：
  - 进一步运用问题 1 的分类模型对该 10 名实验人员数据进行分类（此时，不考虑实验人员的活动状态标签），比较问题 2 中判别模型和问题 1 的分类模型的结果，分析采用分类模型对不同活动类型分类时的分类准确度。
  - 附件 3 中收集有某实验人员 30 次活动的状态数据，请运用提出的判别模型，给出该人员的活动状态，在论文中将结果填入表 2。
- 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。请分析不同人员的同一活动状态是否存在差异？活动状态数据与实验人员

的年龄、身高、体重有无关系，能否使用活动传感器数据进行人员画像。进一步，附件 5 中给出了问题 2 的 10 位实验人员中的 5 位的某次活动数据，数据包含了每人的 12 类活动状态，使用提出的模型判断他们分别最可能来源于问题 2 中哪一名实验人员。在论文中将判别结果填入表 3。

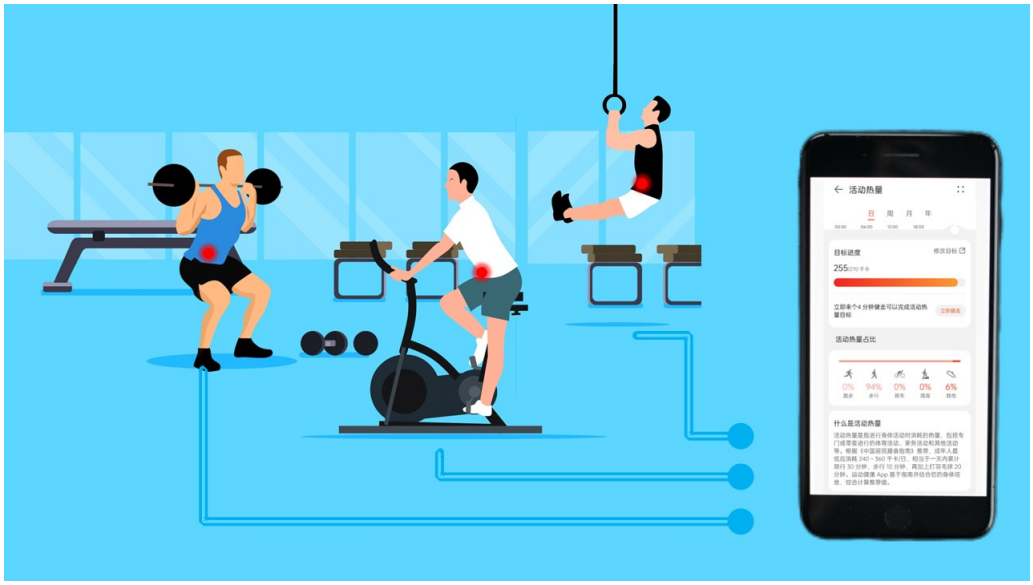

图 1: 人体动作识别

1.3 资料条件

附件提供了人体动作识别的工作原理和模型构建的方案需求，各文件的详细说明如下：

- 附件 1.xlsx：该文件提供了 3 名实验人员在 12 种活动状态下的运动数据，包括加速度计和陀螺仪的数据。其中，acc_x, acc_y, acc_z, gyro_x, gyro_y, gyro_z 等指标可能是模型分析的重点。
- 附件 2.xlsx：该文件包含 10 名实验人员的活动数据，每位实验人员在 12 种活动状态下的运动数据，包括加速度计和陀螺仪的数据，记录了每组数据所代表的活动状态。此数据将用于提取典型特征和建立判别模型。
- 附件 3.xlsx：该文件收集了某实验人员 30 次活动的状态数据，用于验证判别模型的准确性。
- 附件 4.xlsx：该文件包含了参与实验的 13 位实验人员的年龄、身高、体重等数据，用于分析不同人员的同一活动状态是否存在差异，以及活动状态数据与年龄、身高、体重之间的关系。

- 附件 5.xlsx：该文件包含问题 2 的 10 位实验人员中的 5 位某次活动数据，用于验证模型能否判断出最可能来源于哪一名实验人员。

通过以上附件提供的数据和详细说明，我们考虑基于加速度计和陀螺仪的时间序列数据进行数据清洗以及机器学习识别，建立人体活动状态和对应时间序列数据的映射模型。

图 2: MEMS 加速度计传感器

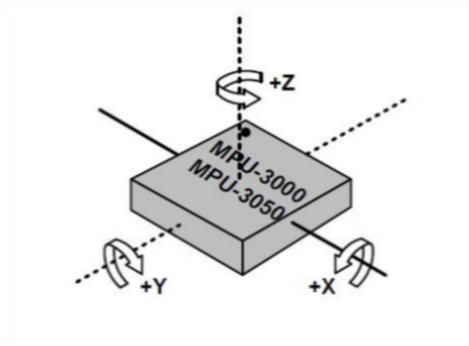

图 3: 陀螺仪

2. 模型假设与符号说明

2.1 模型基本假设

(1) 假设状态不包含复合状态，即采集的每一个动作之间互不包含，例如一个人不会先乘坐电梯上行，之后再乘坐电梯下行。

(2)

2.2 符号说明

表 1: **符号说明**

符号	含义	单位
a	加速度	g
acc_x	加速度计 X 轴记录值	g
acc_y	加速度计 Y 轴记录值	g
acc_z	加速度计 Z 轴记录值	g
$\|a\|$	合成加速度	g
θ	加速度与加速度在 YOZ 平面上投影的夹角	degree
ϕ	YOZ 平面上投影与 Z 轴夹角	degree
$\overline{acc_x}$	加速度 X 轴方向上的平均值	g
$\overline{acc_y}$	加速度 Y 轴方向上的平均值	g
$\overline{acc_z}$	加速度 Z 轴方向上的平均值	g
$\overline{a}$	合成加速度平均值	g
σ_x^2	X 方向上上的加速度方差	–
σ_y^2	Y 方向上上的加速度方差	–
σ_z^2	Z 方向上上的加速度方差	–
σ^2	合成加速度方差	–
a_{pv}	加速度峰谷值	g
$gyro\_x$	陀螺仪 X 轴记录值	$dps(degreepersecond)$
$gyro\_y$	陀螺仪 Y 轴记录值	$dps(degreepersecond)$
$gyro\_z$	陀螺仪 Z 轴记录值	$dps(degreepersecond)$

表 2: **[续] 符号说明**

符号	含义	单位
a_1	活动状态：向前走	–
a_2	活动状态：向左走	–
a_3	活动状态：向右走	–
a_4	活动状态：步行上楼	–
a_5	活动状态：步行下楼	–
a_6	活动状态：向前跑	–
a_7	活动状态：跳跃	–
a_8	活动状态：坐下	–
a_9	活动状态：站立	–
a_10	活动状态：躺下	–
a_11	活动状态：乘坐电梯向上移动	–
a_12	活动状态：乘坐电梯向下移动	–
age	年龄	岁
height	身高	cm
weight	体重	kg

3. 数据预处理

3.1 数据清洗

为了提高数据的准确性和可靠性，减少异常数据在建模与分析时造成的误差，需要先进行数据清洗，此处主要指处理异常值。

实验数据包含由加速度计测量的 X、Y、Z 轴的加速度和由陀螺仪测量的 X、Y、Z 轴的角速度。加速度的变化范围为 ±6g，角速度的变化范围为 ±500dps。所以若线加速度值出现 [-6,6] 以外的数据，或者角速度值出现 [-500,500] 以外的数据，则将该采样时刻的 3 个线加速度和 3 个角加速度删除。

