姓名：方盛俊
学号：201300035

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 $\mathcal{Y} = \{c_1, c_2, \ldots, c_N\}$, $\lambda_{ij}$ 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 $\lambda_h$ 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 $\lambda_h$ 取何值时, 分类器 $h^\star$ 将一直拒绝分类? 当 $\lambda_h$ 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. 我们设定一个新的分类 $c_h$, 用其来表示” 拒绝”, 也就是交给人类专家处理. 则我们可知, 将 $\boldsymbol{x}$ 分类为 $c_h$ 的期望损失, 也即风险为 $\lambda_h$, 所以有 $R(c_h|\boldsymbol{x}) = c_h$.

   令 $\mathcal{Y}' = \mathcal{Y} \cup \{c_h\}$. 此时, 贝叶斯最优分类器 $h^*(\boldsymbol{x})$ 的表达式仍可以写为

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

   其中 $R(c_h|\boldsymbol{x}) = c_h$.

2. 当 $R(c_h|\boldsymbol{x})$ 是 $R(c_i|\boldsymbol{x})$ 之中的最小值, 也就是 $c_h \leq R(c_i|\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_{ij} P(c_j|\boldsymbol{x})$, 对于所有 $1 \leq i \leq N$ 时, 分类器将一直拒绝分类.

   如果令表达式与 $\boldsymbol{x}$ 无关, 则我们有 $c_h \leq \min\{\sum_{j=1}^{N} \lambda_{ij} P(c_j|\boldsymbol{x})\} = \min\{\lambda_{ij}\}$. 也即是 $c_h$ 小于等于 $\lambda_{ij}$ 的最小值时.

   当 $R(c_h|\boldsymbol{x})$ 是 $R(c_i|\boldsymbol{x})$ 之中的最大值, 也就是 $c_h \geq R(c_i|\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_{ij} P(c_j|\boldsymbol{x})$, 对于所有 $1 \leq i \leq N$ 时, 分类器将一直拒绝分类.

   如果令表达式与 $\boldsymbol{x}$ 无关, 则我们有 $c_h \leq \max\{\sum_{j=1}^{N} \lambda_{ij} P(c_j|\boldsymbol{x})\} = \max\{\lambda_{ij}\}$. 也即是 $c_h$ 大于等于 $\lambda_{ij}$ 的最大值时.

3. 我们将二分类问题两种分类的风险和拒绝的风险计算得

   $R(c_1|\boldsymbol{x}) = \lambda_{11}P(c_1|\boldsymbol{x}) + \lambda_{12}P(c_2|\boldsymbol{x}) = 1 - p_1$

   $R(c_2|\boldsymbol{x}) = \lambda_{21}P(c_1|\boldsymbol{x}) + \lambda_{22}P(c_2|\boldsymbol{x}) = p_1$

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

   由贝叶斯最优分类器表达式 $h^*(\boldsymbol{x}) = \arg\min_{c\in\mathcal{Y}'} R(c|\boldsymbol{x})$ 可知

   当 $p_1 \le \lambda_h$ 且 $p_1 \le 1 - p_1$, 即 $p_1 \le \min\{\lambda_h, 0.5\} = 0.3$ 时, 预测为第二类.

   当 $\lambda_h \le 1 - p_1$ 且 $\lambda_h \le p_1$, 即 $0.3 = \lambda_h \le p_1 \le 1 - \lambda_h = 0.7$ 时, 拒绝预测.

   当 $1 - p_1 \le \lambda_h$ 且 $1 - p_1 \le p_1$, 即 $p_1 \ge \max\{1 - \lambda_h, 0.5\} = 0.7$ 时, 预测为第二类.

   则我们有 $\theta_1 = 0.3, \theta_2 = 0.7$.

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 $\theta$, 某抛硬币专家主观认为 $\theta \sim \mathcal{N}(\frac{1}{2}, \frac{1}{900})$, 即 $\theta$ 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 $\theta$ 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. 我们令 $D = \{x_1, x_2, \cdots, x_{99}\}$.

