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：国防科技大学电子科学学院

参赛队员（打印后签名）：1. 刘昕昊 刘昕昊

2. 刘洋 刘洋

3. 柏恒 柏恒

指导教师或指导教师组负责人（打印后签名）：刘森

日期：2024 年 7 月 6 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。）

第九届湖南省研究生数学建模竞赛

题　目：基于惯性传感器的人体活动状态识别

摘要：

基于惯性传感器的人体运动模式识别是一种新兴起的人机交互方式，已经在健康监控、体育竞技、军事等诸多领域有着广泛的应用。本文针对现有基于惯性传感器的人体运动姿态识别中存在的特征数量较多、识别精度不足等问题，开展了多种人体运动姿态识别算法研究，通过数据预处理、特征提取与处理，建立模型，完成了动作分类与动作判别等运动姿态识别任务，进一步实现了基于活动状态数据的身体数据人员画像。实验结果表明，本文动作分类模型的准确度达到了 88.9%，动作标签判别模型的平均准确度达到了 91.43%，对身体数据的预测最高达到了 0.97 的均方根误差。

针对问题一：已知 3 名实验人员未带活动状态标签的各类活动状态数据，本文首先对其进行了均值滤波、卡尔曼滤波等预处理操作，降低信号均值偏移、空值与噪声的影响。然后对周期性数据量进行了滑窗切片处理，增加了样本数量；对切片后的活动状态数据进行了特征提取，得到了其时频域的多维信息，通过对比无监督特征处理算法性能，本文选择了主成分分析算法将特征维数从 152 降低至 30；对比分析了 K-均值算法与高斯混合模型对 3 名实验人员活动状态数据的聚类效果，鉴于高斯混合模型在 KL 散度上的优势，最终选择了该模型对 3 名实验人员活动状态进行分类。为了降低数据相关性的影响，本文使用马氏距离衡量特征向量间的距离。对每个实验人员单独分类后，通过横向对比相关性，对齐了各个实验人员的动作类别。

针对问题二：已知活动状态的 10 名实验人员活动数据经过预处理后，通过有监督特征处理算法 Relief-F 处理特征，按照 9：1 的比例划分训练集与验证集，对 SVM 算法进行有监督学习，使用 F1-分数与平均准确度分析了模型性能，并分析了对不同活动状态的模型判别性能；使用问题一提出的分类模型对这些带标签数据进行分类，分类模型的分类准确度达到了 88.9%，分析了不同活动类型分类精度出现差异的原因，同时与判别模型的结果进行了比较分析；最后使用判别模型对某实验人员 30 次活动状态数据进行标签判别。

针对问题三：对不同人员同一活动状态的归一化特征进行了比较，结果表明实验人员的不同会使活动数据特征出现明显差异；使用回归算法分析实验人员身体数据与活动状态数据的联系，分别使用多元线性回归算法与支持向量回归算法对 10 名带活动状态标签的人员数据进行交叉训练，以均方根误差和决定系数衡量回归效果，对比分析了两种算法的性能；最后使用适合于非线性与复杂数据建模的支持向量回归模型，基于 5 名未知实验人员的活动数据对其身体数据进行预测，并推测出了这些活动数据可能属于的实验人员。

最后，我们对提出的模型进行全面的评价：本文的模型贴合实际，可以分别解决已知活动类型与未知活动类型的动作识别问题，能够使用活动状态数据对人员身体数据进行预测分析，进行具有实用性强，算法效率高等特点，该模型在交通工具姿态识别与智能驾驶方面也一定推广价值。

关键词：主成分分析法 Relief-F 算法 高斯混合模型 支持向量机 支持向量回归

目录

1 问题综述

1.1 问题背景

在科技高速发展的今天，人类的社会生活与智能设备的联系日益密切。人体姿态识别是智能化生活领域中的一个重要组成部分[1]，拥有着巨大的应用价值，广泛应用于健康医疗、体育竞技、虚拟现实、智能监控等领域。人体姿态识别通过使用算法对人体姿态进行检测、跟踪和识别，准确捕捉和分析人体的姿势和动作，实现人与人或人与周围环境的交互。

随着惯性传感器件的发展，基于惯性传感器的定位技术研究受到了越来越多学者的关注。目前，大多数智能手机都具备如加速度传感器、角速度传感器等惯性传感装置，使用惯性传感器进行人体动作姿态识别对于提升用户体验、丰富应用场景有着重要作用，能够极大地扩展手机应用的功能和体验，使其能够更加智能化、个性化，从而提高用户的使用体验[2]。在手机端进行用户姿态识别受到硬件计算性能与存储空间的限制，在有限的计算资源下实现高效的数据处理与模式识别有着迫切的需要。因此，在现有基础上提出更为高效的数学模型，针对高数据处理量、低计算资源的应用场景，利用传感器数据进行动作识别分类有着重要的意义。

1.2 问题提出

基于动作识别的基本流程，需要从动作分类、动作标签判别和回归预测人物画像角度考虑，解决以下 3 个问题：

(1) 问题一：动作分类问题。在已知动作数量为 12 个的条件下，建立分类模型，对附件 1 中无活动状态标签的活动数据进行分类，不同用户的同一个动作需要分到同一类型中；

(2) 问题二：动作标签判别问题。利用附件 2 中标定了活动状态标签的数据对问题 1 中建立的模型进行准确度检验与分析，进一步建立判别模型，利用模型标定附件 3 中活动数据的状态标签；

(3) 问题三：回归预测人物画像问题。分析不同人员同一活动状态下的活动数据差异，判断这些差异是否与人员的年龄、身高和体重等身体数据有关。若存在关系，根据附件 5 给出的五名实验人员的活动数据，判断这些数据最有可能属于哪五名测试人员。需要建立回归预测模型，寻找实验人员的活动数据与其身体数据的关系，并利用模型从活动数据推导身体数据范围，并推测活动数据可能出自的实验人员。

1.3 资料条件

附件提供了各实验人员的活动状态数据, 各文件的详细说明如下:

- 附件 1 提供了 3 名实验人员的 12 类未带活动状态标签的活动状态数据，每名实验人员每类活动状态数据有 5 组，共 60 组数据，SYx.xlsx 为某活动状态的测量数据。
- 附件 2 提供了 3 名实验人员的 12 类带活动状态标签的活动状态数据，每名实验人员每类活动状态数据有 5 组，共 60 组数据，axty.xlsx 为某活动状态的测量数据。
- 附件 3 提供了某实验人员的 30 组活动状态数据，SYx.xlsx 为某活动状态的测量数据。
- 附件 4 提供了 13 名实验人员的身高、体重和年龄。
- 附件 5 提供了 5 名未知人员的 12 类带活动状态标签的活动状态数据。

2 模型假设与符号说明

2.1 模型基本假设

(1) 假设各组实验重力加速度差别忽略不计；
(2) 假设采集的样本之内独立同分布，样本序列信号认为是平稳信号；
(3) 假设测量信号的收发设备为同一设备，测量过程受到温度，光照等实验环境引起的噪声变化可以通过去噪算法同等的去除，且系统误差可以通过去中心化的方式消除；
(4) 对数据的保留精确到小数点后 9 位，因此引起的计算累计误差不会造成分类的错误。

2.2 符号说明

本文定义了如下 16 个使用次数较多的符号, 其余符号在使用时注明。

表1 符号说明

符号	含义
gyro	陀螺仪测量的角速度数据
acc	加速度计的测量加速度数据
row_{data}	滤波处理前数据
$filtering_{data}$	滤波处理后数据
μ_k	高斯分布的均值
Σ_k	协方差矩阵
π_k	混合系数
x_i	第 i 个样本的特征向量
k	行为动作类别
$\gamma(z_{ik})$	第 i 个样本属于第 k 类行为的后验概率
w	权重向量
b	偏置项
ξ_i	松弛变量
C	惩罚参数
f(x)	决策函数
ϵ	误差项

3 加速度计和陀螺仪数据处理

3.1 数据预处理

由于采集数据时，环境复杂多变，所得到的运动数据就会存在一些噪声以及缺失。直接使用这些数据进行姿态识别，会产生较大偏差，降低识别效率与识别精度。因此必须对原始数据进行预处理操作。这一部分主要包括：合成一维序列信号，利用插值去除原始数据空值、去均值、卡尔曼滤波去噪，数据分割[3] 。

3.1.1 合成一维序列信号

在给定的动作中，除去一些特定的动作（如电梯行动，左右走等）需要使用特定的单轴来辨别方向之外，通常将三轴加速度合成一个加速度值。为了降低传感器摆放位置等外界条件的影响，使运算变得更加简单，将三维加速度数据合为一维加速度，来描述人体的运动情况。加速度计和陀螺仪记录的数据分别按照下式进行处理：

$$gyro_{total} = (gyro_x^2 + gyro_y^2 + gyro_z^2)^{1/2} \tag{1}$$

$$acc_{total} = (acc_x^2 + acc_y^2 + acc_z^2)^{1/2} \tag{2}$$

式中 *gyro* 表示陀螺仪测量的角速度数据，acc 表示加速度计的测量加速度数据。下标的 *x,y,z* 分别是三个给定方向上的分量。下图展示了这些数据的信号形态。

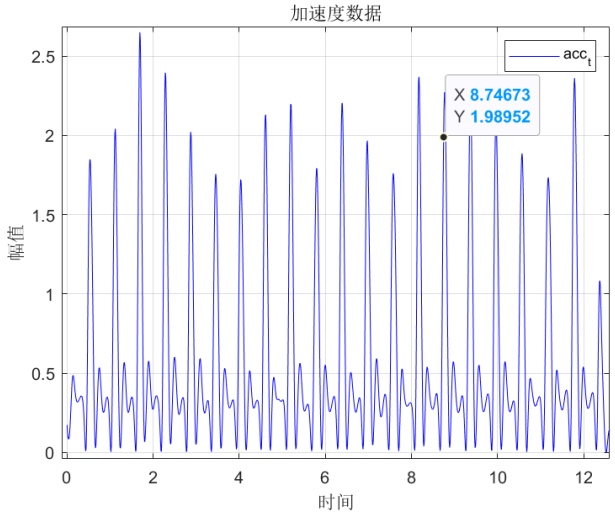


图1　合加速度信号

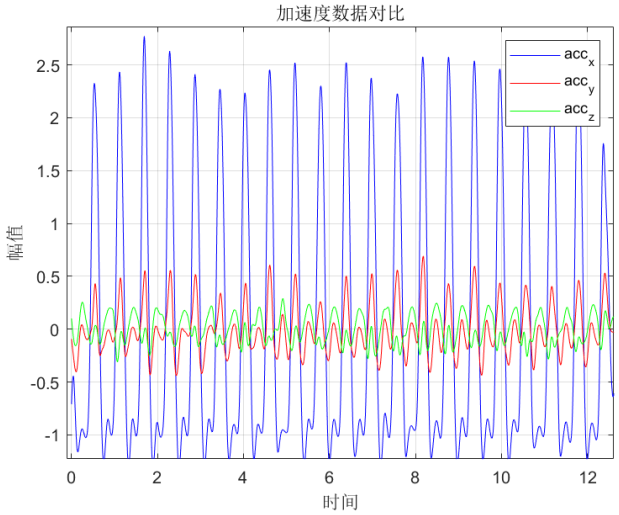


图2　三维分量上加速度信号

3.1.2 去空值与去均值

采集到的运动数据中，可能会存在部分空值，直接使用此数据进行特征处理等操作，会降低算法精度，影响算法效率，因此，需要进行去除空值操作。去空值的常用方法有插值法和拟合法。

由于特征值比较大时，会使分类算法在训练过程中存在过拟合现象，使算法的识别结果不理想，因此，需要对运动数据进行去均值处理，使得运动数据的各个维度都中心化趋向于 0。

本文所使用的去空值方法是插值法，具体而言，通过使用均值滤波，将空值周围的点进行加权求和的方式来替换空值。

本文使用的去均值方法则是在原始数据上减去其序列和的平均值。如图 3 所示.