3.2 异常数据处理

附件 4 中给出的身高和体重对应的数据对应错误，对调身高栏和体重栏数据。

3.3 利用滑动窗口法来制作时序数据集

传感器测量的加速度数据可能因为运动抖动或其他因素不稳定，通过滑动窗口可以减少抖动，使数据更稳定可靠。滑动窗口是一种简便有效的数据分割方法，其关键在于

确定窗口大小。如果窗口设置过大，可能会引起运动识别的延迟，并且一个窗口可能包含多种动作类型，导致训练误差。而如果窗口设置过小，可能无法包含一个完整的动作，导致运动信息缺失，影响识别结果。近年来，许多学者经过多次实验研究发现，将滑动窗口的重叠率设置为 50% 是一种较为有效的方法 [1]。在我们的实验中，综合分析了 12 种运动模式的特点，并统计了完成每种动作所需的时间，选用一个滑动窗口的宽度为 20 个采样点，每两个相邻窗口重叠 50%。以走路时 X 方向的加速度数据为例，选用 20 个采样点作为窗口宽度，如图 3 所示，这样规整了不同动作的加速度信号的长度，对接下来的特征提取和运动识别尤为重要。

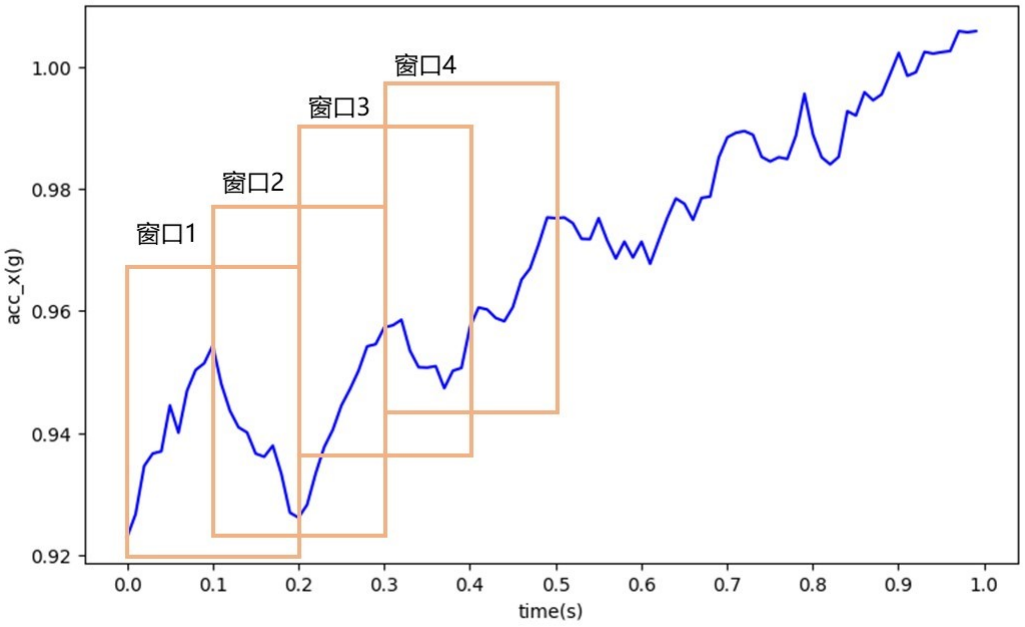

图 4: 滑动窗口处理示意图

3.4 滤波处理

在具体的数据处理中，由于速度计和陀螺仪的时间序列数据受到了抖动、电路、传输噪声等的影响，所以采集到的信号中包含许多无关信号，考虑使用滤波方式进行降噪处理 [14]。数据滤波是去除噪声还原真实数据的一种数据处理技术。目前较常使用的两种数据滤波方法分别是互补滤波和卡尔曼滤波。因此使用滤波方法可以去除加速度传感器和陀螺仪的噪声，取得精准的姿态数据。

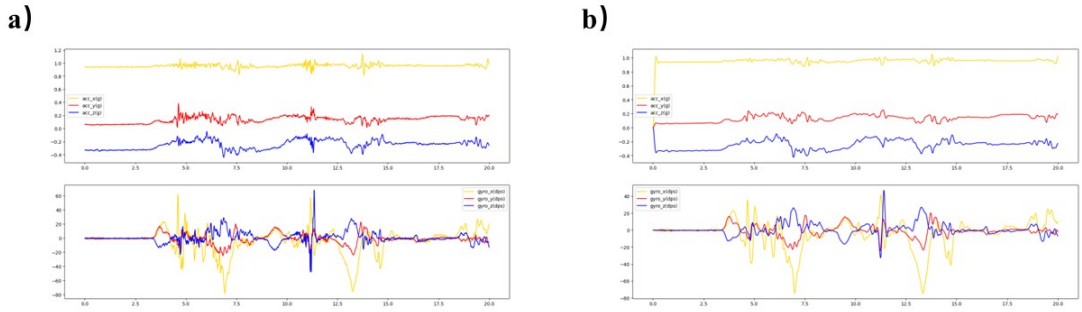

图 5: 滤波前后数据对比
图 a) 显示为滤波前数据，图 b) 显示为滤波后数据。

3.5 特征提取

3.5.1 去除重力影响

实验数据中 X 轴加速度值为融合了重力加速度的值，重力加速度会对后续分析人体运动方向造成干扰，所以对 X 轴加速度做减 1 处理，去除重力加速度影响 [15]。

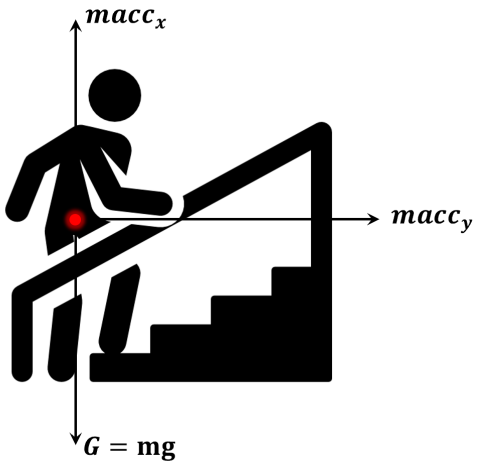

图 6: 步行上楼受力示意图

3.5.2 计算合成加速度

由于数据给出的是加速度在 X、Y、Z 轴三个方向的分量，不能直观体现人体运动的加速度大小，通过公式 $|a| = \sqrt{acc_x^2 + acc_y^2 + acc_z^2}$ 得到合成加速度大小。类似地，通过公式 $|gyro| = \sqrt{gyro_x^2 + acc_y^2 + acc_z^2}$ 得到合成角速度大小。

3.5.3 计算加速度方向

数据中加速度在 X、Y、Z 轴三个方向的分量难以反映加速度方向，通过公式 $\theta = arcsin\frac{acc_y}{\sqrt{acc_y^2+acc_z^2}}$ 计算加速度 a 与 a 在 YOZ 平面上投影的夹角 θ，通过公式 $\phi = arcsin\frac{acc_x}{|a|}$ 计算 YOZ 平面上投影与 Z 轴的夹角 ϕ。