   使用最大似然估计, 假设硬币正面向上的概率为 $\theta$, 记 $x = 1$ 为正面朝上, $x = 0$ 为反面朝上. 则对数似然为

$$\begin{aligned} LL(\theta) &= \log P(D|\theta) \\ &= \sum_{x\in D} \log P(x|\theta) \\ &= \sum_{x\in D} \log\theta \\ &= 99\log\theta \end{aligned}$$

   则参数 $\theta$ 的极大似然估计为 $\hat{\theta} = \arg\max_\theta LL(\theta) = 1.0$.

   则第 100 次抛硬币正面朝上的概率为 $P(1|\theta) = \theta = 1.0$.

2. 如果依然使用频率主义学派的思想, 认为参数 $\theta$ 是一个客观存在的固定值, 那么根据专家的见解" 肯定有 50

   设前 99 次均为正面为事件 $A$, 第 100 次为正面为事件 $B$, 则有

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)} = \frac{\theta \cdot \theta^{99}}{\theta^{99}} = \theta = 0.5$$

   则在第 100 次抛硬币时, 其正面朝上的概率为 $P(B|A) = 0.5$.

   如果使用贝叶斯学派的思想, 将 $P(x) = 0.5$ 视作先验, 对其进行最大后验估计. 由于我们无法确定参数 $\theta$ 的分布, 因此我们也无法准确地计算出 $\hat{\theta}$ 的值, 也就无法知道最终的概率. 但是我们知道的是, 最终概率介于 0.5 和 1.0 之间, 具体的值依选取的分布而定.

3. 我们令 $D = \{x_1, x_2, \cdots, x_{400}\}$.

   首先做极大似然估计:

   对数似然为

$$\begin{aligned} LL(\theta) &= \log P(D|\theta) \\ &= \sum_{x \in D} \log P(x|\theta) \\ &= 100 \log \theta + 300 \log(1-\theta) \\ &= 100(\log \theta + 3 \log(1-\theta)) \end{aligned}$$

   则参数 $\theta$ 的极大似然估计为 $\hat{\theta} = \arg\max_\theta LL(\theta) = 0.25$.

   然后做最大后验估计:

   由贝叶斯公式可知

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

   由于先对于 $\theta$ 来说, $P(x)$ 与 $\theta$ 无关, 可以视作常数, 因此

$$\hat{\theta} = \arg\max_\theta P(\theta|D) = \arg\max_\theta P(D|\theta)P(\theta)$$

   由于专家认为 $\theta \sim \mathcal{N}(\frac{1}{2}, \frac{1}{900})$, 因此先验为 $P(\theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\theta-\mu)^2}{2\sigma^2}} = \frac{1}{30\sqrt{2\pi}} e^{-450(\theta-0.5)^2}$.

   则有

$$\begin{aligned} \hat{\theta} &= \arg\max_\theta P(D|\theta)P(\theta) \\ &= \arg\max_\theta \theta^{100} \cdot (1-\theta)^{300} \cdot \frac{1}{30\sqrt{2\pi}} e^{-450(\theta-0.5)^2} \\ &= \arg\max_\theta 100 \log \theta + 300 \log(1-\theta) - 450(\theta-0.5)^2 \\ &= \arg\max_\theta 2 \log \theta + 6 \log(1-\theta) - 9(\theta-0.5)^2 \end{aligned}$$

   令 $L(\theta) = 2 \log \theta + 6 \log(1-\theta) - 9(\theta-0.5)^2$, 则有

$$\begin{aligned} \frac{\mathrm{d}L(\theta)}{\mathrm{d}\theta} &= \frac{2}{\theta} - \frac{6}{1-\theta} - 18\theta + 9 \\ &= \frac{-18\theta^3 + 27\theta^2 - \theta - 2}{\theta(\theta-1)} \end{aligned}$$

令 $-18\theta^3+27\theta^2-\theta-2=0$ 且 $0<\theta<1$ 可得 $\hat{\theta}=\theta=\dfrac{1}{3}\approx 0.33$.

4. 极大似然估计是频率主义学派的思想, 认为参数 $\theta$ 是一个客观存在的固定值, 可以通过优化似然函数等准则来确定参数值, 因此最后的估计出的 $\theta$ 是由所测数据唯一决定的, 不会被人类专家的先验概率知识影响, 因此得到的结果常常比较" 极端". 例如第 (1) 问中 $\theta$ 被干脆地估计成了 1.0, 也就是必然事件. 而第 (3) 问中地极大似然估计所得结果是 0.25, 比较偏离 0.5.

