姓名：杜兴豪
学号：201300096

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. 易知，分类器将真实标记为 c_i 的样本分类错误带来的期望损失为

$$R(c_i|\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_{ij} P(c_j|\boldsymbol{x})$$

同时，分类器可以选择拒绝判断，带来期望损失为 λ_h。分类器每次判断时将选择最小化损失，假设分类器交给人类判断将得到结果为 c_0 类，有

$$h^*(\boldsymbol{x}) = \begin{cases} \arg\min\limits_{c \in \mathcal{Y}} R(c|\boldsymbol{x}) & \min\limits_{c \in \mathcal{Y}} R(c|\boldsymbol{x}) < \lambda_h \\ c_0 & \min\limits_{c \in \mathcal{Y}} R(c|\boldsymbol{x}) \geq \lambda_h \end{cases}$$

或简记为

$$h^*(\boldsymbol{x}) = \arg \min_{i \in [0,N]} R(c_i|\boldsymbol{x})$$

其中

$$R(c_0|\boldsymbol{x}) = \lambda_h$$

为分类器拒绝判断产生的损失。

2. 根据第一小问的结论，容易知道
   (a) 当 $\min\limits_{c \in \mathcal{Y}} R(c|\boldsymbol{x}) \geq \lambda_h$ 时，分类器将一直拒绝分类
   (b) 当 $\max\limits_{c \in \mathcal{Y}} R(c|\boldsymbol{x}) < \lambda_h$ 时，分类器不会拒绝分类任何样例
3. 由期望损失公式容易知道

(a) 将 1 类错分为 2 类的期望损失为 $1-p_1$

(b) 把 2 类分为 1 类的期望损失为 p_1

由第二小问知，分类器拒绝预测的充要条件为

$$\min\{1-p_1, p_1\} \geq \lambda = 0.3$$

得到

$$0.3 \leq p_1 \leq 0.7$$

此时分类器任何预测损失都要大于拒绝分类，将拒绝预测。同样的，当

$$p_1 < 0.3$$

时，由于将 2 类错分为 1 类的损失太小，以至于分类器将永远预测为 1 类。当

$$p_1 > 0.7$$

时，同理可知分类器将永远预测为 2 类。因此我们找到了这样的一组解

$$\theta_1 = 0.3$$
$$\theta_2 = 0.7$$

存在性得以证明。

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. 由于抛硬币只有正面和反面两种情况，我们假设其分布模型为二项分布。令正面概率为 θ，则对本次观测的数据集 D，我们观测到的似然为

$$P(D|\theta) = \theta^{99}$$

取负对数，则可获得对数似然

$$LL(\theta) = -99\log\theta$$

观察到

$$\frac{\partial LL(\theta)}{\partial \theta} = -\frac{99}{\theta} < 0$$

也即原似然函数随着 θ 增大而概率逐渐增大。因此这里的 θ 取最大值 1。则可估计

$$P(\text{第 100 次为正面}|\theta) = \theta = 1$$

2. 根据专家的意见，我们知道 $\theta = 0.5$ 因此有

$$P(\text{第 100 次为正面}|\theta) = \theta = 0.5$$

3. (a) 先进行极大似然估计。似然函数为

$$P(\boldsymbol{x}|\theta) = \theta^{100}(1-\theta)^{300}$$

取负对数，得

$$LL(\theta) = -100\log\theta - 300\log(1-\theta)$$

对 θ 求偏导，令其等于 0 得

$$\frac{\partial LL(\theta)}{\partial \theta} = -\frac{100}{\theta} - \frac{300}{1-\theta} = 0$$

可解出

$$\theta = 0.25$$

(b) 再进行最大后验估计。根据 θ 服从 $N(\frac{1}{2}, \frac{1}{900})$ 的高斯分布，有

$$P(\theta) = \frac{30}{\sqrt{2\pi}} e^{-450(\theta-0.5)^2}$$

因此我们可以得到优化目标为

$$\arg\max_{\theta} P(\boldsymbol{x}|\theta)P(\theta) = \theta^{100}(1-\theta)^{300}\frac{30}{\sqrt{2\pi}} e^{-450(\theta-0.5)^2}\mathrm{d}x$$

取负对数，得到

$$LL(\theta) = -100\log\theta - 300\log(1-\theta) - 450(\theta-0.5)^2 + \text{const}$$

对 θ 求偏导，令其等于 0 得

$$\frac{\partial LL(\theta)}{\partial \theta} = -(\frac{100}{\theta} - \frac{300}{1-\theta} + 4.5\theta - 4.5\theta^2) = 0$$

解得

$$\hat{\theta} = \frac{1}{3}$$

满足 $\theta \in [0,1]$ 范围

4. 极大似然估计认为 θ 是一个固定存在的值，通过似然估计找到即可，结果是针对当前投硬币得到的样本，找出最可能出现样本情况的概率 θ;
而最大后验估计认为要考虑先验的影响，需要两方面考虑，结果为结合概率 θ 的实际可能分布情况，找出最可能出现样本情况的 θ 的取值。