对应地，通过公式 $\theta' = arcsin\frac{gyro_y}{\sqrt{gyro_y^2+gyro_z^2}}$ 计算角速度 $gyro$ 与 $gyro$ 在 YOZ 平面上投影的夹角 θ'，通过公式 $\phi' = arcsin\frac{gyro_x}{|gyro|}$ 计算 YOZ 平面上投影与 Z 轴的夹角 ϕ'。

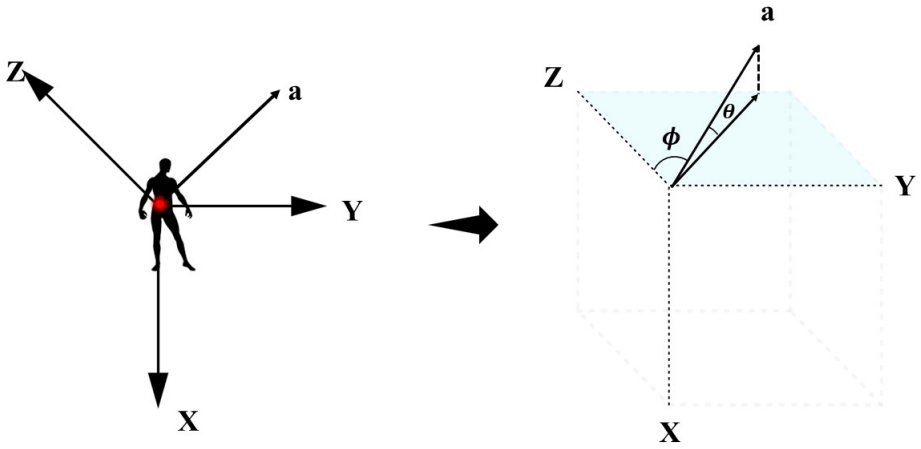

图 7: 计算加速度方向

3.5.4 提取时域特征

为了进行机器学习，对附件给出的时序数据进行统计时域特征提取，分别使用加速度平均值、合成加速度平均值、标准差、中位数、95% 分位点、5% 分位点、95% 分位点-5% 分位点。

加速度平均值 为了反映人体在运动时 3 个方向上的剧烈程度，使用能够反映信号在 X,Y,Z 轴各自平均状态的加速度均值来作为特征，计算公式如下：

$$\overline{acc_x} = \frac{\sum_{k=1}^{n} acc_{x,k}}{n} \tag{1}$$

$$\overline{acc_y} = \frac{\sum_{k=1}^{n} acc_{y,k}}{n} \tag{2}$$

$$\overline{acc_z} = \frac{\sum_{k=1}^{n} acc_{z,k}}{n} \tag{3}$$

式中，$acc_{x,k}$，$acc_{y,k}$，$acc_{z,k}$ 为一个动作时序数据中第 k 个采样点得到的 X，Y，Z 三轴加速度信号；n 为每个动作时序数据中采样点的个数。

合成加速度平均值 使用合成加速度平均值 $\overline{a}$ 综合前面各个加速度平均值，计算公式如下：

$$\overline{a_k} = \sqrt{a_{x,k}^2 + a_{y,k}^2 + a_{z,k}^2} \tag{4}$$

$$\overline{a} = \frac{\sum_{k=1}^{n} a_k}{n} \tag{5}$$

上式中，a_k 为动作时序数据中第 k 个采样点三轴合加速度的模，将 a_k 带入式5可得合成加速度均值 $\overline{a}$。

加速度标准差 以 X 方向为例，加速度方差 σ_x^2 计算公式如下：

$$\sigma_x^2 = \frac{\sum_{k=1}^{n} (a_{x,k} - \overline{a}_x)^2}{n} \tag{6}$$

同理可得 σ_y^2，σ_z^2:

$$\sigma_y^2 = \frac{\sum_{k=1}^{n} (a_{y,k} - \overline{a}_y)^2}{n} \tag{7}$$

$$\sigma_z^2 = \frac{\sum_{k=1}^{n}(a_{z,k} - \overline{a}_z)^2}{n} \tag{8}$$

合成加速度方差 合成加速度方差的计算方式为：

$$\sigma^2 = \frac{\sum_{k=1}^{n}(a_k - \overline{a})^2}{n} \tag{9}$$

加速度峰谷值 加速度峰谷值表示在一个动作时序数据中，加速度的最大值和最小值的差值，用 a_{pv} 表示：

$$a_{pv} = Max(a) - Min(a) \tag{10}$$

利用式10可以计算出合加速度的峰谷值以及 3 个方向加速度峰谷值。

中位数 为修正极大或极小值对平均值的影响，考虑中位数。

95% **分位点和** 5% **分位点** 为考虑较大值和较小值范围，取 95% 分位点和 5% 分位点分别表示较大值和较小值。

95% **分位点-**5% **分位点** 为观察曲线大部分数据点的波动宽度，使用 95% 分位点-5% 分位点作为特征。

4. 问题一分析与模型建立

4.1 问题一分析

已知有 3 组实验数据，每组实验数据包含 12 种动作状态各 5 个样本，每组数据共包含 60 个样本。样本中经过预处理后数据有 $|a|$、θ、ϕ、$|gyro|$、θ'、ϕ'。通过这些数据，设计聚类模型，将每组数据中的 60 个样本聚为 12 类，最佳情况下每类恰好有 5 个样本。

将样本按照时间画出信号曲线，可发现 12 种动作状态在某些特征上有明显区别，再根据实际含义可初步判断出某类动作状态。例如根据重力方向加速度的波动情况可以将 12 种动作状态划分为两大类：静止状态和运动状态。其中静止状态重力方向加速度曲线平缓，包含坐下、站立、躺下、乘坐电梯向上移动、乘坐电梯向下移动共 5 种动作状态；运动状态重力方向加速度曲线波动较大，包含向前走、向左走、向右走、步行上楼、不行下楼、向前跑、跳跃共 7 种动作状态。在静止状态中，“躺下”动作状态由于传感器受向上的支撑力，所以重力方向加速度会在 0g 左右，其他动作状态重力方向加

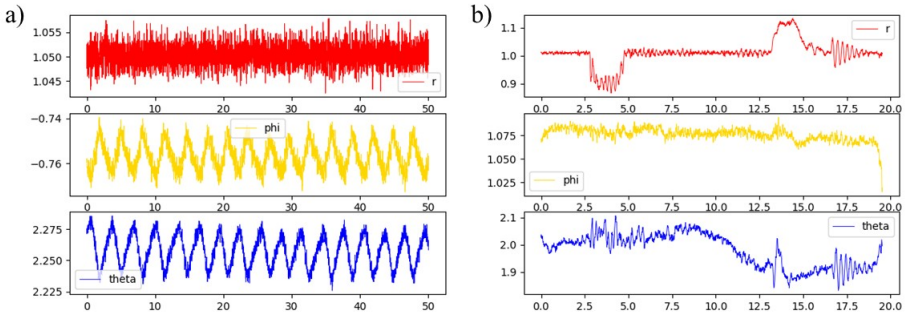

图 8: 示例：样本的 $|a|$、θ、ϕ 值

速度会在 1g 左右。在运动状态中，“跳跃”动作状态、“向前跑”动作重力加速度方向波动范围比较大。

表 3: **动作划分表**

活动状态	划分大类
a_1	动作状态
a_2	动作状态
a_3	动作状态
a_4	动作状态
a_5	动作状态
a_6	动作状态
a_7	动作状态
a_8	静止状态
a_9	静止状态
a_10	静止状态
a_11	静止状态
a_12	静止状态