   最大后验估计是贝叶斯学派地思想, 认为参数 $\theta$ 是未观察到的随机变量, 其本身也可以有分布, 而不是一个固定值. 所以我们可以引入人类专家的先验 $\theta\sim\mathcal{N}(\frac{1}{2},\frac{1}{900})$, 然后用其参与 $\theta$ 的估计. 最后得出的结果受数据影响也较小, 因此最终结果 0.33 没有那么偏离 0.5.

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. 检查训练集上类别分类情况可知, 分类为类别 0 的数量为 39, 分类为类别 0 的数量为 37, 分类为类别 0 的数量为 44.

   由题意可知 $n = 120$. 设随机变量 $Y_0$ 为 $n$ 次中分类为类别 0 的次数, $Y_1$ 为分为类别 1 的次数, $Y_2$ 为分为类别 2 的次数. 则我们有多项分布

   $$P(Y_0 = y_0, Y_1 = y_1, Y_2 = y_2) = \frac{n!}{y_0!y_1!y_2!} p_0^{y_0} p_1^{y_1} p_2^{y_2}$$

   其中 $y_0 + y_1 + y_2 = n$, $p_0 + p_1 + p_2 = 1$, $0 \le p_i \le 1$, 且 $p_0, p_1, p_2$ 分别对应分类为对应类别的概率.

   由极大似然估计可知

   $$\begin{aligned}
   \hat{p}_0, \hat{p}_1, \hat{p}_2 &= \arg\max_{p_0,p_1,p_2} P(Y_0 = 39, Y_1 = 37, Y_2 = 44) \\
   &= \arg\max_{p_0,p_1,p_2} \frac{120!}{39!37!44!} p_0^{39} p_1^{37} p_2^{44} \\
   &= \arg\max_{p_0,p_1,p_2} 39\log p_0 + 37\log p_1 + 44\log p_2
   \end{aligned}$$

   我们编写 Python 代码优化可得最终结果

   $$\hat{p}_0 = 0.32500634, \hat{p}_1 = 0.30835317, \hat{p}_2 = 0.3666405$$

   因此先验为

   $$P(y) = \begin{cases} 0.32500634, & y = 0 \\ 0.30835317, & y = 1 \\ 0.3666405, & y = 2 \end{cases}$$

2. 代码为:

```
GNB_classifier = GaussianNB()
GNB_classifier.fit(feature_train, label_train)
print(f"score: {GNB_classifier.score(feature_test, label_test)}")
```

   最终测试性能得分为 0.9666667.

3. 代码为:

```
GNB_classifier = GaussianNB(priors=[1./3., 1./3., 1./3.])
GNB_classifier.fit(feature_train, label_train)
print(f"score: {GNB_classifier.score(feature_test, label_test)}")
```

   手动指定先验为三个类上的均匀分布, 即各个类均为 $\frac{1}{3}$, 最终测试性能得分仍为 0.9666667.

