姓名：麻超
学号：201300066

一. (20 points) **贝叶斯决策论**

教材 7.1 节介绍了贝叶斯决策论, 它是一种解决统计决策问题的通用准则. 考虑一个带有“拒绝”选项的 N 分类问题, 给定一个样例, 分类器可以选择预测这个样例的标记, 也可以选择拒绝判断并将样例交给人类专家处理. 设类别标记的集合为 $\mathcal{Y} = \{c_1, c_2, \ldots, c_N\}$, λ_{ij} 是将一个真实标记为 c_i 的样例误分类为 c_j 所产生的损失, 而人类专家处理一个样例需要额外 λ_h 费用. 假设后验概率 $P(c \mid \boldsymbol{x})$ 已知, 且 $\lambda_{ij} \geq 0$, $\lambda_h \geq 0$. 请思考下列问题:

1. 基于期望风险最小化原则, 写出此时贝叶斯最优分类器 $h^\star(\boldsymbol{x})$ 的表达式;
2. 人类专家的判断成本 λ_h 取何值时, 分类器 $h^\star$ 将一直拒绝分类? 当 λ_h 取何值时, 分类器 $h^\star$ 不会拒绝分类任何样例?
3. 考虑一个具体的二分类问题, 其损失矩阵为

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{1}$$

且人类专家处理一个样例的代价为 $\lambda_h = 0.3$. 对于一个样例 $\boldsymbol{x}$, 设 $p_1 = P(c_1 \mid \boldsymbol{x})$, 证明存在 $\theta_1, \theta_2 \in [0, 1]$, 使得贝叶斯最优决策恰好为: 当 $p_1 < \theta_1$ 时, 预测为第二类, 当 $\theta_1 \leq p_1 \leq \theta_2$ 时, 拒绝预测, 当 $\theta_2 < p_1$ 时, 预测为第一类.

解:

1. 可以定义判定准则 $h : \mathcal{X} \to \mathcal{Y}$ 的条件风险为:

$$R(h(\boldsymbol{x})|\boldsymbol{x}) = \min\ (\min_{c_i \in \mathcal{Y}}\ R(c_i|x), \lambda_h)$$

故应当寻找一个 h 以最小化期望风险:

$$R(h) = \mathbb{E}_x[R(h(\boldsymbol{x})|\boldsymbol{x})]$$

所以最小化条件风险即可得到判定准则:

$$h^*(\boldsymbol{x}) = \arg\min_{c \in \mathcal{Y}}\ R(c|x)$$

2. 当满足条件 $\lambda_h \leq \min_{\boldsymbol{x} \in \mathcal{X}}(\min_{c \in \mathcal{Y}} P(c|x))$ 时, 分类器会一直拒绝分类.

当满足条件 $\lambda_h > \max_{\boldsymbol{x} \in \mathcal{X}}(\min_{c \in \mathcal{Y}} P(c|x))$ 时, 分类器不会拒绝分类.

3. 首先计算各个类别的条件风险值:

$$R(c_1|\boldsymbol{x}) = P(c_2|\boldsymbol{x}) = 1 - p_1, R(c_2|\boldsymbol{x}) = P(c_1|\boldsymbol{x}) = p_1$$
$$\lambda_h = 0.3$$

当 $p_1 < \theta_1$ 时, 预测为第二类, 则有:

$$p_1 \leq 1 - p_1$$
$$p_1 \leq 0.3$$

得到 $\theta_1 \leq 0.3$.

当 $\theta_1 < p_1 < \theta_2$ 时, 有:

$$1 - p_1 \geq 0.3$$
$$p_1 \geq 0.3$$

得到 $\theta_1 \geq 0.3, \theta_2 \leq 0.7$

当 $p_1 > \theta_2$ 时, 有:

$$1 - p_1 \leq p_1$$
$$1 - p_1 \leq 0.3$$

得到 $\theta_2 \geq 0.7$

所以存在 $\theta_1 = 0.3, \theta_2 = 0.7$ 满足题目要求.

二. (20 points) **极大似然估计**

教材 7.2 节介绍了极大似然估计方法用于确定概率模型的参数. 其基本思想为: 概率模型的参数应当使得当前观测到的样本是最有可能被观测到的, 即当前数据的似然最大. 本题通过抛硬币的例子理解极大似然估计的核心思想.