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. 假设样本分布符合多项分布，则最大化似然写为

$$\begin{aligned} \max \quad & \frac{n!}{\prod_i n_i!} \prod_i p_i^{n_i} \\ \text{s.t.} \quad & \sum_i n_i = n \end{aligned}$$

统计 label_test，得出结果：39 个 0，37 个 1，44 个 2，共三类，因此对数似然写为

$$LL(p_1, p_2) = \sum_{i=1}^{3} n_i \log p_i = 39 \log p_1 + 37 \log p_2 + 44 \log(1 - p_1 - p_2)$$

分别求偏导令为 0 得

$$\frac{\partial LL(p_0,p_1)}{\partial p_0} = \frac{39}{p_0} - \frac{44}{1-p_0-p_1} = 0$$
$$\frac{\partial LL(p_0,p_1)}{\partial p_1} = \frac{37}{p_1} - \frac{44}{1-p_0-p_1} = 0$$

解出概率

$$p_0 = \frac{13}{40}$$
$$p_1 = \frac{37}{120}$$
$$p_2 = 1 - p_0 - p_1 = \frac{11}{30}$$

2. 构建分类器及 2，3 问的问题代码如下

```python
from collections import Counter
import numpy as np
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
import matplotlib.pyplot as plt

def load_data ():
    feature, label = datasets.load_iris(return_X_y=True)
    print(feature.shape)
    print(label.shape)
    return feature, label

feature, label = load_data()
feature_train, feature_test, label_train, label_test = train_test_split(feature,
    label, test_size = 0.2, random_state = 0)
# print(Counter(label_train))
# print(Counter(label_test))

# question 2
classifier = GaussianNB()
classifier = classifier.fit(feature_train, label_train)
print(classifier.predict(feature_test))
print(classifier.score(feature_test, label_test))

# question 3
classifier.class_prior_ = [1./3, 1./3, 1./3]
classifier = classifier.fit(feature_train, label_train)
print(classifier.predict(feature_test))
print(classifier.score(feature_test, label_test))
```

测试集上的准确率为 0.9666666666666667

3. 指定为均匀分布后，测试集上的准确率仍为 0.9666666666666667

4. 统计每个类别的数据数值分布并绘图，可见所有都符合正态分布的性质（两边少中间多）。因此条件概率可选为正态分布。

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 x 对应的标记 y(x) −1, +1. 记第 t 个基学习器及其权重为 ht 和 t, 采用 T 个基学习器加权得到的集成学习器为 H(x) = T t=1 tht(x). AdaBoost 最小化指数损失: exp = Ex D

[e−y(x)H(x)]. 1. 在 AdaBoost 训练过程中, 记前 t 个弱学习器的集成为 Ht(x) = t i=1 ihi(x), 该阶段优化目标为: exp,t = Ex D [e−y(x)Ht(x)] . (3) 如果记训练数据集的初始分布为 D0 = D, 那么第一个弱学习器的训练依赖于数据分布 D0. 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 D1, 然后基于 D1 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 Dt−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 Dt 与 Dt−1 的关系; 第 4 页（共 8 页）

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. 由加性模型可知，分类器 H_{t-1} 和 H_{t-2} 的关系为

$$H_{t-1}(\boldsymbol{x}) = H_{t-2}(\boldsymbol{x}) + \alpha_{t-1}h_{t-1}(\boldsymbol{x})$$

   其中 α_{t-1} 为基分类器 h_{t-1} 的权重。由 $\mathcal{D}_t$ 的表达式可以推导

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

2. 该损失函数满足 $H(x)$ 与 $\boldsymbol{y}$ 的相似性越大，其值越小。并且零均值化的输出可以帮助去除数值大小对结果的影响，让输出结果可读性更好。

3. 由加性模型和基分类器的所有类别对应位置和为零可知

$$\sum_{i=1}^{N}[H(\boldsymbol{x})]_i = \sum_{i=1}^{N}[\sum_{j=1}^{K}\alpha_j h_j(\boldsymbol{x})]_i = \sum_{j=1}^{K}\alpha_j\sum_{i=1}^{N}[h_i(\boldsymbol{x})]_i = 0 \tag{1}$$

若 $H(\boldsymbol{x})$ 能最小化损失函数，则可写出优化问题

$$\begin{aligned} \min \quad & \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-\frac{1}{N}\boldsymbol{y}^\top H(\boldsymbol{x})}\right] \\ \text{s.t.} \quad & \sum_{i=1}^{N}[H(x)]_i = 0 \end{aligned}$$

引入 Lagrange 乘子，得到

$$L(H(x),\lambda) = \sum_{i=1}^{N} e^{-\frac{1}{N-1}[H(x)]_i} P(y_i \mid x) + \lambda \sum_{i=1}^{N}[H(x)]_i$$