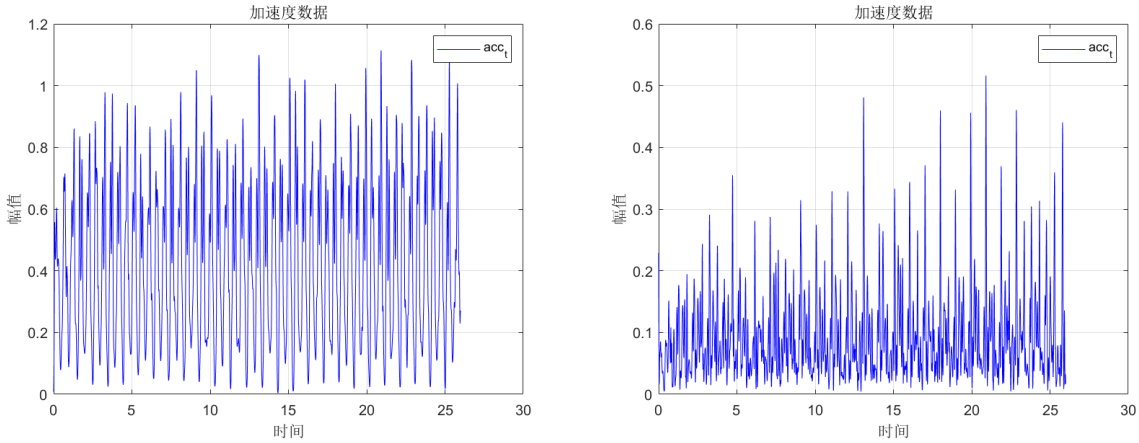


图3 去均值前后数据对比(左图：未去均值加速度数据，右图：去均值后加速度数据)

3.1.3 卡尔曼滤波

噪声的干扰会导致算法精度的降低。为提高识别的准确率与识别效率，需要对采集数据进行滤波，降低噪声的影响。在本次建模中，由于传感器采集的序列信号是典型的平稳随机过程，因此我们选择使用卡尔曼滤波进行去噪。

卡尔曼滤波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。卡尔曼滤波算法是一种最优化自回归数据处理算法，它在计算当前时刻的估计值时，需要使用前一时刻的估计值，以及当前时刻的观测值，来更新状态变量的估计值，在计算的过程中所使用均方误差最小的估计准则。

在实际实现中，通过调用 Matlab 函数实现卡尔曼滤波。3.1.4 图 4 以加速度为例，展示了一组数据经过卡尔曼滤波处理后的时序信号图像。row_{data} 和 $filtering_{data}$ 分别表示处理前后的数据。可以看到，滤波后的信号保留了信号的特征并且更加平滑，为后续的特征提取打下基础。

3.1.4 数据分割

在采集数据的过程中，实验人员是持续运动的，为了防止过长的实验时间对特征提取的影响，需要对数据进行加窗分割。数据分割是一种简单有效的预处理方法，有助于增强算法的鲁棒性和泛化能力。常用的数据分割技术包括基于滑动窗口的分割、基于事件定义的窗口分割和基于动作定义的窗口分割。在本次建模中，选取了基于滑动窗口的分割方法。如图 5 所示淡黄色矩形表示被窗口采样的数据，窗口大小设置为 200，步长为 100，重叠区域设置为窗长的一半。

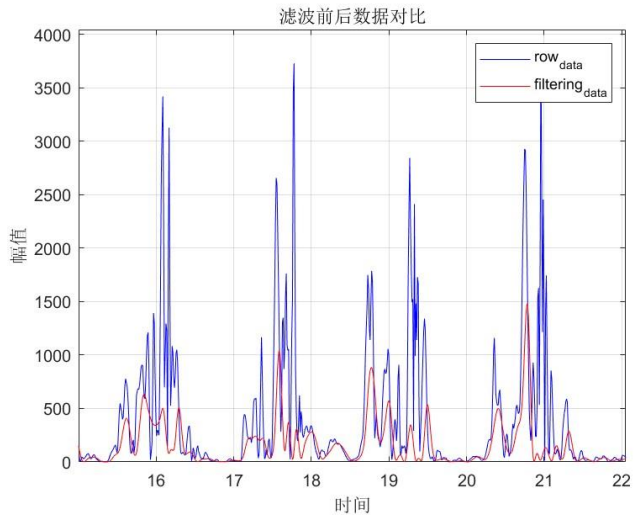


图4 滤波前后数据对比

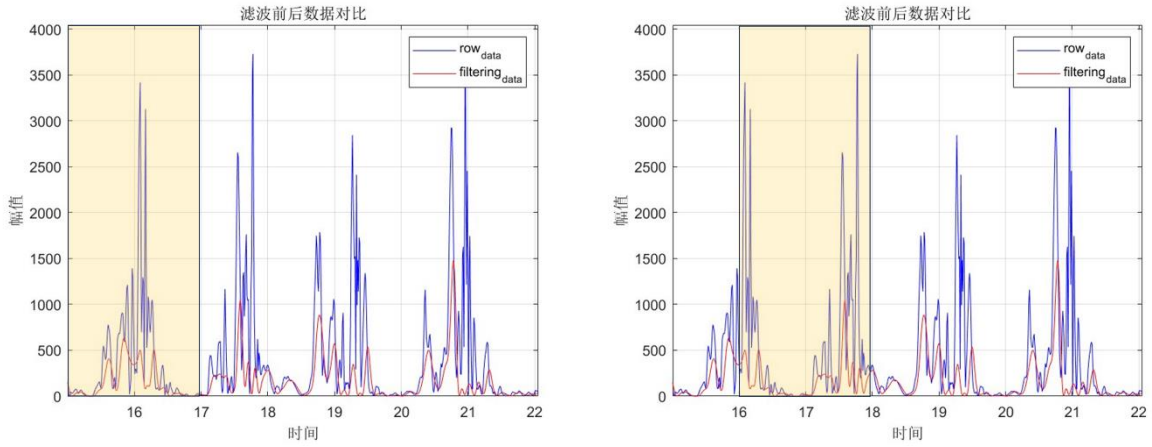


图5 滑动窗口流程详解

3.2 数据特征处理

批量的样本数据是建模的基石。由于原始数据冗余信息较多，数据长度较大，需要提取预处理后的数据特征，将特征作为运动状态识别的输入[4]-[7]。特征提取方法主要包括时域特征提取、频域特征提取。本文涉及的特征提取方式，作用在直接获取的三个维度 *x,y,z* 上的加速度信号 *acc* 和角速度信号 *gyro*,以及合成的一维信号 acc_{total} , $gyro_{total}$ 上，总计 8 个维度。

3.2.1 特征提取

为了完整表达信号统计信息，选择时域频域共计 19 种特征，其中包括有量纲，无量纲，自定义特征三大类，并按照 1~152 进行编号。如表 2 所示：

表2 特征列表

特征	物理含义	编号
Mean():均值	平均值	1~8
Max():最大值	上限	9~16
Min():最小值	下限	17~24

特征	物理含义	编号
Median():中位数	中间值	25~32
Std():标准差	信号稳定程度	33~40
Prctile():四分位点（上中下）	信号的稳定度，规避奇异点	41~64
Kurtosis():峰度	信号的陡峭程度	65~72
Skewness():偏度	信号的偏斜程度	73~80
Rms():均方根	信号有效值	81~88
DC:直流分量	所有信号频率的能量	89~96
PSD:功率谱密度幅值	功率谱密度	97~104
二次谐波频率	主频频率	105~112
波形因子	脉冲因子/峰值因子	113~120
峰值因子	峰值在波形中的极端程度	121~128
峰谷差	波峰、波谷跨度大小	129~136
频率标准差	频谱的分散程度	137~144
均值频率	频域振动能量的大小	144~152

3.2.2 特征处理

时域特征和频域特征相结合，使得特征能够更加全面的表示原始数据，但会导致特征维数过多，增大后续计算的计算量，使得算法速度与准确度降低。因此，需要对特征集进行处理，筛选最主要的特征信息，主要的特征处理方法有特征筛选和特征降维[9]-[12]。

特征筛选和特征降维可以分为有监督算法和无监督算法。有监督算法需要标签数据，而无监督算法不需要标签数据。本文分类模型使用不带活动状态标签的数据，因此使用无监督算法进行特征处理；判别模型使用带标签的数据，使用有监督算法进行特征处理。下面将对各个特征处理方法进行介绍和比较。

常用特征筛选算法主要有无监督特征选择（Unsupervised Feature Selection，UFS）[13]、Relief-F [14]、信息增益(Information Gain, IG)[15] 等，其中 Relief-F 和信息增益算法为有监督算法，而 UFS 为无监督算法。常用的特征降维方法有主成分分析(Principal Component Analysis，PCA)和线性判别分析(Linear Discriminant Analysis，LDA),前者为无监督算法，后者为有监督算法。

Relief-F 比较每个特征下随机样本与同类近邻样本的距离和与异类样本的距离，同类距离小的特征则增加权重，反之，则降低权重。重复多次迭代以更新权重；信息增益算法通过计算信息熵评估特征对分类结果的影响程度来进行特征筛选，以减少分类不确定性的特征；UFS 算法通过求解稀疏特征问题和 L1 规则化最小二乘法问题，选择最好的保留数据多集群结构的特征。

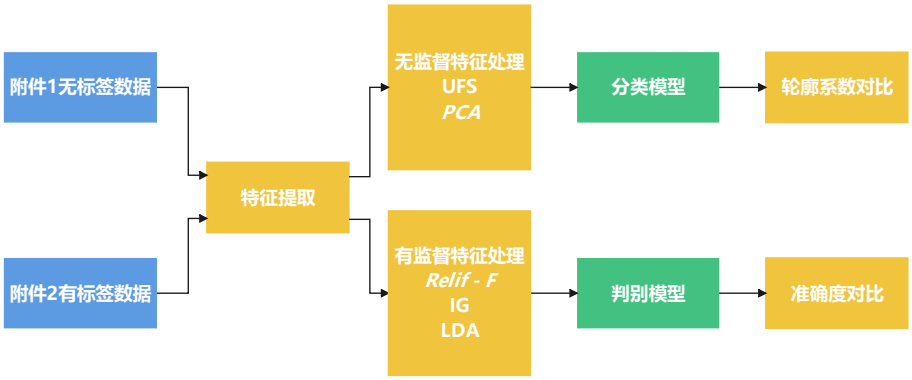


图6 特征提取流程

PCA 将维数较高的数据投影到数据中的方差最大的方向，寻找类间间距最大的特征向量投影[16] 。LDA 算法将维数比较高的数据投影到最佳鉴别矢量空间，从而达到相同种类的数据间的方差是最小的，而种类不同的数据之间的方差最大的结果[17] 。

为了选择最有效的特征处理方法，本文对比了这几种算法的处理能力与处理后的特征性能。

使用分类模型进行聚类处理，计算不同无监督算法的对实验结果的影响。本文使用到附件 1 中给定的不带标签的 3 个人 180 个数据样本，按照 3.1 步骤进行预处理后（此时已经经过分割，因而获得更多的样本数）获得的 6390 组长度为 200 帧样本数据，每组样本数据包含其原始的时序信号以及提取的 152 种特征。将这些 6390 组特征数据随机划分为 3 组数据集，每组包含 2130 个样本数据。使用轮廓系数，即某点到各个非本身所在簇的所有点的平均距离作为衡量标准，轮廓系数越大表示聚类效果越好，实验结果如表 3 所示。