1. 现有一枚硬币, 抛掷这枚硬币后它可能正面向上也可能反面向上. 我们已经独立重复地抛掷了这枚硬币 99 次, 均为正面向上. 现在, 请使用极大似然估计来求解第 100 次抛掷这枚硬币时其正面向上的概率;
2. 仍然考虑上一问的问题. 但现在, 有一位抛硬币的专家仔细观察了这枚硬币, 发现该硬币质地十分均匀, 并猜测这枚硬币 “肯定有 50% 的概率正面向上” . 如果同时考虑已经观测到的数据和专家的见解, 第 100 次抛掷这枚硬币时, 其正面向上的概率为多少?
3. 若同时考虑专家先验和实验数据来对硬币正面朝上的概率做估计. 设这枚硬币正面朝上的概率为 θ, 某抛硬币专家主观认为 $\theta \sim \mathcal{N}(\frac{1}{2}, \frac{1}{900})$, 即 θ 服从均值为 $\frac{1}{2}$, 方差为 $\frac{1}{900}$ 的高斯分布. 另一方面, 我们独立重复地抛掷了这枚硬币 400 次, 记第 i 次的结果为 x_i, 若 $x_i = 1$ 则表示硬币正面朝上, 若 $x_i = 0$ 则表示硬币反面朝上. 经统计, 其中有 100 次正面向上, 有 300 次反面向上. 现在, 基于专家先验和观测到的数据 $\boldsymbol{x} = \{x_1, x_2, \ldots, x_{400}\}$, 对参数 θ 分别做极大似然估计和最大后验估计;
4. 如何理解上一小问中极大似然估计的结果和最大后验估计的结果?

解:

1. 设样本满足参数为 θ 的 0-1 分布, 则其似然为:

$$L(D|\theta) = \prod_{x \in D} P(x|\theta) = \theta^{99}.$$

其对数似然为:

$$\ln L(D|\theta) = 99 \ln \theta.$$

该对数似然函数为关于 θ 的增函数, 故 θ 的极大似然估计为 $\hat{\theta} = 1$...

所以第 100 次掷硬币正面朝上概率的极大似然估计为 1.

2. 无法使用最大后验估计. 根据其说法, 假设前 99 次都正面朝上为事件 A, 事件 B 为掷一次硬币结果为正面朝上, 则有 $P(B) = 0.5$. 可得:

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)} = 0.5.$$

所以第 100 次正面朝上概率为 0.5

3. 极大似然估计:

$$\hat{D|\theta} = \arg\max_\theta \theta^{100}(1-\theta)^{300}$$
$$L(D|\theta) = \theta^{100}(1-\theta)^{300}$$
$$LL(D|\theta) = 100\ln\theta + 300\ln(1-\theta)$$

对对数似然函数中的 θ 求偏导可得:

$$\frac{\partial LL(\theta)}{\partial\theta} = 0$$

可得 $\hat{\theta} = \frac{1}{4}$

最大后验估计:

可以得到后验函数与指数后验函数如下:

$$Pos(\theta|D) = \theta^{100}(1-\theta)^{300}\frac{1}{\frac{1}{30}\sqrt{2\pi}}e^{-\frac{(\theta-\frac{1}{2})^2}{2\cdot\frac{1}{900}}}$$
$$\ln Pos(\theta|D) = 100(\ln\theta + 300\ln(1-\theta)) - 450(\theta-\frac{1}{2})^2 + c$$

对对数后验函数的参数 θ 求偏导:

$$2(\frac{1}{\theta} - \frac{3}{1-\theta}) - 18(\theta - \frac{1}{2}) = 0, 0 \le \theta \le 1.$$

可得 θ 的最大后验估计为 $\hat{\theta} = \frac{1}{3}$

4. 极大似然估计完全依靠已有样本来估计概率，只会受到样本集内容的影响，通过调节未知参数使已有数据出现概率最大.

而最大后验估计考虑了先验概率的影响，因此与极大似然估计结果不同。在极大似然估计的基础上，对参数假设其服从一个先验分布, 根据得到的数据计算参数的后验分布. 不仅关注已有的样本数据, 还会通过调节未知参数使得已有数据出现概率最大.