4.2 基于 K-means 的聚类模型建立

问题包含 12 类动作状态，即固定聚为 12 类，这一条件很好的弥补了 K-means 需要指定类别的缺点。且 K-means 适用于将样本划分为互不重叠的簇的问题，所以选用 K-means 聚类。为了消除不同人身高、体重、年龄等对聚类效果的影响，所以对于每

组测试数据分别进行聚类，每组测试数据中的一个样本即为一个数据点。在本问题中，K-means 聚类的步骤为：

(1) 随机选择 12 个数据点作为聚类中心。

(2) 将每个数据点分配给距离最近的聚类中心。此处需要决定特征向量特征值的选取以及距离度量方法。

(3) 更新聚类中心，采用簇内各维特征值的平均值作为新的聚类中心。

(4) 对分配数据点和更新聚类中心进行迭代，直到聚类中心不再发生改变，或迭代次数达到设定最大值 1200。

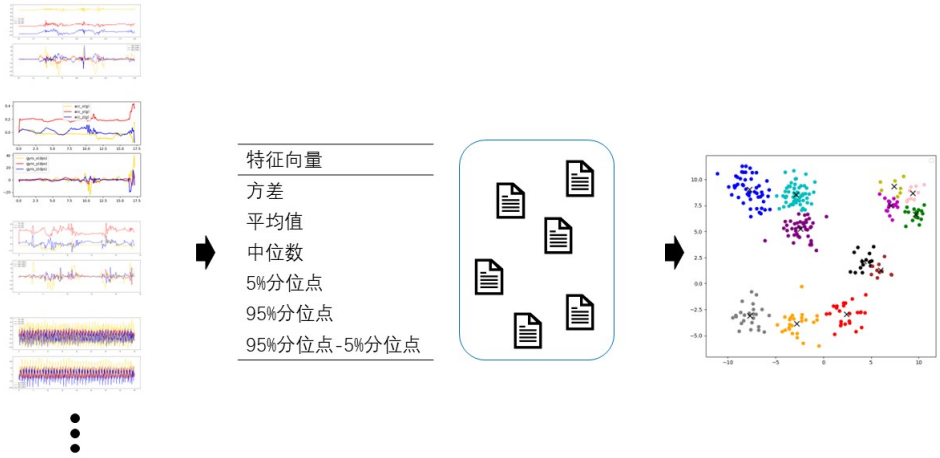

图 9: 聚类模型流程

4.2.1 特征向量特征值的选取

根据问题分析可得，信号曲线的波动情况、平稳曲线趋近的数值大小、信号整体剧烈程度、信号波动范围都可以作为动作状态的区分指标。所以选用的特征及其原因如下表所示。

表 4: **特征值表**

特征	原因
方差	反映曲线波动是否剧烈
平均值	反映曲线所代表速度的平均大小
中位数	消除过大过小值对平均值的影响
5% 分位点	曲线中较小数据点大小
95% 分位点	曲线中较大数据点大小
95% 分位点-5% 分位点	曲线大部分数据点的波动宽度

对 $|a|$、θ、ϕ、$|\omega|$、$|gyro|$、θ'、ϕ' 均计算上表中的 6 个特征值，得到样本的特征向量。

4.2.2 距离度量方法

因为欧氏距离适用范围广，便于理论分析，所以采用欧式距离作为数据点之间的距离。欧式距离计算公式：

$$\rho = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{11}$$

4.2.3 更新聚类中心

采用簇内各维特征值的平均值作为新的聚类中心，公式如下：

$$c_i = \frac{1}{|S_i|} \sum_{x \in S_i} x \tag{12}$$

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。

4.3 聚类模型应用

表 5: **问题一结果**

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

5. 问题二分析与模型建立

5.1 问题分析

本题与问题一的区别在于，本题数据带有类别标签，而问题一无标签。本题要求建立多分类模型对数据进行分类。测试数据共 10 组，可以采用交叉验证的方式划分数据集，训练得到准确率较好的模型后预测附件 3 中的 30 次活动状态。通过交叉验证可以评估模型在每一类上的准确度，确保模型的泛化能力。

在问题一中的聚类模型基础上，分析 10 组测试数据，可以得到每种动作状态的准确度，并按类别对动作状态分析两个模型的准确度。聚类模型提供的基础特征可以帮助我们更好地理解不同动作状态的特征分布，为分类模型的建立提供参考。

5.2 基于随机森林的分类模型建立

每个样本特征值高达 36 个，随机森林算法可以处理高维数据和过拟合问题，因此选用随机森林算法。随机森林通过构建多个决策树并综合它们的输出结果，能够有效提高分类的准确性和稳定性。在本问题中，随机森林中使用的特征为每个样本对三个轴方向的加速度、角速度提取的问题一表中的 6 个特征。使用信息增益作为划分子树的评估标准，可以有效地选择最具区分能力的特征。

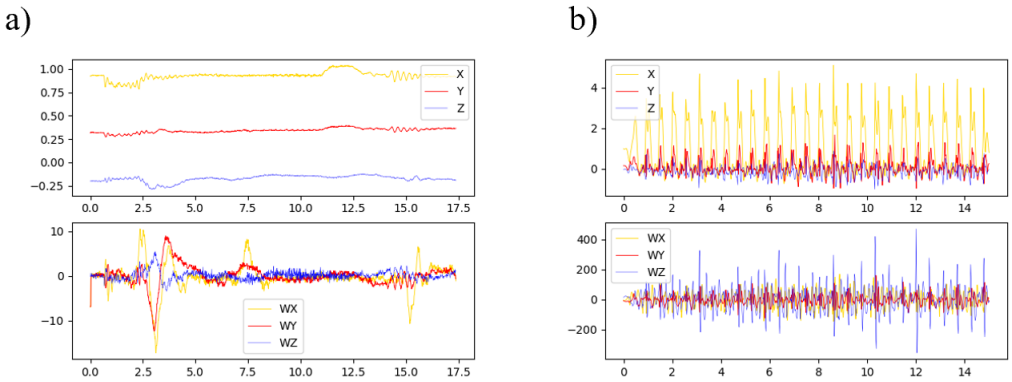

图 10: 样本原始数据曲线

建立随机森林分类模型的详细步骤如下：

(1) **数据预处理**：读取原始数据，对加速度和角速度数据进行标准化处理，确保不同特征之间具有相似的量级，避免特征值范围差异过大对模型训练的影响。

(2) **特征提取**：对每个样本提取三个轴方向上的加速度和角速度的 6 个特征值，包括方差、平均值、中位数、5% 分位点、95% 分位点以及 95% 分位点-5% 分位点。这些特征能够反映信号的波动情况、整体剧烈程度和波动范围。

(3) **数据集划分**：将数据集按照训练集：测试集 =8:2 的方式划分，确保训练集和测试集具有代表性和均衡性。

(4) **模型训练**：使用训练集训练随机森林分类器，设置适当的参数（如树的数量为 100，最大深度为 None），通过网格搜索和交叉验证优化模型参数，提高模型的性能。

(5) **模型评估**：在测试集上评估分类器的性能，计算准确率、召回率、F1-score 等指标，综合衡量模型的分类效果。绘制混淆矩阵以直观展示分类效果，分析模型在不同类别上的分类准确度和误差情况。