表3 无监督特征处理算法对比

数据集 （2130 样本，152 个特征）	保留特征维数	轮廓系数
1 号数据集	**PCA：32**	**0.73**
	UFS：29	0.61
2 号数据集	PCA：27	0.62
	UFS：30	**0.65**
3 号数据集	**PCA：30**	**0.87**
	UFS：25	0.70

使用判别模型进行类型判别，对比不同有监督算法的对实验结果的影响。对于附件 2 中带有标签的 10 个人 600 个数据样本，同样按照 3.1 步骤进行预处理，获得的 21300 组长度为 200 帧样本数据，每组样本数据包含其原始的时序信号以及提取的 152 种特征。分为 3 组 7100 个样本数据及对应标签的数据集，用这些数据集按照 9：1 的比例作为特征筛选实验的训练集及测试集。使用判别模型进行判别测试并保留计算得到的分类准确率达到最大时所选的特征数，实验结果如表 4 所示。

表4 有监督特征处理算法对比

数据集 （7100 样本，152 个特征）	保留特征维数	准确率
1 号数据集	**Relief-F：26**	**85.4%**
	IG：20	81.1%
	LDA：26	84.9%
2 号数据集	Relief-F：23	82.5%
	IG：23	82.4%
	LDA：30	**82.7%**
3 号数据集	**Relief-F：24**	**88.2%**
	IG：22	84.8%
	LDA：25	84.8%

由表可知，使用 PCA 算法和 Relief-F 算法能够在兼顾准确度的同时，尽可能少的提取特征数量，这有利于特征的精准选取，提升模型性能。因此，本文选用 PCA 作为分类模型的特征处理方法，选用 Relief-F 作为判别模型的特征处理方法。

特征筛选过后，由于不同特征的量纲不同，量级会有较大的差异，过大的特征值会使分类算法在训练过程中存在过拟合现象，使算法的识别结果侧重在该特征值而忽略其他小特征值的作用中，不利于信息的充分利用，因此需要进行量纲统一。本文采用的统一方式是最大-最小归一化（max-min normalization），将原始的特征数据线性映射到[0，1]，这样的处理不会改变已有数据的性质和相对关系。处理的方式如(3)

$$x' = \frac{x - \min_{\text{feature}}}{\max_{\text{feature}} - \min_{\text{feature}}} \tag{3}$$

其中,$\mathbf{x}'$是归一化后的值,$\mathbf{x}$ 是原始数据点,$\min_{\text{feature}}$ 和 $\max_{\text{feature}}$ 是该特征在整个数据集中的最小值和最大值。

4 问题一分析与模型建立

4.1 问题一分析

问题一要求建立分类模型，根据附件 1 中 3 名实验人员的运动数据，将其活动状态分为 12 类。分析加速度计与陀螺仪数据，不考虑具体活动状态，仅需要作种类区分。在没有标签的情况下，根据数据特征本身的相似度，对数据进行类别的划分。题目给出的数据包含 12 类活动数据，每类活动数据有 5 组数据，因此可设定分类模型的簇数为 12。

为了以最少的特征数量提取足够的特征信息，本题中使用 PCA 进行特征降维，在减少冗余特征过拟合风险的同时，最大化聚集有用特征，将其投影到低维子空间中，进

一步减少了计算量。由于量纲的有无及不同，特征矩阵的数值变化幅度很大，会严重影响算法的效果，因此首先需要对特征矩阵按行进行归一化。

聚类算法选择很重要，本文使用 K 均值聚类（K-means）方法和高斯混合模型（Gaussian Mixture Model，GMM）两种聚类算法，分别对每名实验人员的活动状态数据进行分类。在分类完成后，需要再次进行横向分类以实现不同实验人员活动状态样本的对齐。通过对 3 名实验人员分类后的活动特征进行相似度比较，匹配 3 名实验人员相同的活动状态，得到分类结果，图 7 给出了问题一的实现流程。

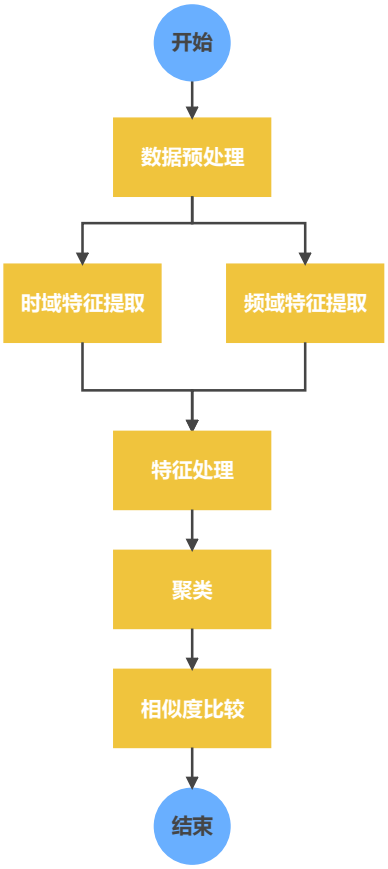


图7 问题一思路流程图

4.2 分类模型指标

分类模型效果通常可以通过以下几种指标进行衡量：

4.2.1 内部指标

轮廓系数：计算每个样本点的轮廓系数，综合考虑了样本与其自身簇内的距离和与其他簇的距离，取值范围为[-1, 1]，值越接近 1 表示聚类效果越好。

戴维森堡丁指数（Davies-Bouldin Index，DBI）：聚类内部的紧密度和不同簇之间的分离度的比值，值越小表示聚类效果越好。

4.2.2 KL 散度

KL 散度是一种衡量两个概率分布之间差异的指标，它不是一种直接用于评估聚类效果的指标，而是用于度量生成的样本分布与真实数据分布之间的差异。KL 散度的计算公式如(4)。

$$D_{KL}(P \| Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{4}$$

其中，P(i)和 Q(i)分别是 P 和 Q 在第 i 个事件上的概率。

4.2.3 本文采用的衡量指标

由于在问题一中未给出明确真值，但题中给定了先验知识：每种类别均有 5 个样本数据，给定了真实数据的分布函数。因此我们采用内部指标轮廓系数，DBI，以及 KL 散度作为衡量指标。

4.3 分类模型建立与应用

4.3.1 K-means 算法简介

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对象划分为 K 个类别，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。通过迭代的方式不断优化聚类结果，使得每个聚类内的对象尽可能紧密，而不同聚类间的对象则尽可能分开。

4.3.2 GMM 算法简介

GMM 可以用来对数据进行分类，其假设数据点是由一个或多个高斯分布生成的，并通过最大似然估计的方法来估计每个簇的高斯分布的参数。GMM 聚类算法可以用于许多领域，如人脸图像分类和语音分类等。

4.3.3 算法流程

具体来说，K-means 算法的执行流程包括以下几个步骤：

Step 1 随机选择 K 个特征向量 x_i 作为初始的行为动作类别簇质心 k_i；
Step 2 根据每个特征向量与各个簇质心的距离，将其分配给最近的簇；
Step 3 重新计算每个簇的质心, 取簇内所有特征向量的平均值作为新的质心；
Step 4 重复 **Step2-Step3**。直到聚类中心不再变化。

GMM 算法执行流程如下：

Step 1 随机初始化每个高斯分布的均值 μ_k、协方差矩阵 Σ_k 和混合系数 π_k,下标 k 表示第 k 个高斯分布；

Step 2 计算每个特征向量 x_i 属于每个行为动作类别 k 的后验概率 $\gamma(z_{ik})$,其中 z_{ik} 是指示变量，表示第 i 个特征向量属于第 k 个类别的概率.后验概率计算公式如

$$\gamma(z_{ik}) = \frac{\pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)} \tag{5}$$

Step 3 使用当前的后验概率 $\gamma(z_{ik})$ 更新模型参数(μ_k， Σ_k， π_k)：

更新混合系数 π_k:

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma(z_{ik}) \tag{6}$$

更新均值 μ_k:

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma(z_{ik}) x_i}{\sum_{i=1}^{N} \gamma(z_{ik})} \tag{7}$$

更新协方差矩阵 Σ_k：

$$\Sigma_k = \frac{\sum_{i=1}^{N} \gamma(z_{ik})(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{N} \gamma(z_{ik})} \tag{8}$$

Step 4 重复 **Step 2~Step3**，直到模型收敛；

Step 5 得到每个特征向量属于每个行为动作类别的后验概率 $\gamma(z_{ik})$ 以及最终的均值 μ_k、协方差矩阵 Σ_k 和混合系数 π_k。

基于以上两种算法，本文分别对问题一中提取的特征进行聚类，下面是本文模型分类结果对比与分析。

4.3.4 模型分类结果对比与分析

本文对两种算法的输入数据进行了相同的预处理。在经过同样的步骤提取特征后，固定随机数种子，以便观察实验的结果。对于 K-means，设置簇数为 12，采用高斯核对 GMM 进行初始化。同样将 3.2.2 中使用到的划分数据集的方法用于问题一中数据集划分，获得 3 个具有 2100 个样本的数据集进行实验。由于模型在分类时只能固定簇数为 12，不能够固定每一簇包含 5 个样本，因此，分类后的结果不一定会生成均匀的 12×5 的分类结果。为得到均匀的分类结果，本文对每名实验人员的活动状态进行分类时，调用 100 次分类模型，统计每次分类结果，根据分类结果出现的频次，得到均匀的 12×5 的分类结果。由于缺乏标签检验，本文采用了两种方法进行聚类效果的衡量：一是观察聚类后数量的分布是否均衡，这是最直观而准确的衡量方式，由于题意给定的每种类别的样本数为 5，分类均匀是分类正确的必要条件。对于分类是否符合均匀分布，度量生成的样本分布与真实数据分布之间的差异，我们采用 KL 散度进行衡量。二是使用内部指标轮廓系数和 DBI 作为衡量指标，模型分类指标如下表：

表5 两类模型分类指标评价

聚类方法	数据集（2100 个样本）	轮廓系数	DBI	KL 散度
K-means	1 号	0.6524	0.6215	0.0845
	2 号	0.5414	0.6619	0.0205
	3 号	0.8410	0.7452	0.0192
GMM	1 号	0.5528	0.5941	0.00673
	2 号	0.5942	0.6619	0.01732
	3 号	0.7412	0.7406	0.00926

从表中数据可以观察得出，在内部指标的衡量上，k-means 算法与 GMM 各有优劣，比如 K-means 在轮廓系数上更有优势，而 GMM 在 DBI 指标上更接近 0，更具有优势。这说明，内部指标并不能很好的区分出两种算法的优劣。不能仅仅根据内部指标判断两种算法的优劣。但是，在度量生成的样本分布与真实数据分布之间的差异时，GMM 算法相较于 K-means 方法的 KL 散度数值更小，换言之，GMM 算法最终分类的样本分布更为接近理想的真值分布。在这种情况下，从符合题意中给定的先验信息来考虑，GMM 算法更加具有优势，更适合问题一的求解。因此最终我们选择使用 GMM 算法作为最终的聚类方法。

设定 GMM 为问题一的实现算法，采用高斯核初始化参数，输入样本为：由问题一给定的三名实验人员分别采样得到的 60 组数据划分滑动窗口形成的 2100 个样本，由这些样本提取特征组成的 2100×12 的特征矩阵，其行代表样本而列代表特征。簇数为 12，迭代直至收敛。