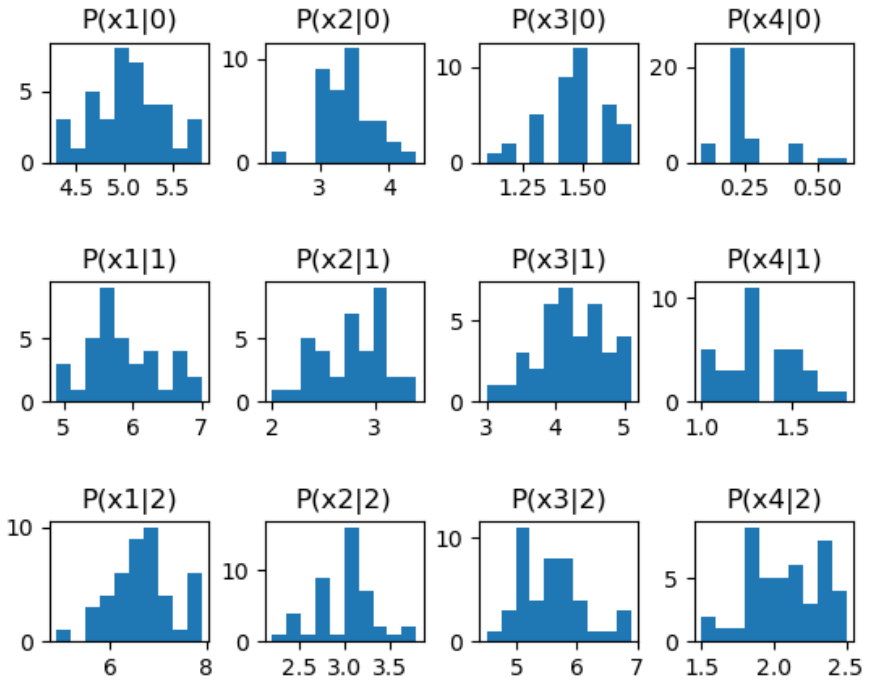


Figure 1: 直方图

4. 我们画出每个类别 $(y = 0, 1, 2)$ 下不同特征 $(x_1, x_2, x_3, x_4)$ 对应的频率直方图, 这些图像部分地展现了其真实的数值分布的形状, 我们可以根据这些图像形状判断出应该使用哪一类概率分布形式.

   如图所示, 基本所有图像都呈现出" 中间高, 两边低" 的形状, 基本没有出现" 均匀分布" 或者" 多峰" 的形状, 因此我们选择高斯分布, 就能较为真实地反映条件概率的形式.

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 $\alpha_t$, 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. AdaBoost 算法在获得 $H_{t-1}$ 之后样本分布会进行调整, 使下一轮的基学习器 $h_t$ 能纠正 $H_{t-1}$ 的一些错误. 理想的 $h_t$ 能够最小化

$$\begin{aligned}\ell_{\exp}(H_{t-1}+\alpha_t h_t|\mathcal{D}) &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})(H_{t-1}(\boldsymbol{x})+\alpha_t h_t(\boldsymbol{x}))}] \\ &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}e^{-\alpha_t f(\boldsymbol{x})h_t(\boldsymbol{x})}]\end{aligned}$$

注意到 $f^2(\boldsymbol{x}) = h_t^2(\boldsymbol{x}) = 1$, 因此可以对 $e^{-\alpha_t f(\boldsymbol{x})h_t(\boldsymbol{x})}$ 的泰勒展式近似为

$$\begin{aligned}\ell_{\exp}(H_{t-1}+\alpha_t h_t|\mathcal{D}) &\simeq \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}(1-\alpha_t f(\boldsymbol{x})h_t(\boldsymbol{x})+\frac{\alpha_t^2 f^2(\boldsymbol{x})h_t^2(\boldsymbol{x})}{2})] \\ &\simeq \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}(1-\alpha_t f(\boldsymbol{x})h_t(\boldsymbol{x})+\frac{\alpha_t^2}{2})]\end{aligned}$$

于是, 理想的基学习器

$$\begin{aligned}h_t(\boldsymbol{x}) &= \arg\min_h \ell_{\exp}(H_{t-1}+\alpha_t h|\mathcal{D}) \\ &\simeq \arg\min_h \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}(1-\alpha_t f(\boldsymbol{x})h(\boldsymbol{x})+\frac{\alpha_t^2}{2})] \\ &= \arg\max_h \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}f(\boldsymbol{x})h(\boldsymbol{x})] \\ &= \arg\max_h \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\frac{e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}]}f(\boldsymbol{x})h(\boldsymbol{x})]\end{aligned}$$

注意到 $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}]$ 是一个常数. 令 $\mathcal{D}_t$ 表示一个分布

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

则根据数学期望的定义, 这等价于令

$$\begin{aligned}h_t(\boldsymbol{x}) &= \arg\max_h \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\frac{e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}]}f(\boldsymbol{x})h(\boldsymbol{x})] \\ &= \arg\max_h \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}[f(\boldsymbol{x})h(\boldsymbol{x})]\end{aligned}$$

由 $f(\boldsymbol{x}), h(\boldsymbol{x}) \sim \{-1, +1\}$, 有

$$f(\boldsymbol{x})h(\boldsymbol{x}) = 1 - 2\mathbb{I}(f(\boldsymbol{x}) \neq h(\boldsymbol{x}))$$