三. (20 points) **朴素贝叶斯分类器**

朴素贝叶斯算法有很多实际应用, 本题以 sklearn 中的 Iris 数据集为例, 探讨实践中朴素贝叶斯算法的技术细节. 可以通过 sklearn 中的内置函数直接获取 Iris 数据集, 代码如下:

```
def load_data():
    # 以feature, label的形式返回数据集
    feature, label = datasets.load_iris(return_X_y=True)
    print(feature.shape) # (150, 4)
```

```
    print(label.shape) # (150,)
    return feature, label
```

上述代码返回 Iris 数据集的特征和标记, 其中 feature 变量是形状为 $(150, 4)$ 的 numpy 数组, 包含了 150 个样本的 4 维特征, 而 label 变量是形状为 (150) 的 numpy 数组, 包含了 150 个样本的类别标记. Iris 数据集中一共包含 3 类样本, 所以类别标记的取值集合为 $\{0, 1, 2\}$. Iris 数据集是类别平衡的, 每类均包含 50 个样本. 我们进一步将完整的数据集划分为训练集和测试集, 其中训练集样本量占总样本量的 80%, 即 120 个样本, 剩余 30 个样本作为测试样本.

```
feature_train, feature_test, label_train, label_test = \
    train_test_split(feature, label, test_size=0.2, random_state=0)
```

朴素贝叶斯分类器会将一个样例的标记预测为类别后验概率最大的那一类对应的标记, 即:

$$\hat{y} = \arg\max_{y\in\{0,1,2\}} P(y)\prod_{i=1}^{d} P(x_i \mid y)\ . \tag{2}$$

因此, 为了构建一个朴素贝叶斯分类器, 我们需要在训练集上获取所有类别的先验概率 $P(y)$ 以及所有类别所有属性上的类条件概率 $P(x_i \mid y)$.

1. 请检查训练集上的类别分布情况, 并基于多项分布假设对 $P(y)$ 做极大似然估计;
2. 在 Iris 数据集中, 每个样例 $\boldsymbol{x}$ 都包含 4 维实数特征, 分别记作 x_1, x_2, x_3 和 x_4. 为了计算类条件概率 $P(x_i \mid y)$, 首先需要对 $P(x_i \mid y)$ 的概率形式做出假设. 在本小问中, 我们假设每一维特征在给定类别标记时是独立的（朴素贝叶斯的基本假设）, 并假设它们服从高斯分布. 试基于 sklearn 中的 GaussianNB 类构建分类器, 并在测试集上测试性能;
3. 在 GaussianNB 类中手动指定类别先验为三个类上的均匀分布, 再次测试模型性能;
4. 在朴素贝叶斯模型中, 对类条件概率的形式做出正确的假设也很重要. 请检查每个类别下特征的数值分布, 并讨论该如何选定类条件概率的形式.

解:

1. 对训练集中的样本进行分析, 可以得到其中第 0 类样本数量为 39, 第一类为 37, 第二类为 44. 故可以对该样本进行分析. 令 $p_0 = P(y=0), p_1 = P(y=1), p_2 = P(y=2)$, 为了让当前样本出现的概率最大, 故可以得到如下优化问题:

$$\begin{aligned}\max_{p_0,p_1,p_2}\ & \frac{120!}{n_0!n_1!n_2!}p_0^{n_0}p_1^{n_1}p_2^{n_2}\\ s.t.\ & p_0+p_1+p_2=1\\ & 0\le p_0,p_1,p_2\le 1.\end{aligned}$$

故其对数似然可以写作如下形式:

$$\ln LL = n_0\ln p_0 + n_1\ln p_1 + n_2\ln(1-p_0-p_1)$$

对 p_0, p_1 求偏导:

$$\begin{aligned}\frac{\partial \ln LL}{\partial p_0} &= \frac{n_0}{p_0} - \frac{n_2}{1-p_0-p_2} = 0\\ \frac{\partial \ln LL}{\partial p_1} &= \frac{n_1}{p_1} - \frac{n_2}{1-p_0-p_2} = 0\end{aligned}$$