在分别对每名实验人员进行聚类分析行为后，由于表格中横向对齐的要求，本文采用了距离度量算法，对三名实验人员的不同行为进行特征空间上的距离衡量，即相似度检验。相似性度量与距离度量本质上是同一件事情。如果两组数据之间的距离越大，那么相似性越小；反之，相似性越大，距离越小。距离近的特征所对应的行为应当划分在同一类别中，也就是在同一列中。

具体而言，在将每名实验人员的行为划分为 12×5，共计三组后，我们从 3 组，12 个类别中的 5 个样本随机抽取一份样本作为验证样本，共计抽取 3×12=36 分样本。然后，对这些样本对应的特征向量分别使用马氏距离进行距离度量，这种度量方式相较于欧氏距离能消除特征之间的相关性的影响。最终，问题一的分类算法结果如表 6 所示。

表6 问题 1 分类结果

分类	Person1	Person2	Person3
第1类	SY7,SY33,SY38,SY42,SY58	SY2,SY10,SY12,SY26,SY47	SY13,SY27,SY29,SY51,SY56
第2类	SY14,SY17,SY28,SY41,SY47	SY16,SY21,SY29,SY40,SY43	SY9,SY14,SY30,SY37,SY43
第3类	SY5,SY8,SY46,SY56,SY59	SY3,SY20,SY38,SY51,SY60	SY8,SY23,SY39,SY42,SY52
第4类	SY24,SY25,SY29,SY48,SY53	SY7,SY24,SY33,SY44,SY57	SY11,SY25,SY31,SY44,SY60
第5类	SY27,SY31,SY37,SY49,SY52	SY19,SY23,SY30,SY37,SY46	SY19,SY24,SY35,SY36,SY49
第6类	SY19,SY26,SY43,SY44,SY50	SY1,SY4,SY48,SY49,SY50	SY10,SY41,SY57,SY58,SY59
第7类	SY1,SY2,SY15,SY23,SY36	SY13,SY27,SY28,SY34,SY42	SY4,SY5,SY18,SY22,SY53
第8类	SY4,SY13,SY18,SY39,SY57	SY5,SY11,SY22,SY54,SY56	SY6,SY17,SY21,SY33,SY38
第9类	SY9,SY10,SY12,SY20,SY24	SY6,SY15,SY25,SY35,SY58	SY2,SY12,SY34,SY47,SY48

分类	Person1	Person2	Person3
第10类	SY22,SY32,SY34,SY35,SY40	SY31,SY32,SY39,SY52,SY59	SY3,SY32,SY50,SY54,SY55
第11类	SY3,SY6,SY11,SY16,SY55	SY8,SY9,SY36,SY41,SY55	SY1,SY7,SY26,SY40,SY45
第12类	SY30,SY45,SY51,SY54,SY60	SY14,SY17,SY18,SY45,SY53	SY15,SY16,SY20,SY28,SY46

5 问题二分析与模型建立

5.1 问题二分析

问题二要求建立判别模型，根据活动状态数据判断实验人员的活动状态。问题一建立的分类模型对活动状态进行种类区分，问题二建立的判别模型则在其基础上进一步进行标签标定。

下面详细分析问题一、二的异同，问题二在问题一的基础上，其数据内容发生了改变，一方面，实验的数据增多，由原本的三名实验人员增加到了十名实验人员。实验人员的增多，也就是所需处理的数据增多，会导致对算法训练时间的进一步约束，要求算法处理能力的提高；另一方面，实验二给出了带标签的数据，目标也由原来的聚类变成了实现对未知人员的 30 次活动状态的分类问题，这标志着第一问的分类模型不能直接应用于第二问，而是需要设计另一种判别算法实现更高精度的模型，是典型的“训练”-“测试”结构。

下面，详细介绍本文采用的解决思路。首先，由于对运算速度的要求，需要进一步提取更具有代表性且数目更少的特征，去除冗余数据，因此在数据预处理的过程中，为了充分利用到标签信息，本文不再采用无监督的算法 PCA 和 UFS，而是采用利用到标签信息的 LDA 和 Relief-F 算法，LDA 能充分利用到类别信息，在进行降维的同时选择向类间距离最大的子空间方向进行投影；而 Relief-F 算法法采用距离去衡量同类、异类数据中特征的聚集、离散程度。由于单独采用某一种降维或聚类方法会导致该类特征排序算法表现出对数据信息应用的单一性。结合以上分析，本文提出多信息融合的特征筛选算法。

进一步，在进行更细致的特征提取和筛选后，需要使用高效的分类算法进行问题二的求解，建立判别模型。由于本文中的数据特征数较多，数据维度大，需要一种适用于高维空间且不易受到异常特征影响的算法建模判别模型。在本文中采用支持向量机（Support Vector Machine，SVM），虽然朴素 SVM 仅适用于二分类问题，但通过一对其他策略，可以将其拓展为多分类问题的求解手段。因此，最终以附件 2 中已经标定活动状态标签的 Person4 至 Person14 数据提取得到的特征矩阵为先验知识对 SVM 进行有监督学习。

对于问题二中的三个任务，首先，我们使用 Person4 至 Person14 的数据对 SVM 进行训练；其次，我们使用问题一中的 GMM 算法对问题二 Person4 至 Person14 的数据进行分类，验证 GMM 的准确度。之后，使用 SVM 算法对 Person1 至 Person3 数据进行判别，比较分析问题二中判别模型 SVM 和问题一的分类模型 GMM 的结果。最后，使用 SVM 对附件 3 中 SY1 至 SY30 数据进行活动状态判别。图 8 给出了本文问题二的思路分析。

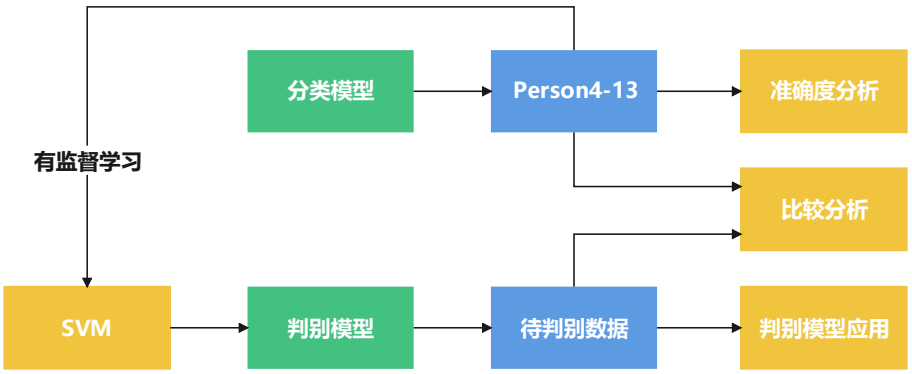


图8　问题二思路流程图

5.2 判别模型指标

5.2.1 精确率

表示被分类为正类别的样本中，实际为正类别的比例。即 $\text{Precision}=\frac{TP}{TP+FP}$，其中 TP 是真正例（被正确分类为正例的样本数），FP 是假正例（被错误分类为正例的样本数）。通常不会直接使用该公式运算，而是计算平均精确率（Average Precision，AP）或者平均类别精确率（Mean Average Precision，mAP）

5.2.2 召回率

表示实际为正类别的样本中，被正确分类为正类别的比例。即 $\text{Recall}=\frac{TP}{TP+FN}$，其中 TP 是真正例，FN 是假反例（被错误分类为负例的样本数）。

5.2.3 F1 分数

是精确率和召回率的调和平均数，可以看作是这两者的综合度量。即 $\text{F1-score}=2\cdot\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$。

5.2.4 混淆矩阵

显示分类器预测结果与真实标签之间的对应关系。

5.2.5 本文采用的衡量指标

本文采用平均精确率（Mean Average Precision，mAP），F1 分数，混淆矩阵三种指标，衡量 SVM 判别模型的准确性。

5.3 判别模型建立与应用

5.3.1 SVM 算法简介

SVM 主要用于解决模式识别领域中的数据分类问题，属于有监督学习算法，其基本原理是构建一个能够对数据进行分类的超平面，使得不同类别的数据点尽可能远离这个超平面，从而在分类过程中具有较好的泛化能力。同时，由于 SVM 通过非线性映射函数，把样本空间映射到一个高维的特征空间，使得在高维空间中可以应用线性学习的方法解决原始样本空间中的非线性问题。同时，通过应用核方法，无需求解复杂的映射函数即可求得内积，大大减小了计算复杂度。

在本次建模中，由于特征维度大，分类数量多，特征向量的非线性明显，因此本文采用了容忍度较高的软间隔非线性 SVM 作为求解的算法，并采用一对其它策略将二分类拓展至可以解决 12 类分类问题。

5.3.2 算法流程

Step 1 包含 m 个来自行为动作训练集的训练样本$(x_i, y_i)_{i=1}^m$，其中$x_i \in \mathbb{R}^n$是特征向量，$y_i \in \{1,-1\}$是对应的类标签。选择的核函数为高斯径向基核函数$K(x,x') = \exp(-\gamma \mid x-x' \mid^2)$，其中$\gamma$是超参数。将输入特征映射到高维空间中。通过最小化带有正则化项的优化目标来找到最优的决策函数：

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2} \mid w \mid^2 + C\sum_{i=1}^{n} \xi_i \\ &\text{st } y_i(w \cdot x_i + b) \geq 1-\xi_i, \quad \xi_i \geq 0, \quad \forall i = 1,\ldots,n \end{aligned} \tag{9}$$

其中,w 是权重向量, b 是偏置项,ξ_i是松弛变量,C 是惩罚参数，用于控制间隔的大小和误分类的数量。

Step 2 为了处理约束条件，SVM 引入拉格朗日乘子$\alpha_i \geq 0$,然后，构建拉格朗日函数如下：

$$\mathcal{L}(w,b,\alpha,\xi,\mu) = \frac{1}{2} \mid w \mid^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left[y_i(w \cdot x_i + b) - 1 + \xi_i \right] - \sum_{i=1}^{n} \mu_i \xi_i \tag{10}$$

Step 3 解拉格朗日函数的对偶问题，即最大化关于α的函数$g(\alpha)$,其中：

$$g(\alpha) = \inf_{w,b} \mathcal{L}(w,b,\alpha) \tag{11}$$

最终得到的对偶问题如下：

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j) \\ &\text{st } \mathrm{C} \geq \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0_j, i = 1,2,\ldots N \end{aligned} \tag{12}$$

Step 4 这个对偶问题是一个凸优化问题，可以通过一些优化算法来求解。求解得到的α^*后即可得到支持向量的重要性及其对应的权重，从而确定最终的超平面w^*和偏置b^*：

$$\begin{aligned} w^* &= \sum_{i=1}^{n} \alpha_i^* y_i K(x_i \cdot x_j) \\ b^* &= y_j - \sum_{i=1}^{n} \alpha_i^* y_i K(x_i \cdot x_j) \end{aligned} \tag{13}$$

Step 5 得到的分类决策函数为：

$$f(x) = \text{sign}(\sum_{i=1}^{n} \alpha_i^* y_i K(x_i \cdot x) + b^*) \tag{14}$$

构建 12 组 SVM，每组分别实现一个类别的是与否的判别，最终，12 组 SVM 可以区分出不同类别并进行正确的划分。

5.3.3 SVM 算法分类结果

设定 m=21300（样本数），为多次实验消除偶然性，将这些样本随机划分为三个等大小的数据集，每个数据集中的类别及样本分布是一致的，数量为 7100 组样本。每组实验设置的参数一致，$\gamma=1$为默认设定，训练迭代直至收敛。图 以热力图的形式展示了分类的详细结果，表 7 展示了 SVM 分类的指标，表 7 图 9 则具体展示了 3 号数据集各种类别的 AP 值。