(6) **模型预测**：使用训练好的模型对附件 3 中的 30 次活动状态进行预测，验证模型的泛化能力和实际应用效果。

5.3 分类模型应用

通过上述方法，将测试集按照 8：2 的方式划分数据，进行数据测试，最终得到的训练准确率为 85%。

混淆矩阵如下：

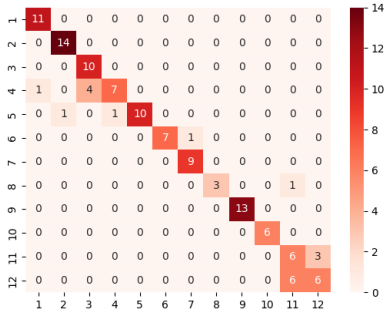

图 11: 混淆矩阵

分类报告如下所示：

表 6: **分类报告**

	precision	recall	f1-score	support
1	0.92	1.00	0.96	11
2	0.93	1.00	0.97	14
3	0.71	1.00	0.83	10
4	0.88	0.58	0.70	12
5	1.00	0.83	0.91	12
6	1.00	0.88	0.93	8
7	0.90	1.00	0.95	9
8	1.00	0.75	0.86	4
9	1.00	1.00	1.00	13
10	1.00	1.00	1.00	6
11	0.46	0.67	0.55	9
12	0.67	0.50	0.57	12
accuracy			0.85	120
macro avg	0.87	0.85	0.85	120
weighted avg	0.87	0.85	0.85	120
Accuracy: 0.85				

通过详细的特征提取、数据预处理、模型训练和评估步骤，最终得到训练准确率为 85%，并使用该模型对附件 3 中的 30 次活动状态进行预测，取得了良好的分类效果 (下表)。

表 7: **问题二结果**

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

5.4 聚类模型与分类模型的比较

在机器学习领域，聚类与分类是两种常用的数据分析方法。由于第一问和第二问分别提出了聚类和分类模型，接下来比较这两种模型。

聚类算法是一种无监督学习方法，它旨在将数据集中的对象划分为若干个群组，使得同一群组内的对象相似度较高，而不同群组间的对象相似度较低。在本研究中，我们采用了 K-means 算法对实验数据进行聚类分析。K-means 算法在处理特征差异较大的状态时，如躺、跳跃等动作，能够实现准确聚类。这是因为这些状态的特征向量与其他状态的特征向量之间存在明显的差异，使得 K-means 算法能够有效地将它们划分到不同的群组中。

然而 K-means 算法在处理特征相似的状态时，如向左走、步行下楼、乘坐电梯上下移动等，其聚类效果并不理想。这些状态的特征向量具有较高的相似性，导致 K-means 算法在划分群组时容易出现不稳定的现象。在这种情况下，聚类结果可能会受到初始中心点选择、数据噪声等因素的影响，从而导致聚类效果不佳。

分类算法是一种有监督学习方法，它旨在根据已知的类别标签，将数据集中的对象划分为相应的类别。在本研究中，我们采用了随机森林算法对实验数据进行分类。随机森林算法具有较高的分类准确率和稳定性，对于大部分状态都能实现准确分类。实验结果表明，随机森林算法在本研究中的分类准确率可达 85% 以上。

但随机森林算法在处理特征差异较大的状态时，其分类效果并不完美。这是因为这些状态的特征向量在空间中分布较为分散，使得随机森林算法难以找到一个合适的分类边界。此外，随机森林算法在训练过程中可能会受到样本不均衡、特征相关性等因素的影响，从而导致分类误差。

综上所述，聚类与分类在实际应用中各有优劣。聚类算法在处理特征差异较大的数据时具有较好的效果，但容易受到特征相似性的影响；而分类算法在处理大部分状态时具有较高的准确率和稳定性，但在处理特征差异较大的状态时存在一定的局限性。因此，在实际应用中，我们需要根据具体问题和数据特点选择合适的方法。

6. 问题三分析与模型建立

6.1 基于 PCA 的相关性分析

要分析不同人员的同一活动状态是否存在差异，考虑使用皮尔逊相关系数或斯皮尔曼相关系数进行衡量。在本文中，一个维度的变量是年龄、身高或体重，另一个维度的变量是活动状态的某个特征。由于原始数据是六维的，这里我们选用主成分分析 (PCA) 进行降维，用一维特征来表示活动状态的特征。我们分别计算年龄、身高、体重和这个一维特征的皮尔逊相关系数，得到不同人员的同一活动状态存在差异，活动状态数据与

实验人员的年龄、身高、体重有关系。

6.1.1 数据预处理

(1) 从附件 4 中提取 13 位实验人员的年龄、身高、体重和活动状态数据。

(2) 检查并清理数据，确保数据完整无误。

```
% Load data
Load data from 'Attachment4.xlsx'

% Extract age, height, weight, and activity data
Extract columns: age, height, weight, and activity_data

% Standardize activity data
Standardize activity_data
```

6.1.2 特征提取与降维

(1) 对每种活动状态的六维数据（加速度计和陀螺仪的三个轴向数据）进行标准化处理。

(2) 使用主成分分析（PCA）对标准化后的六维数据进行降维，提取第一主成分作为该活动状态的一维特征。

```
% Apply PCA for dimensionality reduction
Apply PCA on standardized activity_data to get activity_feature
```

6.1.3 相关性分析

(1) 分别计算年龄、身高、体重与活动状态一维特征之间的皮尔逊相关系数。

(2) 记录每种活动状态的相关系数值，判断不同人员的同一活动状态是否存在显著差异。

```
% Calculate Pearson correlation coefficients
Calculate Pearson correlation between age and activity_feature
Calculate Pearson correlation between height and activity_feature
Calculate Pearson correlation between weight and activity_feature

% Output correlation results
Print correlation_age
Print correlation_height
Print correlation_weight
```

6.1.4 结果分析与讨论

(1) 在向前走的状态中，我们发现年龄与主成分之间的相关系数为 0.68x，这表明在这一特定运动状态下，实验参与者的年龄与运动特征之间存在中等程度的正相关性。这意味着随着年龄的增长，某些运动特征可能呈现出一定的变化趋势，这可能与肌肉力量、灵活性或其他生理因素的退化有关。

(2) 此外，我们还考虑了体重和身高这两个重要的生理指标。体重可能影响运动时的稳定性和能量消耗，而身高可能与步幅和运动范围有关。发现体重与主成分之间的相关系数较低，这表明体重对特定运动状态的影响不如年龄显著。而身高与主成分的相关系数表现出不同的模式，这揭示了身高对于某些运动特征的特定影响，如步长或运动幅度。