根据以上方程可解得:

$$\begin{cases} p_0 &= \frac{n_0}{n} = \frac{39}{120} \\ p_1 &= \frac{n_1}{n} = \frac{37}{120} \\ p_2 &= \frac{n_2}{n} = \frac{44}{120} \end{cases}$$

四. (20 points) **Boosting**

Boosting 算法有序地训练一批弱学习器进行集成得到一个强学习器, 核心思想是使用当前学习器 "提升" 已训练弱学习器的能力. 教材 8.2 节介绍的 AdaBoost 是一种典型的 Boosting 算法, 通过调整数据分布使新学习器重点关注之前学习器分类错误的样本. 教材介绍的 AdaBoost 关注的是二分类问题, 即样本 $\boldsymbol{x}$ 对应的标记 $y(\boldsymbol{x}) \in \{-1, +1\}$. 记第 t 个基学习器及其权重为 h_t 和 α_t, 采用 T 个基学习器加权得到的集成学习器为 $H(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})$. AdaBoost 最小化指数损失: $\ell_{\exp} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-y(\boldsymbol{x})H(\boldsymbol{x})}\right]$.

1. 在 AdaBoost 训练过程中, 记前 t 个弱学习器的集成为 $H_t(\boldsymbol{x}) = \sum_{i=1}^{t} \alpha_i h_i(\boldsymbol{x})$, 该阶段优化目标为:

$$\ell_{\exp,t} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-y(\boldsymbol{x})H_t(\boldsymbol{x})}\right]. \tag{3}$$

   如果记训练数据集的初始分布为 $\mathcal{D}_0 = \mathcal{D}$, 那么第一个弱学习器的训练依赖于数据分布 $\mathcal{D}_0$. AdaBoost 根据第一个弱学习器的训练结果将训练集数据分布调整为 $\mathcal{D}_1$, 然后基于 $\mathcal{D}_1$ 训练第二个弱学习器. 依次类推, 训练完前 $t-1$ 个学习器之后的数据分布变为 $\mathcal{D}_{t-1}$. 根据以上描述并结合 "加性模型"(Additive Model), 请推导 AdaBoost 调整数据分布的具体过程, 即 $\mathcal{D}_t$ 与 $\mathcal{D}_{t-1}$ 的关系;

2. AdaBoost 算法可以拓展到 N 分类问题. 现有一种设计方法, 将样本标记编码为 N 维向量 $\boldsymbol{y}$, 其中目标类别对应位置的值为 1, 其余类别对应位置的值为 $-\frac{1}{N-1}$. 这种编码的一种性质是 $\sum_{n=1}^{N} \boldsymbol{y}_n = 0$, 即所有类别对应位置的值的和为零. 同样地, 学习器的输出为一个 N 维向量, 且约束其输出结果的和为零, 即: $\sum_{n=1}^{N} [h_t(\boldsymbol{x})]_n = 0$. $[h_t(\boldsymbol{x})]_n$ 表示基分类器输出的 N 维向量的第 n 个值. 在这种设计下, 多分类情况下的指数损失为:

$$\ell_{\text{multi-exp}} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{y}_n [H(\boldsymbol{x})]_n}\right] = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-\frac{1}{N}\boldsymbol{y}^\top H(\boldsymbol{x})}\right]. \tag{4}$$

   请分析为何如此设计;

3. 教材 8.2 节已经证明 AdaBoost 在指数损失下得到的决策函数 $\text{sign}(H(\boldsymbol{x}))$ 可以达到贝叶斯最优误差. 仿照教材中的证明, 请从贝叶斯最优误差的角度验证式(4)的合理性.