表7 SVM 模型指标

分类方法	数据集（7100 个样本）	F1-分数	mAP
SVM	1 号数据集	92.37%	92.32%
	2 号数据集	91.64%	90.73%
	3 号数据集	91.98%	91.25%

从表中可以得知，F1-分数的数值要普遍高于准确度 AP 值，这是由于 F1-分数对于具有不平衡类分布（即正负样本比例差别很大）的数据集特别有用，因为它同时考虑了分类器的精确性和召回率，可以帮助在精确率和召回率之间找到一个平衡点。而本题的类别均衡，故高的召回率使得 F1-score 的数值得到提升。在三种数据集中，SVM 的 mAP 在 91%左右，达到一个不错的分类水准。

从图中对详细类别准确率进行分析，我们定量描述造成 SVM 的 mAP 在 91%上下的原因。为了方便分析，我们将定义动态动作与静态动作。对于类别 1-7，我们将其称为动态动作，这些动作在录入的数据全程都有清晰明显的变化，且具有明显的周期性。而对于 8-12，由于这些类别对应的动作（以下称为静态动作）运动少，具有明显变化的部分并不贯穿于所有帧，而是在某些特定时刻发生某个维度上的明显变化。

对于 4-5，8-12 号类别，SVM 的分类效果良好，分类正确率在 95%以上，这种现象可以分成两部分进行解释，一方面，静态动作的大部分时间在各种特征上不会发生明显变化，而是在其自身关键帧时间发生强烈的变化。为了说明这一点，我们以类别 11，12 举例说明。如图 10 类别 11，12 为上下电梯时的动作，很明显，仅仅当电梯启动和停止时，加速度会在 x 轴（重力方向）发生明显的变化，其他时间，电梯匀速运动时，录入的信息不会发生明显变化；而变化的方向（也即先增后减或先减后增）则可以区分是电梯上行或下行。类似的现象会反应在所有的静态动作的特征中，比如，类别 11，12 的区别会反映在加速度 x 数据中滑窗形成的“均值”特征的大小先后变化的顺序上，与动态模型相比，静态特征区分度更加明显。8-12 的 AP 值高是符合逻辑的。

另一方面，4，5 为步行上下楼的动态动作，从一众动态动作中脱颖而出的原因是其 x 轴（重力轴）方向上的变化量数值上更大，体现在特征中的方差为所有动作中的最大值。而区分上下楼之间的特征则为均值的符号。这些区分在 SVM 中是非常简单容易提取的特征，因此对于这些类别的分类 SVM 成功达到了 96%左右的准确率。

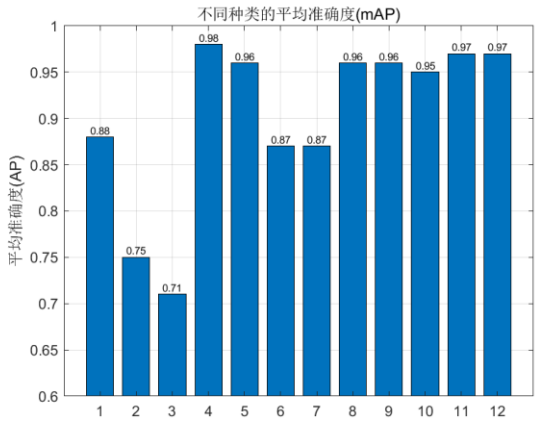


图9　3 号数据集下的 mAP

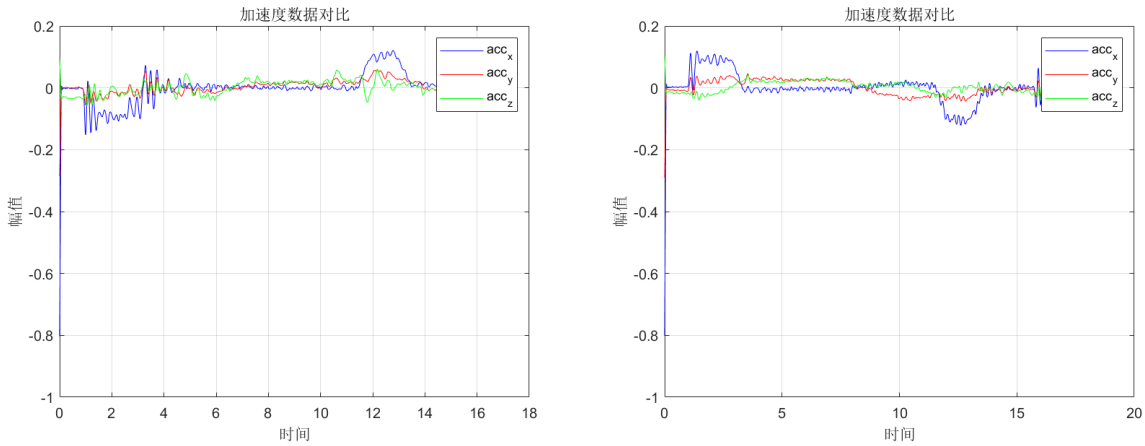


图10　类别 11，12 的原始数据图（左图：电梯上行，右图，电梯下行）

下面分析平均准确率不高的类别。总体来看，动态动作由于重复运动，特征之间相对界限不如静态动作清晰，分类器对其中的分类难度会增加，相较于静态动作其分类正确度有所下降。尤其是 2，3 类的左右走，其提取的特征更加相近，分类器无法很好的区分其中的异同；并且，由于所设立的坐标轴为质心系，x 指向地心，y 指向前，z 垂直于 y，其每个方向的分量无法对“左”，“右”进行区分，而传感器位置在右腰处，对“左”“右”事件的感知主要取决于角速度大小，这种大小上的区分与体型，步频等个人信息密切相关，因人而异，举例说明，对于某一名实验人员的分割阈值在另外一名实验人员身上会失效，因此其表现会更差。

5.3.4 GMM 分类模型准确度验证

本文使用问题一中的分类模型 GMM 来分类问题二给出的数据，验证其分类效果，通过在问题二中带有标签的数据集上进行测试，得出了分类数据。由于分类表格数据较多，本文将其放在附件中，详见支撑材料：分类模型准确度结果.xlsx。

如图 11 所示，GMM 在分类后的准确度检验以热力图形式绘制出。横坐标为识别到的分类结果个数，而纵轴为真值对应的归类个数。在斜对角线上的个数为正确分类的个数，而其他数值则是分类失败的案例。颜色越深则代表该位置对应的样本数越多。可以看出，总体而言，聚类算法由于缺少的分类的先验信息引导，在分类的准确率上要略差于利用到先验分类信息的 SVM。与 SVM 算法相似，在 4-5，8-12 这些类别上的分类效果好，明显高于其他情况下的分类效果。本文认为原因仍然归因到特征的独特性与可

区分度高；而对于 2，3 类，可以看出具有明显的相互误分的现象。这进一步验证我们在 5.4.3 中做出的判断：2，3 类的左右走，其提取的特征更加相近，特征空间中这两者十分相近，无法很好的区分其中的异同。GMM 聚类算法即使通过后验概率的计算也无法两者实现有效区分。这可能说明需要进一步获得更加具有区分度的特征以区别这些类别。

热力图中还揭示了部分局部信息：GMM 算法对于静态动作和动态动作的错估没有明显的倾向，体现在左上角中的数字和为 18 与右下角的数字和为 16，且随机分布这可能是因为 GMM 在数据空间中的高斯混合模型有时会模糊边界或者没有明确的分界线，导致分类器在某些情况下难以区分这些边界。

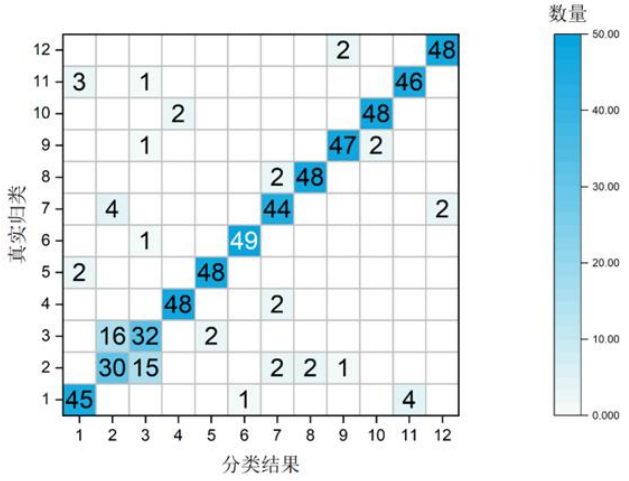


图11 GMM 分类准确度热力图

5.3.5 SVM 判别模型判别结果

使用 SVM 判别模型求解问题二（2），对收集的某实验人员 30 次活动的状态数据进行预测，给出该人员的活动状态如表 8 所示。

表8 问题 2 判别模型结果

类型	状态	类型	状态	类型	状态
SY1	步行下楼	SY11	站立	SY21	向前跑
SY2	向前走	SY12	跳跃	SY22	向左走
SY3	跳跃	SY13	步行上楼	SY23	坐下
SY4	乘电梯上	SY14	向右走	SY24	步行下楼
SY5	跳跃	SY15	步行上楼	SY25	向左走
SY6	躺下	SY16	向前走	SY26	站立
SY7	向左走	SY17	步行上楼	SY27	坐下
SY8	向前跑	SY18	步行下楼	SY28	步行下楼
SY9	跳跃	SY19	坐下	SY29	向前跑
SY10	躺下	SY20	坐下	SY30	步行下楼

6 问题三分析与模型建立

6.1 问题三分析

问题三给出 13 名实验人员的年龄、身高和体重等身体数据，要求分析不同人员的同一活动状态是否存在差异，探究活动状态数据与人员的身体数据是否存在关系，若存在关系，根据附件 5 给出的五名实验人员的活动数据，判断这些数据最有可能属于哪 5 名测试人员。

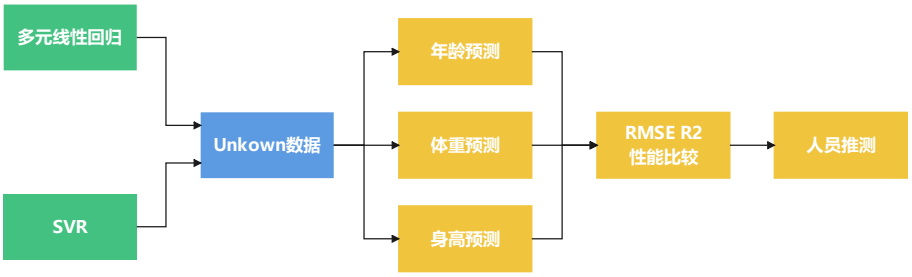


图12　问题三分析流程

首先，需要判断不同人员同一活动状态是否存在差异。本文使用问题二中给出的 10 名测试人员带有活动状态标签的数据，使用问题二中提取并筛选后的特征值，对不同实验人员做同一动作时的数据特征进行绘图，如图 13 所示，其中特征值的幅度使用归一化幅度。