则理想的基学习器

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

考虑到 $\mathcal{D}_t$ 和 $\mathcal{D}_{t-1}$ 的关系, 有

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

因此 $\mathcal{D}_t$ 和 $\mathcal{D}_{t-1}$ 的关系为

$$\mathcal{D}_t = \frac{\mathcal{D}_{t-1}(\boldsymbol{x})e^{-f(\boldsymbol{x})\alpha_{t-1}h_{t-1}(\boldsymbol{x})}}{Z_{t-1}}$$

其中规范化因子 $Z_{t-1}$ 为

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

2. 考虑指数损失函数 $e^{\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}$ 的含义. 对于任意一个样本 $\boldsymbol{x}$ 来说, 预测值 $H(\boldsymbol{x})$ 与真实值 $\boldsymbol{y}$ 越相近时, $\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})$ 就越大, 因此 $e^{\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}$ 就越小, 作为一个损失函数来说满足了需求.

   假如 $H(\boldsymbol{x})$ 无法判断属于哪个类别, 则会有 $[H(\boldsymbol{x})]_n = 0$, 最后使得 $e^{\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})} = 1$, 恰好处于一个分隔线位置.

   而加上了 $\frac{1}{N}$ 也是为了标准化损失函数的值域范围, 因为 $\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})$ 结果值受到分类数也就是 $N$ 的大小的影响, 因此应该乘上一个 $\frac{1}{N}$ 消除分类数目的影响.

3. 可以将 $\ell_{\text{multi-exp}}$ 化为

$$\ell_{\text{multi-exp}} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\sum_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}]$$

   并且我们知道 $\sum_{n=1}^{N}\boldsymbol{y}_n = \mathbf{1}\boldsymbol{y} = 0$ 且 $\sum_{n=1}^{N}[H(\boldsymbol{x})]_n = \mathbf{1}H(\boldsymbol{x}) = 0$, 构造拉格朗日函数有

$$L = \sum_{\boldsymbol{y}}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\sum_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}] + \mu\mathbf{1}\boldsymbol{y} + \lambda\mathbf{1}H(\boldsymbol{x})$$

   若 $H(\boldsymbol{x})$ 能令指数损失函数最小化, 则对 $H(\boldsymbol{x})$ 求偏导得

$$\begin{aligned}\frac{\partial L}{\partial H(\boldsymbol{x})} &= \frac{\partial}{\partial H(\boldsymbol{x})}(\sum_{\boldsymbol{y}}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\sum_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}] + \mu\mathbf{1}\boldsymbol{y} + \lambda\mathbf{1}H(\boldsymbol{x})) \\ &= \sum_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x})\frac{\partial}{\partial H(\boldsymbol{x})}e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})} + \lambda\mathbf{1} \\ &= \sum_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}\frac{\partial}{\partial H(\boldsymbol{x})}(-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})) + \lambda\mathbf{1} \\ &= -\frac{1}{N}\sum_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}\boldsymbol{y} + \lambda\mathbf{1}\end{aligned}$$

令该式等于零即有

$$-\frac{1}{N}\sum_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}\boldsymbol{y}+\lambda\mathbf{1}=0$$

两端同时乘上 $\mathbf{1}^{\mathrm{T}}$ 有

$$-\frac{1}{N}\sum_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}\mathbf{1}^{\mathrm{T}}\boldsymbol{y}+\lambda\mathbf{1}^{\mathrm{T}}\mathbf{1}=N\lambda=0$$

因此 $\lambda=0$, 可以从原式中消除.
对于原式, 我们选取其中一个 $\boldsymbol{y}$, 将与 $\boldsymbol{y}$ 无关的项移到右侧

$$P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}\boldsymbol{y}=-\sum_{\boldsymbol{y}'\neq\boldsymbol{y}} P(\boldsymbol{y}'|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}'^{\mathrm{T}}H(\boldsymbol{x})}\boldsymbol{y}'$$

不妨令 $\boldsymbol{y}_n=1$, 则有 $\boldsymbol{y}'_n=-\dfrac{1}{N-1}$, 也就可以将式子

$$P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}\boldsymbol{y}_n=-\sum_{\boldsymbol{y}'\neq\boldsymbol{y}} P(\boldsymbol{y}'|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}'^{\mathrm{T}}H(\boldsymbol{x})}\boldsymbol{y}'_n$$