解:

1. 理想的基学习器 h_t 可以纠正之前的加权分类器 H_{t-1} 的错误, 可以得到:

$$h_t(x) = \arg\max_h \, l_{exp}(H_{t-1} + h_t|\mathcal{D}) = arg\max_h \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\frac{e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}]} f(\boldsymbol{x})h(\boldsymbol{x})]$$

   生成一个新的数据分布, 并使 $h_t(\boldsymbol{x})$ 的生成基于这个分布, 也就是:

$$h_t(\boldsymbol{x}) = arg\max_h \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}[f(\boldsymbol{x})h(\boldsymbol{x})]$$

   令 $\mathcal{D}_t$ 表示一个分布, 则

$$\mathcal{D}_t(\boldsymbol{x}) = \frac{\mathcal{D}(x)e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}]}$$

   则有:

$$\mathcal{D}_t(\boldsymbol{x}) = \mathcal{D}_{t-1}(\boldsymbol{x})e^{-f(\boldsymbol{x})\alpha_{t-1}h_{t-1}(\boldsymbol{x})} \frac{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-2}(\boldsymbol{x})}]}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}]}$$

2. 该损失函数有特定的惩罚机制, 它会惩罚那些和答案偏差较大的结果, 对结果与答案之间相似程度进行判定, 若相似程度较大则会有较小惩罚值, 若不相似则有较大惩罚值, 同时, 该最小化损失函数婚获得一个等晓得贝叶斯分类器, 以达到理论最优解.

3. 即对于一个理想的基学习器对其多元指数损失函数最小化, 也就是:

$$\arg\min_{H(\boldsymbol{x})} mathbbE_{\boldsymbol{x}\sim\mathcal{D}}[e^{-\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{y}_n[H(\boldsymbol{x})]_n}]$$
$$s.t.\ \sum_{n=1}^{N}[H(\boldsymbol{x})]_n = 0$$

原问题的 Lagrange 函数为:

$$exp(-\frac{[H(\boldsymbol{x})_1]}{N-1})Pr(\boldsymbol{y}_1|\boldsymbol{x}) + ... + exp(-\frac{[H(\boldsymbol{x})_n]}{N-1})Pr(\boldsymbol{y}_n|\boldsymbol{x}) - \lambda(\sum_{n=1}^{N}[H(\boldsymbol{x})]_n)$$

对 $[H(\boldsymbol{x})]_n$ 和 λ 分别求导:

$$-\frac{1}{N-1}exp(-\frac{[H(\boldsymbol{x})_1]}{N-1})Pr(\boldsymbol{y}_1|\boldsymbol{x}) - \lambda = 0$$
$$...$$
$$-\frac{1}{N-1}exp(-\frac{[H(\boldsymbol{x})_N]}{N-1})Pr(\boldsymbol{y}_N|\boldsymbol{x}) - \lambda = 0$$
$$\sum_{n=1}^{N}[H(\boldsymbol{x})]_n = 0$$

解得:

$$[H(\boldsymbol{x})]_k = (N-1)\log\ Pr(\boldsymbol{y}_k|\boldsymbol{x}) - \frac{N-1}{N}\sum_{k'=1}^{N}\log\ Pr(\boldsymbol{y}_{k'}|\boldsymbol{x})$$

所以可以得到:

$$\arg\max_k\ [H(\boldsymbol{x})]_k = \arg\max_k\ Pr(\boldsymbol{y}_k|\boldsymbol{x})$$

故可以得知其最大化后验概率, 故就是一个贝叶斯分类器.

五. (20 points) **Bagging**
考虑一个回归学习任务 $f:\mathbb{R}^d\to\mathbb{R}$. 假设已经学得 T 个学习器 $\{h_1(\boldsymbol{x}),h_2(\boldsymbol{x}),\ldots,h_T(\boldsymbol{x})\}$. 将学习器的预测值视为真实值项加上误差项:

$$h_t(\boldsymbol{x}) = y(\boldsymbol{x}) + \epsilon_t(\boldsymbol{x}). \tag{5}$$

每个学习器的期望平方误差为 $\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2]$. 所有学习器的期望平方误差的平均值为:

$$E_{av} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2]. \tag{6}$$

T 个学习器得到的 Bagging 模型为:

$$H_{bag}(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} h_t(\boldsymbol{x}). \tag{7}$$

Bagging 模型的误差为:

$$\epsilon_{bag}(\boldsymbol{x}) = H_{bag}(\boldsymbol{x}) - y(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} \epsilon_t(\boldsymbol{x}), \tag{8}$$

其期望平均误差为:

$$E_{bag} = \mathbb{E}_{\boldsymbol{x}}[\epsilon_{bag}(\boldsymbol{x})^2]. \tag{9}$$

1. 假设 $\forall t \neq l$, $\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})] = 0$, $\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})\epsilon_l(\boldsymbol{x})] = 0$. 证明:

$$E_{bag} = E_{av}. \tag{10}$$

2. 请证明无需对 $\epsilon_t(\boldsymbol{x})$ 做任何假设, $E_{bag} \leq E_{av}$ 始终成立.