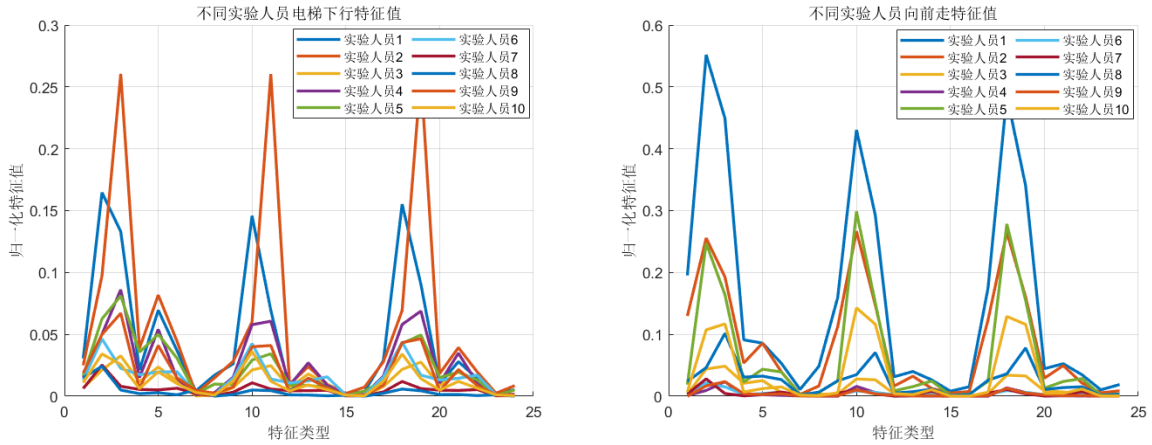


图13　不同实验人员同一活动状态特征值

通过对比特征值，不同实验人员在做同一动作时，特征值的变化具有相同的趋势，但不同实验人员的归一化特征值幅值差异明显，因此，可以通过特征图得出不同实验人员的同一活动状态存在差异。

但通过这种方式，只能判断不同实验人员的同一活动状态存在差异，不能得出该差异是由年龄、身高和体重等身体数据的不同导致的。要探究活动状态数据与实验人员的身体数据是否存在关系，本文采用多元线性回归模型和 SVR 模型进行回归分析。

6.2 回归预测模型指标

6.2.1 均方误差

MSE 是观测值与真值偏差的平方和与观测次数的比值。它是一个衡量模型预测精度的指标，MSE 的范围为[0,+∞)，预测值和真实值完全吻合时，MSE=0，即完美模型，MSE 的值越小，说明预测模型对于目标的拟合程度越精确。

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i)^2 \tag{15}$$

6.2.2 均方根误差

RMSE 是 MSE 的开根号，实质与 MSE 一样。但因为 MSE 单位量级和误差的量级不一样，而 RMSE 跟数据是一个级别的，更容易感知数据，所以更常用 RMSE 来评估模型的性能。

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i)^2} \tag{16}$$

6.2.3 决定系数

决定系数是用来衡量模型对观测数据变异的解释能力。它表示因变量方差中被模型解释的比例，取值范围从 0 到 1。通常，R^2 越接近 1，模型对数据的拟合越好。

$$R^2 = 1 - \frac{\sum_i (y_i - y_i)^2}{\sum_i (\overline{y_i} - y_i)^2} \tag{17}$$

6.2.4 本文采用的衡量指标

本文采用均方根误差和决定系数，作为回归预测模型的衡量指标。

6.3 回归预测模型建立与应用

6.3.1 多元线性回归算法简介

多元线性回归是回归分析中的一种复杂模型，它考虑了多个输入变量对输出变量的影响。与一元线性回归不同，多元线性回归通过引入多个因素，具有更高的预测和解释能力，更全面地建模了系统关系。

在本题中，要探究身高、年龄和体重等因素与实验人员活动状态是否存在关系，以身高、体重和年龄作为自变量，以实验人员活动状态的特征作为因变量，需要使用到多元线性回归代替传统的一元线性回归。

6.3.2 SVR 算法简介

SVR 是一种基于 SVM 理论的回归方法，适用于解决连续性目标变量的预测问题。SVR 与传统的线性回归或多项式回归相比，其优势在于能够处理非线性关系，并在高维空间中进行复杂的模式识别。它利用了最小化间隔和损失函数来优化模型，同时通过核函数将数据映射到高维空间中，以增加数据的线性可分性或者非线性可分性。常见的核函数包括线性核、多项式核、高斯径向基函数核等。选择合适的核函数可以显著影响 SVR 的预测性能。

SVR 通过结合 SVM 高维特征空间映射和非线性能力，提供一种有效的方法来解决回归问题。本题中，特征维度大，分类数量多，特征向量的非线性明显，因此非常适合采用 SVR 根据活动状态数据对实验人员进行回归预测。

6.3.3 算法流程

多元线性回归算法流程如下：

Step 1 准备包含因变量和多个自变量的训练数据集，假设因变量与自变量之间存在线性关系，并且误差项ε是独立同分布的，并且服从正态分布；

Step 2 通过最小化预测值$\hat{y}$与实际观测值y之间的残差平方和来拟合模型。即，找到使得残差平方和最小的参数$\boldsymbol{\beta}$；

Step 3 使用最小二乘法或其它优化算法，估计出最优的参数。

SVR 算法流程如下：

Step 1 准备训练数据集，通常需要进行特征缩放操作，以确保各个特征在相似的数值范围内，以提高 SVR 模型的性能；

Step 2 SVR 的目标是建立一个回归函数，该函数能够预测目标值，通过最小化目标函数的误差来拟合数据。目标函数和约束通常表示为

$$\begin{cases} \min \dfrac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ st. y_i - (\boldsymbol{w}\cdot\boldsymbol{x}_i + b) \le \varepsilon + \xi_i \\ (\boldsymbol{w}\cdot\boldsymbol{x}_i + b) - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \tag{18}$$

其中，$\boldsymbol{w}$和$\boldsymbol{b}$分别是回归平面的权重和偏移量，ξ_i和$\xi_i{}^*$是松弛变量，表示允许误差的量，C是正则化参数，控制模型的复杂度；

Step 3 选择合适的核函数，通常包括线性核、多项式核、高斯径向基函数核等；

Step 4 使用ε-不敏感损失函数进行模型训练，估计出最优权重向量$\boldsymbol{w}$和偏置项$\boldsymbol{b}$，ε-不敏感损失函数表示为

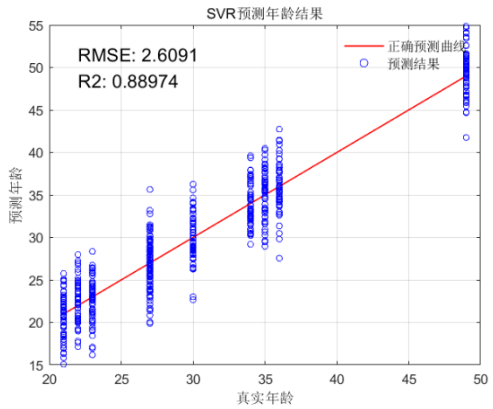
$$L_\varepsilon(y, f(x)) = \begin{cases} 0 & |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon & otherwise \end{cases} \tag{19}$$

6.3.4 模型回归预测结果对比与分析

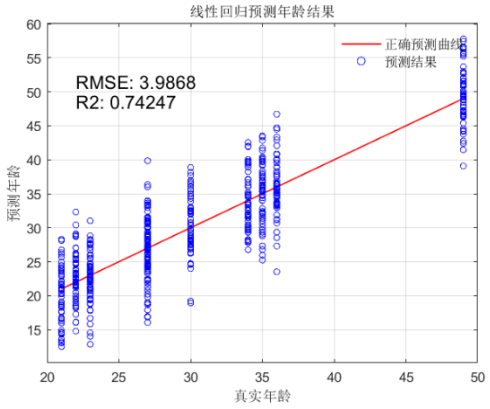
通过建立 SVR 模型和多元线性回归模型，进一步判断实验人员的活动状态数据是否与实验人员身体数据，即年龄、身高和体重有关。由于，实验人员 1-3 的活动状态标签未给出，为避免第一问中分类器存在的分类错误影响回归模型，因此使用实验人员 4-13 的活动状态数据建模。首先，按照问题二中的 Relief-F 特征筛选方法筛选特征，将年龄、身高和体重 3 类身体数据作为回归模型的自变量，将活动状态数据的特征作为回归模型的因变量。选取高斯径向基函数核作为 SVR 模型的核函数，使用最小二乘法估计多元线性回归模型的最优参数。采用 k 折交叉验证的方法进行模型训练，k 值取 10，在数据集上，进行 10 次划分，每次划分都进行以此训练、评估，对所有划分结果求取平均值作为最终的评价结果。

如图 14 所示为 SVR 模型与多元线性回归模型预测效果对比图，根据实验人员活动状态的特征，分别预测出实验人员的年龄、身高和体重 3 类身体数据。图中横轴为实验人员的真实身体数据，纵轴为模型预测数据，红线表示正确预测曲线，蓝圈表示模型预测结果，蓝圈的位置越靠近红线说明预测得越准确。由于仅有十名实验人员的活动状态数据，因此每类真实的身体数据不会超过 10 种，其中真实年龄共 9 种（两名实验人员年龄相同），真实身高共 8 种（四名实验人员身高分别相同），真实体重共 10 种，真实体重与预测体重呈现图中的条形分布。

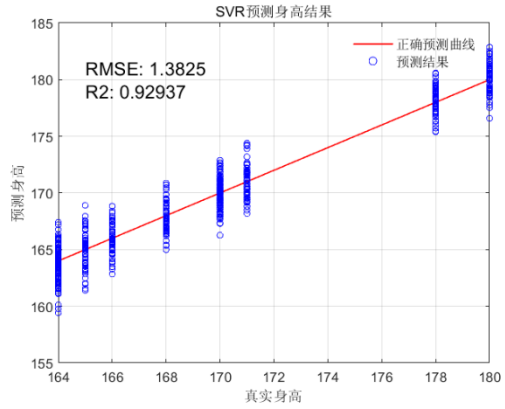
通过建立的两种回归模型，可知实验人员的活动状态数据与实验人员身体数据存在关系，可以使用传感器数据对实验人员进行画像。



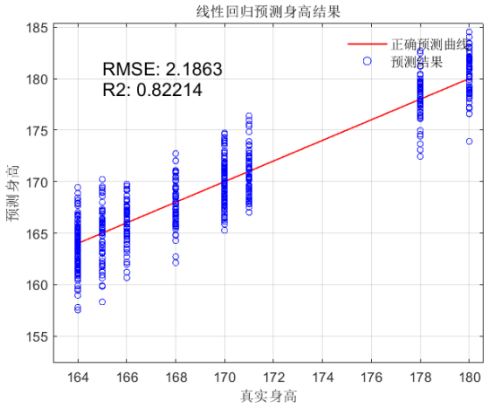
(a)



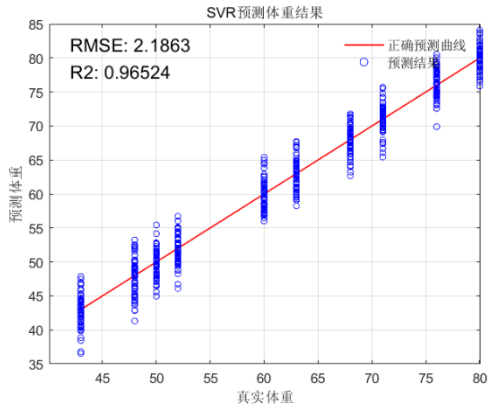
（b）



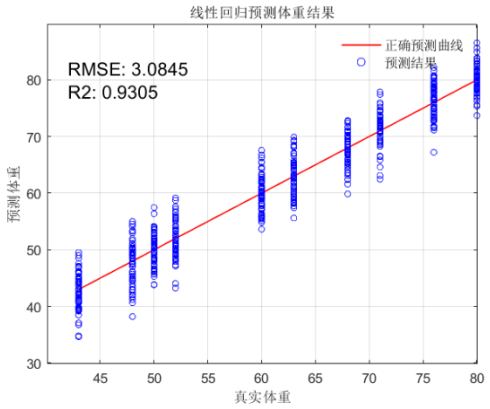
(c)