变为

$$P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}=-\sum_{\boldsymbol{y}'\neq\boldsymbol{y}} P(\boldsymbol{y}'|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}'^{\mathrm{T}}H(\boldsymbol{x})}(-\frac{1}{N-1})$$

则我们有

$$\sum_{\boldsymbol{y}'\neq\boldsymbol{y}}(P(\boldsymbol{y}'|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}'^{\mathrm{T}}H(\boldsymbol{x})}-P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})})=0$$

对任意 $\boldsymbol{y}$ 都成立, 对 $N$ 个 $\boldsymbol{y}$ 对应的 $N$ 个式子进行变换, 则最终可推出

$$P(\boldsymbol{y}'|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}'^{\mathrm{T}}H(\boldsymbol{x})}=P(\boldsymbol{y}|\boldsymbol{x})e^{-\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})}$$

对任意 $\boldsymbol{y}\neq\boldsymbol{y}'$ 都成立. 因此由

$$e^{\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})-\frac{1}{N}\boldsymbol{y}'^{\mathrm{T}}H(\boldsymbol{x})}=\frac{P(\boldsymbol{y}|\boldsymbol{x})}{P(\boldsymbol{y}'|\boldsymbol{x})}$$

解得

$$\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})-\boldsymbol{y}'^{\mathrm{T}}H(\boldsymbol{x})=N\ln\frac{P(\boldsymbol{y}|\boldsymbol{x})}{P(\boldsymbol{y}'|\boldsymbol{x})}$$

所以可知 $\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})\geq\boldsymbol{y}'^{\mathrm{T}}H(\boldsymbol{x})$ 当且仅当 $P(\boldsymbol{y}|\boldsymbol{x})\geq P(\boldsymbol{y}'|\boldsymbol{x})$, 对于 $\forall\boldsymbol{y}'$.
因此有

$$\arg\max_{\boldsymbol{y}}[\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})]=\arg\max_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x})$$

并且又有

$$\begin{aligned}\arg\max_{\boldsymbol{y}}[\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})] &= \arg\max_n \sum_{n'=1}^{N} \boldsymbol{y}_{n'}[H(\boldsymbol{x})]_{n'} \\ &= \arg\max_n (1\cdot[H(\boldsymbol{x})]_n + \sum_{n'\neq n}(-\frac{1}{N-1})[H(\boldsymbol{x})]_{n'}) \\ &= \arg\max_n ((1+\frac{1}{N-1})[H(\boldsymbol{x})]_n + \sum_{n'=1}^{N}(-\frac{1}{N-1})[H(\boldsymbol{x})]_{n'}) \\ &= \arg\max_n [H(\boldsymbol{x})]_n\end{aligned}$$

这也就意味着目标函数 $\arg\max_n[H(\boldsymbol{x})]_n$ 或 $\arg\max_{\boldsymbol{y}}[\boldsymbol{y}^{\mathrm{T}}H(\boldsymbol{x})]$ 达到了贝叶斯最优错误率.

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. 对式子展开则有

$$
\begin{aligned}
E_{bag} &= \mathbb{E}_{\boldsymbol{x}}[(\frac{1}{T}\sum_{t=1}^{T}\epsilon_t(\boldsymbol{x}))^2] \\
&= \frac{1}{T^2}\mathbb{E}_{\boldsymbol{x}}[(\sum_{t=1}^{T}\epsilon_t(\boldsymbol{x}))^2] \\
&= \frac{1}{T^2}\mathbb{E}_{\boldsymbol{x}}[\sum_{t=1}^{T}\epsilon_t(\boldsymbol{x})^2 + 2\sum_{t<l}\epsilon_t(\boldsymbol{x})\epsilon_l(\boldsymbol{x})] \\
&= \frac{1}{T^2}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2] + \frac{2}{T^2}\sum_{t<l}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})\epsilon_l(\boldsymbol{x})] \\
&= \frac{1}{T^2}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2] \\
&= \frac{1}{T}E_{av}
\end{aligned}
$$

因此我们有 $E_{bag}$ 和 $E_{av}$ 关系为 $E_{bag} = \frac{1}{T}E_{av}$.