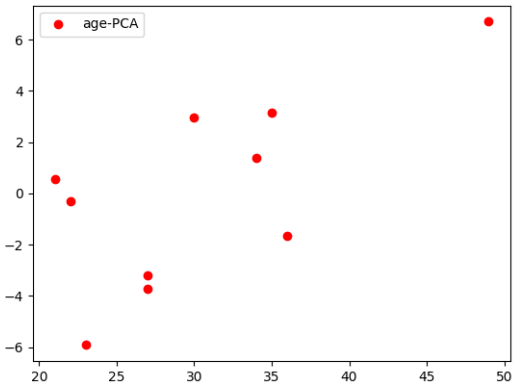

图 12: 向前走动作中年龄与主成分散点图

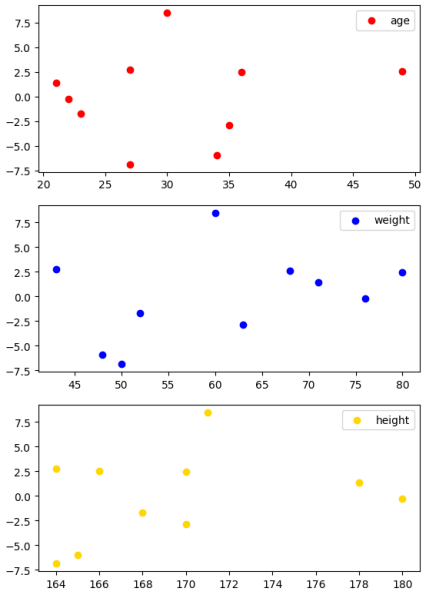

图 13: 站立动作中年龄、身高、体重与主成分散点图

6.2 使用活动传感器数据进行人员画像

6.2.1 数据预处理

(1) 从附件 5 中提取 10 位实验人员中的 5 位某次活动的状态数据，包括每人的 12 类活动状态。

(2) 对数据进行标准化处理，确保不同量纲的数据具有可比性。

6.2.2 特征提取

(1) 使用 PCA 对每种活动状态的数据进行降维，提取一维特征。

(2) 将每位实验人员的 12 类活动状态的一维特征组合成特征向量，作为该人员的活动特征画像。

6.2.3 判别模型的建立与验证

(1) 将问题 2 中建立的判别模型应用于附件 5 的数据，根据特征向量判断这 5 位实验人员分别最可能来源于问题 2 中的哪一名实验人员。

(2) 使用 KNN 分类模型对实验人员进行分类，比较不同模型的分类准确度，选择最佳模型。

```
% Load new data
Load data from 'Attachment5.xlsx'

% Apply PCA for dimensionality reduction and combine features
Apply PCA on new_data to get new_data_features
Combine new_data_features into profiles

% Train and apply KNN model
Train KNN model with activity_profiles and person_id
Predict new_data_profiles using trained KNN model

% Save results to Excel file
Save predictions to 'Table3.xlsx'
```

6.2.4 判别结果

根据模型分类结果，将每位实验人员的判别结果填入下表。

表 8: **问题三结果**

活动类型	判别结果
Unknow1	5
Unknow2	11
Unknow3	12
Unknow4	9
Unknow5	4

7. 模型评价与改进

7.1 模型的优点

7.1.1 问题一的聚类模型

(1) **特征选择合理**：通过方差、平均值、中位数等特征值，充分考虑了信号曲线的波动情况、剧烈程度等特性，能够较好地反映不同动作状态之间的差异。

(2) **K-means 聚类效果好**：聚类模型能够有效地将 12 种动作状态分类，每组测试数据中的样本分配较为均匀，保证了分类的准确性。

(3) **算法简单高效**：K-means 聚类算法计算复杂度低，适用于大规模数据的处理，能够快速收敛，满足实验需求。

7.1.2 问题二的分类模型

(1) **处理高维数据能力强**：随机森林算法能够处理高维数据，避免了过拟合问题，适用于多分类问题。

(2) **模型鲁棒性强**：随机森林通过集成多个决策树，具有较强的鲁棒性和泛化能力，能够在不同数据集上取得较好的分类效果。

(3) **特征提取全面**：根据加速度和角速度的 6 个特征值进行特征提取，能够全面反映样本的特征，提高模型的分类准确性。

7.1.3 问题三的判别模型

(1) **综合性强**：判别模型通过分析年龄、身高、体重等个人信息以及活动状态的特征，能够全面评估不同人员的活动差异。

(2) **方法灵活**：通过皮尔逊相关系数和主成分分析等方法，能够灵活处理不同维度的数据，提高模型的适应性。

(3) **结果直观**：模型能够直观展示不同人员的活动差异，便于分析和应用。

7.2 模型的不足

7.2.1 问题一的聚类模型

(1) **对初始值敏感**：K-means 聚类算法对初始聚类中心较为敏感，容易陷入局部最优解，影响聚类效果。

(2) **无法处理噪声和异常值**：K-means 算法对噪声和异常值较为敏感，可能会影响聚类结果的准确性。

(3) **类别数固定**：K-means 需要预先指定类别数，无法自动确定最佳类别数，可能会限制其应用范围。

7.2.2 问题二的分类模型

(1) **训练时间较长**：随机森林算法需要构建大量决策树，训练时间较长，对于大规模数据集可能存在性能瓶颈。

(2) **模型复杂性高**：随机森林模型较为复杂，不易解释，难以直观理解每个特征对分类结果的贡献。

(3) **需要大量特征**：模型依赖于大量特征提取，特征选择不当可能会影响分类效果。

7.2.3 问题三的判别模型

(1) **特征降维可能丢失信息**：主成分分析进行特征降维可能会丢失部分信息，影响模型的判别准确性。

(2) **相关性分析局限性**：皮尔逊相关系数只能反映线性相关关系，可能无法全面捕捉复杂的非线性关系。

(3) **个体差异较大**：不同个体的活动状态差异较大，单一模型可能难以全面适应所有个体的特征。

7.3 模型的改进

7.3.1 问题一的聚类模型

(1) **使用改进的聚类算法**：采用如 K-means++、DBSCAN 等改进的聚类算法，减少对初始值的敏感性，提高聚类效果。

(2) **引入噪声处理机制**：通过预处理步骤对噪声和异常值进行过滤，减少其对聚类结果的影响。

(3) **自动确定类别数**：采用如轮廓系数等方法，自动确定最佳类别数，提升模型的适应性。

7.3.2 问题二的分类模型

(1) **优化模型参数**：通过网格搜索和交叉验证优化随机森林模型的参数，提高模型的分类性能和训练效率。

(2) **集成多种算法**：结合如支持向量机、神经网络等其他分类算法，构建集成模型，提高分类准确性。

(3) **特征选择优化**：通过特征选择算法筛选出最具代表性的特征，减少模型的复杂性，提高分类效果。

7.3.3 问题三的判别模型

(1) **结合非线性降维方法**：引入如 t-SNE、UMAP 等非线性降维方法，捕捉复杂的非线性关系，提高模型的判别准确性。

(2) **增加数据多样性**：通过收集更多样本数据，增强模型的泛化能力，适应不同个体的特征差异。

(3) **改进相关性分析方法**：采用如距离相关系数等更全面的相关性分析方法，捕捉更复杂的关系，提高模型的分析能力。

参考文献

[1] 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 .” 计算机系统应用 22.6 (2013): 132-135.

[2] 孔俊其. 基于三维加速度传感器的手势识别及交互模型研究 [D]. 江苏: 苏州大学, 2009.

[3] Ling B, Intille S S. Activity Recognition from User-Annotated Acceleration Data[J]. Proc Pervasive, 2004, 3001:1-17.

[4] Ermes M, Parkka J, Mantyjarvi J, et al. Detection of Daily Activities and Sports With Wearable Sensors in Controlled and Uncontrolled Conditions[J]. IEEE Transactions on Information Technology in Biomedicine, 2008, 12(1):20-26.

[5] Wu J, Pan G, Zhang D, et al. Gesture Recognition with a 3-D Accelerometer[C]. International Conference on Ubiquitous Intelligence and Computing. Springer-Verlag, 2009:25-38.