(d)



(e)



(f)

图14 SVR 模型与多元线性回归模型预测效果对比

如表 9 所示，为 SVR 模型及多元线性回归模型预测的 RMSE 与 R2 值，通过这两种指标，可以看出，两种回归模型对于身高的预测效果最好，对于体重的预测效果次之，对于年龄的预测效果最差。这说明身高对活动状态的影响要大于体重和年龄对活动状态的影响，年龄对活动状态的影响相对来说较弱。

表9 SVR 模型与多元线性回归模型预测 RMSE 与 R^2

算法	身体数据	RMSE	R^2
SVR	年龄	2.6091	0.88974
	身高	1.3895	0.92937
	体重	2.1863	0.96524
线性回归	年龄	3.9868	0.74247
	身高	2.1863	0.82214
	体重	3.0845	0. 93050

对比两种模型发现，SVR 模型的预测效果要优于多元线性回归模型，这是由于 SVR 模型适用于非线性和复杂数据模式的建模，因此本文使用 SVR 模型使用问题 3 的五组 Unknow 活动状态数据进行实验人员画像，判别结果如表 10 所示。

表10 问题 3 预测与判别结果（括号内为预测值）

活动类型	判别结果	年龄	身高	体重
Unknow1	实验人员 10	49（48）	166（168）cm	68（70）kg
Unknow2	实验人员 7	35（37）	170（168）cm	63（61）kg
Unknow3	实验人员 6	27（30）	164（165）cm	50（47）kg
Unknow4	实验人员 9	22（25）	180（177）cm	76（72）kg
Unknow5	实验人员 13	36（38）	170（168）cm	80（78）kg

7 模型评价与推广

7.1 模型的优点

(1) 模型充分结合实际, 简化了数据分布，仪器误差等条件, 考虑了诸多重要因素得到合理的模型, 如: 重力，误差传播等。 从序列信号的特征出发，而非系统经验地人工设计每个动作的单独特征，得到的模型贴合实际,并且泛化性能良好，具有较高的应用价值。仅更改原始数据，重新进行特征提取和训练就可以推广到其他场景下的运动姿态识别；

(2) 模型运用层次分析和对比试验的思想, 对问题单独分析建立模型，有助于问题解决的专业化和准确度的提升；同时，为了探究更优的算法，在选择模型时，本文设计了对比实验，并科学的多次实验以消除偶然性。相比直接实现的模型，本文设计的模型更精确可靠；

(3) 本文使用的算法具有训练时间短，预测时间快的优点, 这首先是我们特征向量短的优点，其次来源于我们选择的算法并不复杂，开销小。本文设计的模型可以较为方便的移植至边缘平台实现应用；

(4) 针对单一特征筛选和降维的方法过于依赖数据的某一方面特性导致特征有偏的问题，本文创新性的提出了多信息融合的特征筛选算法。通过对不同降维方式的合理运用，在保证特征的信息空间足以满足后续算法的要求的同时，最大限度了减少了冗余信息的残留，降低需要的特征数。

7.2 模型的不足

(1) 实际应用中, 陀螺仪和加速度传感记录的信号会明显的受到温度，光照等环境因素的影响，从而变成非平稳随机信号。我们的模型对其进行了假设简化，只适用于平稳随机信号的处理过程。因此，在适用范围上做出了牺牲;

(2) 本文提出的模型对于现有条件使用效果较好, 但由于特征的提取上不够充分，导致部分类别的识别率相对较低；

7.3 模型的改进

进一步考虑小波域的影响，小波变换能有效的兼顾时域和频域信息，从而，针对加滑窗分段的方法不便提取的特征可以更科学有效的提取。

7.4 模型推广

本文建立的基于惯性传感器的活动状态判别模型，对于交通工具姿态识别与智能驾驶具有参考作用。

参考文献

[1] 李佳珍.基于传感器数据和深度学习的日常活动识别方法研究[D].哈尔滨:哈尔滨工业大学,2018.

[2] 夏资厚，刘吉晓，刘均益，郭士杰.基于 RFID 的人体姿态识别方法研究[J].传感器与微系统,2024,第 43 卷(1): 36-39，43

[3] Hsu Y, Wang J, Chang C. A wearable inertial pedestrian navigation system with quaternion-based extended Kalman filter for pedestrian localization[J]. IEEE Sensors Journal, 2017, 17(10): 3193-3206.

[4] 冯国双. 白话统计[M]. 电子工业出版社, 2018.

[5] 张良均. Python 数据分析与挖掘实战[M]. 机械工业出版社, 2016.

[6] 茆诗松, 程依明, 濮晓龙, 等. 概率论与数理统计教程第二版[M]. 北京: 高等教育出版社, 2011

[7]《运筹学》教材编写组. 运筹学.第 4 版[M]. 清华大学出版社, 2012.

[8] 周志华. 机器学习[M]. 清华大学出版社, 2016.

[9] Rachel Schutt, Cathy O'Neil. 数据科学实战[M]. 人民邮电出版社, 2015.

[10] 姜启源, 谢金星, 叶俊. 数学模型.第 4 版[M]. 高等教育出版社, 2011.

[11] 韩中庚. 数学建模方法及其应用-第 2 版[M]. 高等教育出版社, 2009.

[12] 陈永浩.遮挡条件下人体局部运动特征多路径识别方法[J].电子设计工程 2019,27(15):175-178.

[13] Furui Liu;Xiyan Liu.Unsupervised Feature Selection for Multi-cluster Data via Smooth Distributed Score[J].Emerging Intelligent Computing Technology and Applications,2012,Vol.304: 74-79

[14] Ryan J. Urbanowicz, Melissa Meeker, et al,Relief-based feature selection: Introduction and review,Journal of Biomedical Informatics,Volume 85,2018,Pages 189-203.

[15] Zhang Y, Yang A, Xiong C, et al. Feature selection using data envelopment analysis[J]. Knowledge-Based Systems, 2014, 64: 70-80.

[16] 杨晨晨,马春梅,朱金奇.基于智能手机跌倒行为识别算法研究[J].计算机工程.2019,45(02):178-183.

[17] 冯超，徐鲲鹏，陈黎飞.符号序列的 LDA 主题特征表示方法[J].山东大学学报(工学版),2020,第 50 卷(2): 60-65

附　录

附录 A:支撑材料

支撑材料列表

序号	文件名	材料说明
1	分类模型准确度结构.xlsx	使用问题一中分类模型分类问题二 10 名实验人员的分类结果
2	代码	主要模型代码文件
3	图片	文中生成的图片
4	数据	文中使用到的数据

附录 B: 关键数据 1 问题一分类结果

分类	Person1	Person2	Person3
第 1 类	SY7,SY33,SY38,SY42,SY58	SY2,SY10,SY12,SY26,SY47	SY13,SY27,SY29,SY51,SY56
第 2 类	SY14,SY17,SY28,SY41,SY47	SY16,SY21,SY29,SY40,SY43	SY9,SY14,SY30,SY37,SY43
第 3 类	SY5,SY8,SY46,SY56,SY59	SY3,SY20,SY38,SY51,SY60	SY8,SY23,SY39,SY42,SY52
第 4 类	SY24,SY25,SY29,SY48,SY53	SY7,SY24,SY33,SY44,SY57	SY11,SY25,SY31,SY44,SY60
第 5 类	SY27,SY31,SY37,SY49,SY52	SY19,SY23,SY30,SY37,SY46	SY19,SY24,SY35,SY36,SY49
第 6 类	SY19,SY26,SY43,SY44,SY50	SY1,SY4,SY48,SY49,SY50	SY10,SY41,SY57,SY58,SY59
第 7 类	SY1,SY2,SY15,SY23,SY36	SY13,SY27,SY28,SY34,SY42	SY4,SY5,SY18,SY22,SY53
第 8 类	SY4,SY13,SY18,SY39,SY57	SY5,SY11,SY22,SY54,SY56	SY6,SY17,SY21,SY33,SY38
第 9 类	SY9,SY10,SY12,SY20,SY24	SY6,SY15,SY25,SY35,SY58	SY2,SY12,SY34,SY47,SY48
第 10 类	SY22,SY32,SY34,SY35,SY40	SY31,SY32,SY39,SY52,SY59	SY3,SY32,SY50,SY54,SY55
第 11 类	SY3,SY6,SY11,SY16,SY55	SY8,SY9,SY36,SY41,SY55	SY1,SY7,SY26,SY40,SY45

第 12 类	SY30,SY45,SY51,SY54,SY60	SY14,SY17,SY18,SY45,SY53	SY15,SY16,SY20,SY28,SY46

附录 C：关键数据 2 问题二（2）判别结果

类型	状态	类型	状态	类型	状态
SY1	步行下楼	SY11	站立	SY21	向前跑
SY2	向前走	SY12	跳跃	SY22	向左走
SY3	跳跃	SY13	步行上楼	SY23	坐下
SY4	乘电梯上	SY14	向右走	SY24	步行下楼
SY5	跳跃	SY15	步行上楼	SY25	向左走
SY6	躺下	SY16	向前走	SY26	站立
SY7	向左走	SY17	步行上楼	SY27	坐下
SY8	向前跑	SY18	步行下楼	SY28	步行下楼
SY9	跳跃	SY19	坐下	SY29	向前跑
SY10	躺下	SY20	坐下	SY30	步行下楼

附录 D：关键数据 3 问题三回归预测结果

活动类型	判别结果	年龄	身高	体重
Unknow1	实验人员 10	49（48）	166（168）cm	68（70）kg
Unknow2	实验人员 7	35（37）	170（168）cm	63（61）kg
Unknow3	实验人员 6	27（30）	164（165）cm	50（47）kg
Unknow4	实验人员 9	22（25）	180（177）cm	76（72）kg
Unknow5	实验人员 13	36（38）	170（168）cm	80（78）kg