2. 可构造出式子

$$
\begin{aligned}
&\sum_{t<l}^{T}\mathbb{E}_{\boldsymbol{x}}[(\epsilon_t(\boldsymbol{x}) - \epsilon_l(\boldsymbol{x}))^2] \\
&= \mathbb{E}_{\boldsymbol{x}}[\sum_{t<l}^{T}(\epsilon_t(\boldsymbol{x})^2 + \epsilon_l(\boldsymbol{x})^2 - 2\epsilon_t(\boldsymbol{x})\epsilon_l(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x}}[(T-1)\sum_{t=1}^{T}\epsilon_t(\boldsymbol{x})^2 - 2\sum_{t<l}\epsilon_t(\boldsymbol{x})\epsilon_l(\boldsymbol{x})] \\
&= (T-1)\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2] - 2\sum_{t<l}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})\epsilon_l(\boldsymbol{x})] \\
&= T^2((\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2]) - (\frac{1}{T^2}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})^2] + \frac{2}{T^2}\sum_{t<l}\mathbb{E}_{\boldsymbol{x}}[\epsilon_t(\boldsymbol{x})\epsilon_l(\boldsymbol{x})])) \\
&= T^2(E_{av} - E_{bag}) \\
&\geq 0
\end{aligned}
$$

因此无需对 $\epsilon_t(\boldsymbol{x})$ 做任何假设即有 $E_{bag} \leq E_{av}$.

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

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

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:

**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.

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

  **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.

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. 对于 Step 1:
   由于我们有新 $\Gamma'$:

$$\Gamma'_{ij}=\begin{cases}1, & \left\|\boldsymbol{x}_i-\boldsymbol{\mu}_j\right\|^2\le\left\|\boldsymbol{x}_i-\boldsymbol{\mu}_{j'}\right\|^2,\forall j'\\ 0, & \text{otherwise}\end{cases}$$

   因此新目标函数值减去原目标函数值为

$$\begin{aligned}E'-E&=\sum_{i=1}^{m}\sum_{j=1}^{k}\Gamma'_{ij}\left\|\boldsymbol{x}_i-\boldsymbol{\mu}_j\right\|^2-\sum_{i=1}^{m}\sum_{j=1}^{k}\Gamma_{ij}\left\|\boldsymbol{x}_i-\boldsymbol{\mu}_j\right\|^2\\&=\sum_{i=1}^{m}\sum_{j=1}^{k}(\Gamma'_{ij}-\Gamma_{ij})\left\|\boldsymbol{x}_i-\boldsymbol{\mu}_j\right\|^2\\&=\sum_{i=1}^{m}(\left\|\boldsymbol{x}_i-\boldsymbol{\mu}_j\right\|^2-\left\|\boldsymbol{x}_i-\boldsymbol{\mu}_{j'}\right\|^2)\\&\le 0\end{aligned}$$

   其中 $\boldsymbol{\mu}_j$ 是离 $\boldsymbol{x}_i$ 最近的簇, 而 $\boldsymbol{\mu}_{j'}$ 可能是任意一个簇, 因此我们有 $E'\le E$. 即 Step 1 会使目标函数 $J$ 的值降低或不增加.

对于 Step 2:

对于任意一个簇 $j$ 来说, 它的簇中心原来是 $\boldsymbol{\mu}_j$, 可能是任意一个向量, 之后被优化为

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

原来的目标函数可以改写为

$$E = \sum_{j=1}^{k} \sum_{i=1}^{m} \Gamma_{ij} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|^2$$

即交换求和符号, 这样我们只需要证明 $E_j = \sum_{i=1}^{m} \Gamma_{ij} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|^2$ 降低或不增加即可.