[6] Cho S J, Choi E, Bang W C, et al. Two-stage Recognition of Raw Acceleration Signals for 3-D Gesture-Understanding Cell Phones[J]. Tenth International Workshop on Frontiers in Handwriting Recognition, 2006.

[7] 何超, 胡章芳, 王艳. 一种基于改进 DTW 算法的动态手势识别方法 [J]. 数字通信, 2013, 40(3):21-25.

[8] 徐川龙. 基于三维加速度传感器的人体行为识别 [D]. 杭州: 浙江工业大学, 2013.

[9] 常亚南. 基于 HMM 的动态手势识别 [D]. 广州: 华南理工大学, 2012.

[10] BAO Guang-bin, ZHANG Le, ZHAO Hon. Research of Intelligent Car Dual Navigation System Based on Complex Environment[C]. The 8th International Conference on Green Intelligent Transportation System and Safety, GITSS2017, 2017.

[11] Agrawal S, Constandache I, Gaonkar S, et al. PhonePoint pen: using mobile phones to write in air[C]. ACM SIGCOMM Workshop on Networking, Systems, and Applications for Mobile Handhelds, Mobiheld 2009, Barcelona, Spain, August. DBLP, 2009:1-6.

[12] 荆雷, 马文君, 常丹华. 基于动态时间规整的手势加速度信号识别 [J]. 传感技术学报, 2012, 25(1):72-76.

[13] 陈鹏展, 罗漫, 李杰. 基于加速度传感器的连续动态手势识别 [J]. 传感器与微系统, 2016, 35(1):39-42.

[14] 刘玉焘. 基于可穿戴式传感器的人体动作捕获与识别研究. Diss. 哈尔滨工业大学, 2020.

[15] 孙玉杰. 基于可穿戴传感器的人体运动捕捉与识别技术研究 [D]. 山东大学,2023.

附录

1. 问题结果总表

1.1 问题一结果

表 9: **问题一结果**

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

1.2 问题二结果

表 10: **问题二结果**

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

1.3 问题三结果

表 11: **问题三结果**

活动类型	判别结果
Unknow1	5
Unknow2	11
Unknow3	12
Unknow4	9
Unknow5	4

2. 核心代码

2.1 角度计算与可视化

```
pathHead="D:/mathModel/A/fj2/Person5CleanX1/"
pathTail=".xlsx"

for i in range(1,13):
    for j in range(1,6):
        path=pathHead+"a"+str(i)+"t"+str(j)+pathTail
        print(path)
        df=pd.read_excel(path)
        r= np.sqrt(df['acc_x(g)']**2 + df['acc_y(g)']**2 + df['acc_z(g)']**2)
        phi=np.arcsin(df['acc_x(g)']/r)
        theta=np.arccos(df['acc_z(g)']/np.sqrt(df['acc_y(g)']**2 + df['acc_z(g)']**2))

        rW=np.sqrt(df['gyro_x(dps)']**2 + df['gyro_y(dps)']**2 + df['gyro_z(dps)']**2)
        phiW=np.arcsin(df['gyro_x(dps)']/rW)
        thetaW=np.arccos(df['gyro_z(dps)']/np.sqrt(df['gyro_y(dps)']**2 + df['gyro_z(dps)']**2))
        data=df
        data['r']=r
        data['phi']=phi
        data['theta']=theta
        data['rW']=rW
        data['phiW']=phiW
        data['thetaW']=thetaW
        # data=pd.concat([data,r],axis=1)
        # data=pd.concat([data,phi],axis=1)
        # data=pd.concat([data,theta],axis=1)
        # data.columns=['r','phi','theta']
        data.to_excel('D:/mathModel/newData/fj2/P5X1DataRTPW/a{}t{}.xlsx'.format(i,j), index=False)
        # x=np.linspace(0,len(df)/100,len(df))
        # plt.figure()
        # plt.subplot(311)
        # plt.plot(x,r,label='Line 1',color='red',linewidth=0.6)
        # plt.legend(['r'])

        # plt.subplot(312)
        # plt.plot(x,phi,label='Line 1',color='gold',linewidth=0.6)
        # plt.legend(['phi'])

        # plt.subplot(313)
        # plt.plot(x,theta,label='Line 1',color='blue',linewidth=0.6)
        # plt.legend(['theta'])

        # plt.savefig("D:/mathModel/picture/fj2/Person4/rtp/a{}t{}.png".format(i,j))
    #plt.show()
```

2.2 特征提取

```
def getVal(perRou,colum):
    cols=len(perRou)
    data=np.array(perRou[colum])
    mean = np.mean(data)  # 均值
    std = np.std(data)  # 标准差
    median = np.median(data)
    p5 = np.percentile(data, 5)
    p95 = np.percentile(data, 95)
    return [std,mean,median,p5,p95,p95-p5]

# path1="D:/mathModel/newData/fj2/P"
# path2="X1DataRTPW/"
# pathTail=".xlsx"

pathHead="D:/mathModel/A/fj3/SY"
pathTail=".xlsx"

acXind=['acXstd','acXmean','acXmedian','acXp5','acXp95','acXp95s5']
```

```
acYind=['acYstd','acYmean','acYmedian','acYp5','acYp95','acYp95s5']
acZind=['acZstd','acZmean','acZmedian','acZp5','acZp95','acZp95s5']
acWXind=['acWXstd','acWXmean','acWXmedian','acWXp5','acWXp95','acWXp95s5']
acWYind=['acWYstd','acWYmean','acWYmedian','acWYp5','acWYp95','acWYp95s5']
acWZind=['acWZstd','acWZmean','acWZmedian','acWZp5','acWZp95','acWZp95s5']

colum=['type','acXstd','acXmean','acXmedian','acXp5','acXp95','acXp95s5','acYstd','acYmean','acYmedian','acYp5','acYp95
       ','acYp95s5','acZstd','acZmean','acZmedian','acZp5','acZp95','acZp95s5','acWXstd','acWXmean','acWXmedian','acWXp5
       ','acWXp95','acWXp95s5','acWYstd','acWYmean','acWYmedian','acWYp5','acWYp95','acWYp95s5','acWZstd','acWZmean','
       acWZmedian','acWZp5','acWZp95','acWZp95s5']
out=DataFrame(columns=colum)

for k in range(4,5):
    # pathF=path1+str(k)+path2
    for ii in range(1,2):
        for j in range(1,31):
            # path=pathF+"a"+str(ii)+"t"+str(j)+pathTail
            path=pathHead+str(j)+pathTail
            print(path)
            df=pd.read_excel(path)
            dic={}
            dic['type']=ii
            df['acc_x(g)']=df['acc_x(g)']-1
            valI=getVal(df,'acc_x(g)')
            for i in range(6):
                dic[acXind[i]]=valI[i]
            valI=getVal(df,'acc_y(g)')
            for i in range(6):
                dic[acYind[i]]=valI[i]
            valI=getVal(df,'acc_z(g)')
            for i in range(6):
                dic[acZind[i]]=valI[i]
            valI=getVal(df,'gyro_x(dps)')
            for i in range(6):
                dic[acWXind[i]]=valI[i]
            valI=getVal(df,'gyro_y(dps)')
            for i in range(6):
                dic[acWYind[i]]=valI[i]
            valI=getVal(df,'gyro_z(dps)')
            for i in range(6):
                dic[acWZind[i]]=valI[i]
            out = pd.concat([out, pd.DataFrame(dic, index=[0])])
            #print(out)
out.to_excel('D:/mathModel/fj3.xlsx', index=False)
```