附录 E：主要程序/关键代码

代码环境	操作系统: Windows11 编程语言: Python 3.7.1、Matlab2022a 编辑器: Vscode、Matlab 代码详见：支撑材料

代码清单 1 数据处理模块

```
1. function [feature,filter_data]=get_feature(path,windowSize)
2.     [num, ~, ~] = xlsread(path);
3.     acc_x = num(:,1);acc_y = num(:,2);acc_z = num(:,3);
4. %     rox = num(:,4);roy = num(:,5);roz = num(:,6);
```

```
5.      filter_data = zeros(size(num)) ;
6.      dim = 6;%传感器组数
7.      fs = 100 % 采样率
8.      L = length(acc_x(:,1));
9.      t = linspace(0, L/fs, L);  % 时间范围从 0 到 13，生成 1300 个点,100hz
10.
11.     for i=1:dim
12.     %     % 绘制原始数据曲线 figure; sub-
    plot(2, 1, 1); plot(t, num(:,i), '-b');
13.     %     title('原始数据:x 方向加速度'); xlabel('时间'); ylabel('幅值
    '); grid on;
14.
15.         % 设计并应用低通滤波器（使用移动平均滤波作为示例）
16.         %windowSize = 65;  % 滤波窗口大小
17.         b = (1/windowSize)*ones(1, windowSize);  % 滤波器系数
18.         a = 1;  % 分母系数（在移动平均滤波中通常为 1）
19.
20.         % 应用滤波器
21.         i_filtered = filter(b, a, num(:,i));
22.         m = 1;
23.         if i<=3
24.             i_filtered = i_filtered./6;
25.         else
26.             i_filtered = i_filtered./500;
27.         end
28.     %     % 绘制滤波后的数据曲线 subplot(2, 1, 2); plot(t, y_filtered, '-
    r');
29.     %     title('低通滤波后数据：x 方向加速度'); xlabel('时间'); ylabel('幅
    值'); grid on;
30.
31.         % 可选：对比原始数据和滤波后的数据 figure; plot(t, acc_x, '-
    b', t, y_filtered,
32.         % '-r'); legend('原始数据', '低通滤波后数据'); title('原始数据与低
    通滤波后数据对比');
33.         % xlabel('时间'); ylabel('幅值'); grid on;
34.
35.         % 计算频率轴
36.         f=(0:L-1)/fs;
37.         time(i,:)=time_statistical_compute(i_filtered(:,1));%%%%%%%多维
    矩阵计算时域特征矩阵
38.         freqi = abs(2*fft(i_filtered)/L);
39.         fre(i,:)=fre_statistical_compute(f',freqi(:,1),L);%%%%%%%多维矩
    阵计算频域特征矩阵
40.         feature(i,:)=[time(i,:),fre(i,:)];
41.         filter_data(:,i) = i_filtered;
42.
43.     %  %maxv 峰峰值点  maxl：峰峰值点对应的位置
44.     % [maxv,maxl]=findpeaks(freqi,'minpeakdistance',200,'Thresh-
    old', TH);
45.     % %设定两峰值间的最小间隔数,200 if freq(maxl)>3       %设定峰值的最小高
    度 3
```

```
    %     figure, plot(maxl,maxv,'*','color','R');                %绘制
最大值点
    % else
    %     figure,
    % end
        % 绘制单边幅值谱
    %     figure; f = fs*(0:(L/2))/L; plot(f, 2*abs(freqi(1:L/2+1)));
    %     axis([0 10 0 1]) title('单边幅值谱'); xlabel('频率 (Hz)');
    %     ylabel('|Y(f)|');
    end
    temp = filter_data.^2;
    tol = [mean(temp(:,1:3),2),mean(temp(:,4:6),2)];
    filter_data(:,7:8) = tol;

    time(7,:) = time_statistical_compute(tol(:,1));
    freqi = abs(2*fft(tol(:,1))/L);
    fre(7,:) = fre_statistical_compute(f',freqi(:,1),L);
    feature(7,:) = [time(7,:),fre(7,:)];
    time(8,:) = time_statistical_compute(tol(:,2));
    freqi = abs(2*fft(tol(:,2))/L);
    fre(8,:) = fre_statistical_compute(f',freqi(:,1),L);
    feature(8,:) = [time(8,:),fre(8,:)];
      feature = reshape(feature,1,[]);
```

代码清单 2 分类模块

```
function [ res, record] = FunK_meanPolyD(data,k )
    j = 1;
    % 下面是预分配一些空间
    % seedX 和 seedY 中存放着所有种子
    [h w] = size(data);
    cnt = w; % 输入元素的数量
    cntOfDimension = h; % d 中存放着本次处理数据的维度
    %seed 中存放种子，每一行代表种子所在的一个维度，每一列是一个种子向量
    seed = zeros(cntOfDimension,k);
    oldSeed = zeros(cntOfDimension,k);
    % 结果矩阵 res 中，数据存放规则：
    %   以 d 行为一个单位，总共 k 个 d 行
    %   第一个 d 行数据存放着第一类元素集合,其他同理
    res = zeros(k*cntOfDimension,cnt);
    % 用来记录 resX 中每一行有效元素的个数
    record = zeros(1,k);
    r = 0;
    for i = 1:k % 产生 k 个随机种子，注意： 随机种子是来自元素集合
        t = round(rand()*cnt);
        % 为保证种子不重叠
        if i > 1 && t == r
            i = i - 1;
            continue;
```

```
        end

        seed(:,i) = data(:,t);
        r = t;
    end

    while 1
        record(:) = 0; % 重置为零
        res(:) = 0;
        for i = 1:cnt % 对所有元素遍历
            % 下面是判断本次元素应该归为哪一类，这里我们是根据欧几里得距离进行类别判定
            % k-mean 算法认为元素应该归为距离最近的种子代表的类
            distanceMin = 1; % distanceMin 中存放着最短欧几里得距离的种子点的下标
            for j = 2:k
                % 计算高维度的欧几里得距离
                a = 0;
                b = 0;
                for row = 1:cntOfDimension
                    a = a + power(data(row,i)-seed(row,distanceMin),2);
                    b = b + power(data(row,i)-seed(row,j),2);
                end
                if a > b
                    distanceMin = j;
                end
            end
            % 将本次元素点进行类别归并
            row = (distanceMin-1)*cntOfDimension + 1;
            res(row:row+cntOfDimension-1,record(distanceMin)+1) = data(:,i);
            record(distanceMin) = record(distanceMin)+1;
        end
        %record
        oldSeed = seed;
        % 移动种子至其类中心
        for col = 1:k
            if record(col) == 0
                continue;
            end
            % 计算新的种子位置
            row = (col-1)*cntOfDimension + 1;
            seed(:,col) = sum(res(row:row+cntOfDimension-1,:),2)/record(col);
        end
        % 如果本次得到的种子和上次的种子一致，则认为分类完毕。
        if mean(seed == oldSeed) == 1
            break;
        end
    end

```

```
    maxPos = max(record);
    res = res(:,1:maxPos);
end
```

代码清单 3 PCA 模块

```
clc;
clear ;
close all;
load('feature_total.mat')
%%
feature0 = feature_total;
%%
%standard
% feature = feature(1:35,2:11);
% feature1 = [];
% [a,b]=size(feature);
% for i =1:a
%     for j=1:b
%         feature1=[feature1
%             feature(i,j)];
%     end
% end
% feature0=[];
% for i = 1:a*b
%     feature0 = [feature0
%         feature1{i,1}];
% end
%X = feature0;
%%
X=[];
[q,p]=size(feature0);
for i = 1:p
    W=feature0(:,i)';
    %Q=normalize(W);
    Q=mapminmax(W,0,1);
    X = [X Q'];
end
%%
% feature0 = [];
% for i=1:10
%     j=60*(i-1);
%     feature0 = [feature0;feature(1+j:15+j,:)];
% end
%%
%  [M,N] = size(feature0);
% %%分训练集和测试集
% train_data = feature0(1:9*M/10,1:N-1);
```

```
% train_label = feature0(1:9*M/10,N);
% test_data = feature0(9*M/10+1:M,1:N-1);
% test_label = feature0(9*M/10+1:M/10,N);
%  X=train_data;
 %%
% [m,n]=size(X);
% for i=1:n
%     A(1,i)=norm(X(:,i));
% end
% A=repmat(A,m,1);
% train_data = X./A;
%%
%train_data = normalize(X);
train_data = X;
%%
%%PCA 主成分分析
p = 3; %特征维度
[coeff,score,latent,tsquared,explained,mu] = pca(train_data);
cumsum(latent)/sum(latent);%累积贡献值
tran_matrix = coeff(:,1:p);

%%降维后的训练集
train_data0 = bsxfun(@minus,train_data,mean(train_data,1));
low_train_data = train_data0 * tran_matrix;

%%降维后的测试集
% test_data0 = bsxfun(@minus,test_data,mean(test_data,1));
% low_test_data = test_data0 * tran_matrix;
% figure(1);
    % scat-
ter3(low_train_data(:,1),low_train_data(:,2),low_train_data(:,3),'fille
d');
```

代码清单 4 GMM 模块

```
    import matplotlib.pyplot as plt
import pandas as pd
data = pd.read_csv('data1.csv')
#print(data)
plt.figure(figsize=(7,7))
plt.scatter(data['p1'],data['p2'])
plt.xlabel('p1')
plt.ylabel('p2')

plt.title('Data Distribution')
plt.show(block=True)

from sklearn.cluster import KMeans
```

```
kmeans = KMeans(init='k-means++',n_clusters=3)
kmeans.fit(data)
print('2222')
print(kmeans.inertia_)
print('2222')
#predictions from kmeans
pred = kmeans.predict(data)
frame = pd.DataFrame(data)
frame['cluster'] = pred
frame.columns = ['p1', 'p2', 'cluster']

#plotting results
color=['blue','green','cyan']
for k in range(0,3):
    data = frame[frame["cluster"]==k]
    plt.scatter(data["p1"],data["p2"],c=color[k])
plt.show()
print(frame)

import pandas as pd
data = pd.read_csv('data1.csv')

# training gaussian mixture model
from sklearn.mixture import GaussianMixture
gmm = GaussianMixture(n_components=3)
gmm.fit(data)

#predictions from gmm
labels = gmm.predict(data)
frame = pd.DataFrame(data)
frame['cluster'] = labels
frame.columns = ['p1', 'p2', 'cluster']

color=['blue','green','cyan']
for k in range(0,3):
    data = frame[frame["cluster"]==k]
    plt.scatter(data["p1"],data["p2"],c=color[k])
plt.show()
  print(frame)
```

代码清单 5 UFS 模块

```
import numpy as np
import math
from  sklearn.linear_model import LassoLarsCV
import scipy

def UFS(data, K, d):
    k = int(max(0.2*data.shape[0],10))#近邻设置的数目
```

```
    data = np.array(data)
    M = data.shape[1]#数据的维数
    N = data.shape[0]#数据的样本数目
    data = data.transpose()#将矩阵转换成列形式,与文中的形式保持一致
    #寻找数据的样本的K 近邻
    dist = np.zeros((N, N))
    delta = 2#原文中热核函数的参数
    W = np.zeros((N, N))#邻接矩阵的权值
    for i in range(N):#计算样本之间的距离
        for j in range(N):
            dist[i,j] = np.linalg.norm(data[:,i]-data[:,j], ord=2)
    D = np.zeros((N, N))#初始化度矩阵
    for i in range(N):#找出每个样本之间的k 近邻
        neigbors = np.argsort(dist[i,:], axis=0)
        neigbors = neigbors[1:k+1]#只选取前k 个样本,因为第0 个最短矩阵是本身与本身的距离,其距离为0
        for j in neigbors:
            W[i,j] = math.exp(-math.pow(np.linalg.norm(data[:,i]-data[:,j]),2)/delta)#计算权值矩阵中的元素
            W[j,i] = W[i,j]
        D[i,i] = sum(W[i,:])#确定度矩阵中的对角元素
    L = D - W #计算拉普拉斯矩阵
    feature_values, vectors = scipy.linalg.eig(L,D)#求取Eq.(1)中的广义特征值与特征向量
    seq = np.argsort(feature_values)#对特征值进行排序
    seq = seq[1:K+1]#选取从次小后的K 个特征值
    Y = vectors[:,seq]#获取特征向量
    #Y = np.real(Y)
    # 采用最小角回归来气球节a 的参数
    score = np.zeros((1,M))#记录每个特征的得分
    print('......')
    print(score)
    print('......')
    model = LassoLarsCV()#训练一个模型
    for i in range(K):
        model.fit(data.transpose(),Y[:,i])
        a = model.coef_#获取线性回归模型的系数
        score[0,i] = max(a)
        print('......')
        print(score)
        print('......')
    seq = np.argsort(-score)#对得分由大到小排序
    seq = seq[0,0:d]#选取前d 个最大的得分所对应的特征序号
    data = data.transpose()
    data = data[:,seq]#获得最终的结果
    return data,seq
```

代码清单 6　回归模型及 SVM

使用 Matlab 工具箱