解:

1.

$$\begin{aligned}
E_{bag} &= \mathbb{E}_{\boldsymbol{x}}[\epsilon_{bag}(\boldsymbol{x})^2] \\
&= \mathbb{E}_{\boldsymbol{x}}[(\frac{1}{T}\sum_{t=1}^{T}\epsilon_t(\boldsymbol{x}))^2] \\
&= \frac{1}{T^2}\mathbb{E}_{\boldsymbol{x}}[\sum_{t=1}^{T}\epsilon_t(\boldsymbol{x})^2 + 2\sum_{1\leq i<j\leq T}\epsilon_i(\boldsymbol{x})\epsilon_j(\boldsymbol{x})] \\
&= \frac{1}{T^2}(\mathbb{E}_{\boldsymbol{x}}[\sum_{t=1}^{T}\epsilon_t(\boldsymbol{x})^2] + 2\sum_{1\leq i<j\leq T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_i(\boldsymbol{x})\epsilon_j(\boldsymbol{x})]) \\
&= \frac{1}{T^2}(\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2]) \\
&= \frac{1}{T}E_{av}
\end{aligned}$$

2.

$$\begin{aligned}
&\because E_{bag} = \frac{1}{T^2}(\mathbb{E}_{\boldsymbol{x}}[\sum_{t=1}^{T}\epsilon_t(\boldsymbol{x})^2] + 2\sum_{1\leq i<j\leq T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_i(\boldsymbol{x})\epsilon_j(\boldsymbol{x})]) \\
&\text{and} E_{av} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2]
\end{aligned}$$

故欲证 $E_{bag} \leq E_{av}$, 即证

$$\frac{1}{T^2}(\mathbb{E}_{\boldsymbol{x}}[\sum_{t=1}^{T}\epsilon_t(\boldsymbol{x})^2] + 2\sum_{1\leq i<j\leq T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_i(\boldsymbol{x})\epsilon_j(\boldsymbol{x})]) \leq \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2]$$

$$2\sum_{1\leq i<j\leq T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_i(\boldsymbol{x})\epsilon_j(\boldsymbol{x})] \leq (T-1)\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2]$$

$$2\sum_{1\leq i<j\leq T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_i(\boldsymbol{x})\epsilon_j(\boldsymbol{x})] \leq \sum_{1\leq i<j\leq T}(\mathbb{E}_{\boldsymbol{x}}[\epsilon_i(\boldsymbol{x})^2] + \mathbb{E}_{\boldsymbol{x}}[\epsilon_j(\boldsymbol{x})^2])$$

$$\sum_{1\leq i<j\leq T}(\mathbb{E}_{\boldsymbol{x}}[\epsilon_i(\boldsymbol{x})^2] - 2\mathbb{E}_{\boldsymbol{x}}[\epsilon_i(\boldsymbol{x})\epsilon_j(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}}[\epsilon_j(\boldsymbol{x})^2]) \geq 0$$

$$\sum_{1\leq i<j\leq T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_i(\boldsymbol{x})^2 - 2\epsilon_i(\boldsymbol{x})\epsilon_j(\boldsymbol{x}) + \epsilon_j(\boldsymbol{x})^2] \geq 0$$

$$\sum_{1\leq i<j\leq T}\mathbb{E}_{\boldsymbol{x}}[(\epsilon_i(\boldsymbol{x}) - \epsilon_j(\boldsymbol{x}))^2] \geq 0$$

概率密度为非负, 平方亦不小于 0, 故其正确, 得证.