2.3 K-means 聚类

```
def getVal(perRou,colum,ty=0):
    #print(path)
    cols=len(perRou)
    data=np.array(perRou[colum])
    # max_value = np.amax(data)  # 最大值
    # peak_value = np.amax(abs(data))  # 最大绝对值
    # min_value = np.amin(data)  # 最小值data
    mean = np.mean(data)  # 均值
    # p_p_value = max_value - min_value  # 峰峰值
    # abs_mean = np.mean(abs(data))  # 绝对平均值
    # rms = np.sqrt(np.sum(data**2) / cols)  # 均方根值
    # square_root_amplitude = (np.sum(np.sqrt(abs(data))) / cols) ** 2  # 方根幅值
    std = np.std(data)  # 标准差
    # median = np.median(data)
    # p5 = np.percentile(data, 5)
    # p25 = np.percentile(data, 25)
    # p75 = np.percentile(data, 75)
    # p95 = np.percentile(data, 95)
    if ty==1:
        return [mean]
    return [mean,std]
    # absStd = np.std(abs(data))  #绝对值标准差
    # kurtosis = stats.kurtosis(data)  # 峭度
    # skewness = stats.skew(data)  # 偏度
    # clearance_factor = peak_value / square_root_amplitude  # 裕度指标
```

```
    # shape_factor = rms / abs_mean  # 波形指标
    # impulse_factor = peak_value  / abs_mean  # 脉冲指标
    # crest_factor = peak_value / rms  # 峰值指标
    # features = [max_value, peak_value, min_value, mean, p_p_value, abs_mean, rms, square_root_amplitude,
    #             std, kurtosis, skewness,clearance_factor, shape_factor, impulse_factor, crest_factor]
    return features

pathHead="newData/fj2/P4FilterDataRTPW/"
pathTail=".xlsx"

val=[]

for i in range(11,13):
    for j in range(1,6):
        path=pathHead+"a"+str(i)+"t"+str(j)+pathTail
        print(path)
        data=pd.read_excel(path)
        data['index']=data.index
        mea=np.median(data['acc_x(g)'])
        data['valx']=data.apply(lambda x: (x['index']*100)*((x['acc_x(g)']-mea)*100)**3,axis=1)
        valI=[]
        # valI+=getVal(data,"r")
        # valI+=getVal(data,"theta")
        # valI+=getVal(data,"phi")
        valI+=getVal(data,'valx',1)
        # valI+=getVal(data,"gyro_x(dps)")
        # valI+=getVal(data,"gyro_y(dps)")
        # valI+=getVal(data,"gyro_z(dps)")
        val.append(valI)

val=np.array(val)

s=val[:,-1]
print(s)

#print(val)
K=2
val=(val-val.mean(axis=0))/(val.std(axis=0))
#print(val)

kmeans = KMeans(n_clusters=K,n_init='auto',max_iter=1200).fit(val)

out=kmeans.labels_

outt=[]
for i in range(K):
    outt.append([])

for i in range(K):
    for j in range(len(out)):
        if out[j]==i:
            outt[i].append(j+1)
for i in range(K):
    print(outt[i])
```

2.4 随机森林

```
dataset = pd.read_excel('fj2.xlsx')
dataset2 = pd.read_excel('fj3.xlsx')
#print(dataset)

# [1,2,7,8,13,14,19,20,25,26,31,32]
X = dataset.iloc[:, 1:].values
y = dataset.iloc[:, 0].values


# X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)
X_train=X
y_train=y
X_test=dataset2.iloc[:, 1:].values

scaler = StandardScaler()
```

```
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)


classifier = RandomForestClassifier(n_estimators=500, criterion='entropy', random_state=42)
classifier.fit(X_train, y_train)
#print("X_train:", X_train)
#print("y_train:", y_train)



y_pred = classifier.predict(X_test)
#print("X_test:", X_test)
print("y_pred:", y_pred)

# result = confusion_matrix(y_test, y_pred)
# print("Confusion Matrix:")
# print(result)

# xtick=['1','2','3','4','5','6','7','8','9','10','11','12']
# ytick=['1','2','3','4','5','6','7','8','9','10','11','12']

# sns.heatmap(result,fmt='g',cmap='Reds',annot=True,xticklabels=xtick, yticklabels=ytick)
# plt.show()

# result1 = classification_report(y_test, y_pred)
# print("Classification Report:", )
# print(result1)
# result2 = accuracy_score(y_test, y_pred)
# print("Accuracy:", result2)
```

2.5 相关性分析

```
df=pd.read_excel(path)

df=df[df['type']==9]

df=df[df.columns[1:]]

data=np.array(df)

#print(data)

row=data.shape[0]
column=data.shape[1]

n = 5
reshaped_x = data.reshape(-1, n, data.shape[-1])
#print(reshaped_x)
avg_x = reshaped_x.mean(axis=1)
#print(avg_x[:,0])

scaler = StandardScaler()
avg_x = scaler.fit_transform(avg_x)

#print(avg_x[:,0])
pca = PCA(n_components=1)

pca.fit(avg_x)

pcaVal=pca.transform(avg_x)

pcaVal = list(chain.from_iterable(pcaVal))

print(pcaVal)

# for i in range(len(pcaVal)):
#     pcaVal[i]=1.0/pcaVal[i]

path2="D:/mathModel/A/fj4.xlsx"

df=pd.read_excel(path2)
```

```
40
41 age = df['age'].values
42 age=age[3:]
43
44 wei = df['wei'].values
45 wei=wei[3:]
46
47 hei = df['hei'].values
48 hei=hei[3:]
49
50 #print(age)
51 plt.figure()
52 plt.subplot(311)
53
54 li = zip(pcaVal,age)
55 lii = sorted(li,key=lambda x:x[1])
56 result = zip(*lii)
57 pcaVal2, age2 = [list(x) for x in result]
58 # print(pcaVal2)
59 # print(age2)
60 corr_xy = np.corrcoef(age2, pcaVal2)[0, 1]
61 print("Correlation between x and y:", corr_xy)
62 plt.scatter(age2, pcaVal2, c='red')
63 plt.legend(['age'])
64 #plt.show()
65
66 plt.subplot(312)
67
68 li = zip(pcaVal,wei)
69 lii = sorted(li,key=lambda x:x[1])
70 result = zip(*lii)
71 pcaVal2, wei2 = [list(x) for x in result]
72 # print(pcaVal2)
73 # print(age2)
74 corr_xy = np.corrcoef(wei2, pcaVal2)[0, 1]
75 print("Correlation between x and y:", corr_xy)
76 plt.scatter(wei2, pcaVal2, c='blue')
77 plt.legend(['weight'])
78 #plt.show()
79
80 plt.subplot(313)
81
82 li = zip(pcaVal,hei)
83 lii = sorted(li,key=lambda x:x[1])
84 result = zip(*lii)
85 pcaVal2, hei2 = [list(x) for x in result]
86 # print(pcaVal2)
87 # print(age2)
88 corr_xy = np.corrcoef(hei2, pcaVal2)[0, 1]
89 print("Correlation between x and y:", corr_xy)
90 plt.scatter(hei2, pcaVal2, c='gold')
91 plt.legend(['height'])
92 plt.show()
```