$$\begin{aligned}
E'_j - E_j &= \sum_{i=1}^{m} \Gamma_{ij} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}'_j \right\|^2 - \sum_{i=1}^{m} \Gamma_{ij} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|^2 \\
&= \sum_{i=1}^{m} \Gamma_{ij} (\left\| \boldsymbol{x}_i - \boldsymbol{\mu}'_j \right\|^2 - \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|^2) \\
&= \sum_{i=1}^{m} \Gamma_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}'_j + \boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}} (\boldsymbol{x}_i - \boldsymbol{\mu}'_j - \boldsymbol{x}_i + \boldsymbol{\mu}_j) \\
&= \sum_{i=1}^{m} \Gamma_{ij} (\boldsymbol{\mu}_j - \boldsymbol{\mu}'_j)^{\mathrm{T}} (2\boldsymbol{x}_i - \boldsymbol{\mu}'_j - \boldsymbol{\mu}_j) \\
&= (\boldsymbol{\mu}_j - \boldsymbol{\mu}'_j)^{\mathrm{T}} (2(\sum_{i=1}^{m} \Gamma_{ij} \boldsymbol{x}_i) - (\sum_{i=1}^{m} \Gamma_{ij})(\boldsymbol{\mu}'_j + \boldsymbol{\mu}_j)) \\
&= (\sum_{i=1}^{m} \Gamma_{ij}) (\boldsymbol{\mu}_j - \boldsymbol{\mu}'_j)^{\mathrm{T}} (2 \frac{\sum_{i=1}^{m} \Gamma_{ij} \boldsymbol{x}_i}{\sum_{i=1}^{m} \Gamma_{ij}} - \boldsymbol{\mu}'_j - \boldsymbol{\mu}_j) \\
&= (\sum_{i=1}^{m} \Gamma_{ij}) (\boldsymbol{\mu}_j - \boldsymbol{\mu}'_j)^{\mathrm{T}} (2\boldsymbol{\mu}'_j - \boldsymbol{\mu}'_j - \boldsymbol{\mu}_j) \\
&= -(\sum_{i=1}^{m} \Gamma_{ij}) (\boldsymbol{\mu}_j - \boldsymbol{\mu}'_j)^{\mathrm{T}} (\boldsymbol{\mu}_j - \boldsymbol{\mu}'_j) \\
&\leq 0
\end{aligned}$$

因此我们有 $E'_j - E_j$, 也即有 Step 2 会使目标函数 $J$ 的值降低或不增加.

2. 假设算法不会在有限步内停止, 则目标函数的值 $E$ 一直在变化.

   由 (1) 可知, 目标函数 $E$ 的值降低或不增加, 又因为 $E$ 一直在变化, 可以将一系列 $E$ 的值视作严格单调递减数列, 由于 $E \geq 0$, 有下界, 因此一定收敛.

   并且我们可知, $E$ 的值由簇的分类 $C_j$ 和簇中心 $\boldsymbol{\mu}_j$ 唯一确定, 而 $\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{m} \Gamma_{ij} \boldsymbol{x}_i}{\sum_{i=1}^{m} \Gamma_{ij}}$, 因此 $\boldsymbol{\mu}_j$ 也由 $C_j$ 唯一确定, 也即 $E$ 由 $C_j$ 唯一确定, 其中 $j = 1, \cdots, k$.

   由于样本 $\boldsymbol{x}_i$ 是有限个的, 因此 $C_j$ 的划分方式是有限个的, 也就是 $E$ 的取值是离散的有限个的值, 再由 $E$ 是严格单调递减数列且有下界可知, 一定会有一个最小值 $E_{\min} \leq E$, 对于任何一个 $E$ 值来说. 因此 $E$ 一定会在 $E_{\min}$ 的时候停止, 与假设矛盾.

   因此算法会在有限步内停止.

3. 假设我们对于 $k$ 已经有了目标函数的最小值 $E_k$, 也就是说此时的算法已经停止了, 有了一系列的簇中心 $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k$.

   我们在这一系列簇中心的基础上, 加入一个随机的簇中心 $\boldsymbol{\mu}_{k+1}$, 令其等于任意一个样本, 形成一系列新的簇中心 $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k, \boldsymbol{\mu}_{k+1}$, 再重新将这一系列簇中心投入算法中.

   在再次投入算法之前, 可以认为 $C_{k+1}$ 簇中没有任何样本, 因此此时目标函数值依然等于 $E_k$, 没有变化.

   而由 (1) 可知, 等到算法终止之后, 目标函数的值 $E_{k+1}$ 也仍然只会降低或不增加.

   这样, 我们便证明了目标函数的最小值是关于 $k$ 的非增函数.