六. (20 points) k **均值算法**

教材 9.4.1 节介绍了最经典的原型聚类算法—k 均值算法 (k-means). 给定包含 m 个样本的数据集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$, 其中 k 是聚类簇的数目, k 均值算法希望获得簇划分 $\mathcal{C} = \{C_1, C_2, \cdots, C_k\}$ 使得教材式 (9.24) 最小化, 目标函数如下:

$$E = \sum_{i=i}^{k}\sum_{\boldsymbol{x}\in C_i}\|\boldsymbol{x} - \boldsymbol{u}_i\|^2 \ . \tag{11}$$

其中 $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ 为 k 个簇的中心. 目标函数 E 也被称作均方误差和 (Sum of Squared Error, SSE), 这一过程可等价地写为最小化如下目标函数

$$E(\ , \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k) = \sum_{i=1}^{m}\sum_{j=1}^{k}\Gamma_{ij}\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2 \ . \tag{12}$$

其中 $\ \in \mathbb{R}^{m\times k}$ 为指示矩阵 (indicator matrix) 定义如下: 若 $\boldsymbol{x}_i$ 属于第 j 个簇, 即 $\boldsymbol{x}_i \in C_j$, 则 $\Gamma_{ij} = 1$, 否则为 0. k 均值聚类算法流程如算法1中所示 （即教材中图 9.2 所述算法）. 请回答以下问题:

1. 请证明, 在算法1中, Step 1 和 Step 2 都会使目标函数 J 的值降低（或不增加）;
2. 请证明, 算法1会在有限步内停止;
3. 请证明, 目标函数 E 的最小值是关于 k 的非增函数.

解:

1. Step 1: 假设使用 Step 1 会使 J 的值增加, 则说明至少存在一个点, 使得它到当前簇中心的距离大于它到所有簇中心的距离中的最小值, 即:

$$\exists i, j, j^{'}\ , \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\| > \|\boldsymbol{x}_i - \boldsymbol{\mu}_{j'}\|$$

算法 1 k 均值算法

初始化所有簇中心 $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$;
repeat
  Step 1: 确定 $\{\boldsymbol{x}_i\}_{i=1}^m$ 所属的簇, 将它们分配到最近的簇中心所在的簇.

$$\Gamma_{ij} = \begin{cases} 1, & \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2 \leq \|\boldsymbol{x}_i - \boldsymbol{\mu}_{j'}\|^2, \forall j' \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

  Step 2: 对所有的簇 $j \in \{1, \cdots, k\}$, 重新计算簇内所有样本的均值, 得到新的簇中心 $\boldsymbol{\mu}_j$:

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^m \Gamma_{ij} \boldsymbol{x}_i}{\sum_{i=1}^m \Gamma_{ij}} \tag{14}$$

until 目标函数 J 不再变化.

然而其违背了算法的划分依据, 矛盾, 故 Step 1 不会使 J 的值增加.

Step 2: 对 (15) 式 d 对 $\boldsymbol{\mu}_j$ 求偏导可得:

$$\sum_{i=1}^m \Gamma_{ij}(\boldsymbol{\mu}_j - \boldsymbol{x}_i) = 0$$

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^m \Gamma_{ij} \boldsymbol{x}_i}{\sum_{i=1}^m \Gamma_{ij}}$$

故新的簇中心就是簇内所有点距离和最小的点, 故 Step 2 不会引起 J 值增加.

2. 由题, 共有 m 个样本以及 k 个簇, 此时 k-means 算法一定会在有限的 $k^m + 1$ 次内停止.

   理由如下: 因为一共有 k 个簇, 以及 m 个样本, 故一共有 k^m 种可能的分类情况, 由抽屉原理可知, 假设执行 $k^m + 1$ 次时结果为 n, 由于其结果一定在前面出现过, 所以其非递增, 故而得到第 $k^m + 1$ 次的结果一定等于第 k^m 次, 故算法此时一定停止了, 即在有限步内停止.

3. 假设当聚类簇数目为 k 时, 目标函数 E 有最小值 E_0. 此时添加一个新的聚类簇中心, 使其总数加 1, 重复执行算法 1 中的 Step 1,Step 2, 直到终止, 会得到一个比 E_0 更小或者等于 E_0 的返回值, 设为 E_1.

   而在聚类簇数目为 $k+1$ 时目标函数理论最小值假设为 $E_1^{'}$, 有 $E_1^{'} \leq E_1$, 故 $E_1^{'} \leq E_0$. 所以可以证明目标函数 E 的最小值是关于 k 的非增函数.