Hidden Markov Model 用于人体活动识别

1. 公开数据集

数据集：Human Activity Recognition Using Smartphones（UCI HAR）**描述**：由 30 位被试佩戴腰部智能手机的加速度计与陀螺仪采集，共 6 类活动（WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING）。数据经预处理并分为训练测试集，适合时间序列建模与序列标注实验。**数据链接**：<https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones>**数据特点**：训练集 7352 样本，测试集 2947 样本，每个样本为 561 维特征向量。为降低维度并提高模型稳定性，使用 PCA 将特征降至 10 维（解释方差 70.8%）。按被试分组形成序列，训练集 21 条序列，测试集 9 条序列。

2. Hidden Markov Model 形式与参数

采用**离散时间、连续观测的 Gaussian-HMM**，每个隐藏状态的观测服从多元高斯分布：**隐状态数**：K=6（对应 6 类活动）**初始概率向量**：$\boldsymbol{\pi} \in \mathbb{R}^K$，满足 $\sum_i \pi_i = 1$ **状态转移矩阵**：$A \in \mathbb{R}^{K \times K}$，其中 $A_{ij} = P(z_{t+1} = j \mid z_t = i)$（每行和为 1）**发射分布**：每个状态 k 对应多元高斯 $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

需训练的参数（通过 EM/Baum-Welch 算法）：$\theta = \{\boldsymbol{\pi}, A, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K\}$

实现细节：使用 log-space 进行 forward/backward 计算防止数值下溢；对协方差矩阵添加正则项 ϵI 防止奇异；观测维度通过 PCA 降维至 10 维。

3. Baum-Welch (EM) 算法伪代码

输入：观测序列集合 $\{y_{1:T_s}^{(s)}\}$（$s = 1, \ldots, S$），状态数 K，初始化参数 $\theta^{(0)} = (\pi, A, \{\mu_k, \Sigma_k\})$，最大迭代 M，收敛阈值 tol

Code block

```
for iter = 1..M:
    // E-step: 对每条序列 s 执行 forward-backward
    for each sequence s:
        compute log α_t(i) via forward recursion (log-space)
        compute log β_t(i) via backward recursion
        for t=1..T_s:
            γ_t^s(k) ∝ exp(log α_t(k) + log β_t(k))  // 归一化
        for t=1..T_s-1:
            ξ_t^s(i,j) ∝ exp(log α_t(i) + log A[i,j] +
                                log b_j(y_{t+1}) + log β_{t+1}(j))
    // M-step: 汇总期望统计并更新参数
    π_k ← (1/S) Σ_s γ_1^s(k)  // 归一化
```

```
        A[i,j] ← (Σ_s Σ_{t=1}^{T_s-1} ξ_t^s(i,j)) / (Σ_s Σ_{t=1}^{T_s-1} γ_t^s(i))
        μ_k ← (Σ_s Σ_t γ_t^s(k) y_t^s) / (Σ_s Σ_t γ_t^s(k))
        Σ_k ← (Σ_s Σ_t γ_t^s(k) (y_t^s - μ_k)(y_t^s - μ_k)^T) /
              (Σ_s Σ_t γ_t^s(k)) + ε I
        // 计算对数似然 L^{(iter)}
        if |L^{(iter)} - L^{(iter-1)}| < tol: break
    输出：估计参数 θ
```

理论保证：每次迭代不减小观测数据对数似然（Jensen 不等式），算法收敛到局部最优。

4. Inference 问题与测试集表现

4.1 Inference 方法

Filtering（在线后验）：计算 $P(z_t \mid y_{1:t})$，使用 forward recursion：$\alpha_t(j) = b_j(y_t) \sum_i \alpha_{t-1}(i) A_{ij}$ 预测状态取 $\arg\max_j P(z_t \mid y_{1:t})$ **Smoothing（全序列后验）**：计算 $P(z_t \mid y_{1:T})$，使用 forward-backward: $\gamma_t(j) \propto \alpha_t(j)\beta_t(j)$ **Prediction（一步预测）**：$P(z_{t+1} \mid y_{1:t}) = P(z_t \mid y_{1:t})A$ 取 argmax 预测下一时刻状态, **MAP estimation（Viterbi）**：使用动态规划找到 $\arg\max_{z_{1:T}} P(z_{1:T} \mid y_{1:T})$

4.2 测试集结果

在 UCI HAR 测试集上的评估结果（平均准确率）：**评估方法**：由于隐状态标号存在置换不确定性，使用 Hungarian 算法根据估计均值与类别质心的最近匹配进行状态重标号，然后计算逐时刻准确率。

Filtering	Smoothing	Viterbi (MAP)	One-step Prediction
0.7050	0.7126	0.7137	0.6821

4.3 结果讨论

1. **Smoothing 优于 Filtering**（0.7126 > 0.7050）：Smoothing 使用了未来信息，因此后验估计更准确。
2. **Viterbi 表现最佳**（0.7137）：Viterbi 算法寻找全局最优路径，在序列标注任务中通常优于逐时刻最优决策。这与理论预期一致。
3. **One-step Prediction 准确率较低**（0.6821）：一步预测仅依赖转移概率矩阵和当前滤波分布，若活动间转移随机性较高，预测难度较大。
4. **模型适用性**：在 561 维原始特征上直接使用单高斯 emission 难以充分拟合复杂观测分布。通过 PCA 降维至 10 维后，Gaussian-HMM 能够达到约 71% 的准确率，说明降维策略有效。进一步改进可考虑：使用 GMM-HMM（每个状态的发射用混合高斯）、或通过交叉验证选择最优状态数 K、或使用 BIC/AIC 进行模型选择。