分别对 $\lambda, [H(x)]_i$ 求偏导并令为 0，有

$$\begin{aligned} \frac{\partial L(H(x),\lambda)}{\partial [H(x)]_i} &= -\frac{1}{N-1} e^{-\frac{1}{N-1}[H(x)]_i} P(y_i \mid x) + \lambda = 0, i = 1, 2\cdots, N \\ \frac{\partial L(H(x),\lambda)}{\partial \lambda} &= \sum_{i=1}^{N}[H(x)]_i = 0 \end{aligned}$$

解方程组可得

$$[H(x)]_i = (N-1)\log P(y_i \mid x) - \frac{N-1}{N}\sum_{j=1}^{N}\log P(y_j \mid x)$$

因此有

$$\arg\max_i [H(x)]_i = \arg\max_i P(y_i \mid x)$$

即达到了贝叶斯最优错误率

四. (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:

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

修改为：探究两者之间的关系

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. 由假设得

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

2. 由第一问证明过程可知

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

由排序不等式可知，假设排列顺序为 $1, 2, \cdots, T, 1, 2, \cdots$, 我们有：对任意 $k \in [1, T-1]$

$$\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2] = \mathbb{E}_{\boldsymbol{x}}[\sum_{t=1}^{T}\epsilon_t(\boldsymbol{x})^2] \geq \mathbb{E}_{\boldsymbol{x}}[\sum_{i=1}^{T}\epsilon_i(\boldsymbol{x})\epsilon_{i+k}(\boldsymbol{x})] = \sum_{i=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_i(\boldsymbol{x})\epsilon_{i+k}(\boldsymbol{x})]$$

而经过重新排列，我们有

$$\begin{aligned}
\sum_{i=1}^{T}\sum_{j\neq i}^{T}\mathbb{E}_{vx}[\epsilon_i(\boldsymbol{x})\epsilon_j(\boldsymbol{x})] &= \sum_{k=1}^{T-1}\sum_{i=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_i(\boldsymbol{x})\epsilon_{i+k}(\boldsymbol{x})] \\
&\leq (T-1)\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2]
\end{aligned}$$

由此，我们可以计算出

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

因此，在无任何对 $\epsilon_t(\boldsymbol{x})$ 的解释下，$E_{bag} \leq E_{av}$ 始终成立。

五. (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:

$$E = \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in C_i} \|\boldsymbol{x} - \boldsymbol{u}_i\|^2 \ . \tag{11}$$

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

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

其中 $\Gamma \in \mathbb{R}^{m \times k}$ 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:

算法 1 k 均值算法

1: 初始化所有簇中心 $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$;
2: **repeat**
3: **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.

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

4: **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 $\boldsymbol{\mu}_j$:

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

5: **until** 目标函数 J 不再变化.

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. (a) 执行 Step 1 之前，若簇 m 中存在 $\boldsymbol{x}_k, k \in [1, m]$，使得簇 m 不是 $\boldsymbol{x}_k$ 最近的簇中心所在的簇，则存在簇 n（此处直接假设簇 n 为其最近的簇）, 满足

$$\|\boldsymbol{x}_k - \mu_m\|^2 \leq \|\boldsymbol{x}_k - \mu_n\|^2$$

执行 Step 1 之后，将 x_k 划分到簇 n 中，则目标函数变化为

$$J' = J - \|\boldsymbol{x}_k - \mu_m\|^2 + \|\boldsymbol{x}_k - \mu_n\|^2 \leq J$$

显然使得目标函数 J 减小了。

(b) i. 假设所有簇没有加入或分出元素，则显然执行 Step 2 之后 J 不发生变化

ii. 若存在改变簇的元素，由于它距离新簇中心距离很大，在改变簇中心后显然可以知道簇内距离和减小，并且关于这个点 $\boldsymbol{x}_k$ 离开的簇 m 和到达的簇 n，显然它到簇中心的距离也减小了，因此在整体目标函数减小了。

2. 若所有簇都不发生变化，则直接满足结束循环条件，立刻退出。
因此在过程中不会发生所有簇都没变化，由第一小问可以知道，每次循环中目标函数 $\mathcal{J}$ 是严格单调减的，而 $\mathcal{J}$ 的值显然大于等于 0，单调减有下界，则一定收敛。
通过 Step 1 可以知道，每次纳入的新点距离应该比簇中某些其他点距离簇中心的距离要大，因此每次减小的步长并非无穷小量，结合收敛有下界可以知道，必存在有限多步之后，J 到达最小值，退出循环。
因此算法一定在有限步内停止。

3. 由前两小问可以知道，当我们选定一个 k 值之后，算法将在有限步内给出最好划分。此时加入一个新类，则一定存在一些点被纳入新类中去，而目标函数 E 值不会在 Step 12 中增大，因此加入新类后总函数值仍然不增。对于类数大于等于所有样本数的情况，每个点对应一个或多个类，此时目标函数达到极小值 0，此后不再增加。
因此目标函数 E 的最小值关于 k 非增。