

姓名：张凯然
学号：201300009

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 $\mathcal{Y} = \{c_1, c_2, \dots, c_N\}$, 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 $P(c | \mathbf{x})$ 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 $h^*(\mathbf{x})$ 的表达式;
2. 人类专家的判断成本 λ_h 取何值时, 分类器 h^* 将一直拒绝分类? 当 λ_h 取何值时, 分类器 h^* 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

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (1)$$

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 \mathbf{x} , 设 $p_1 = P(c_1 | \mathbf{x})$, 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. 定义增广后的标记集合为 $\mathcal{Y}' = \mathcal{Y} \cup \{c_{N+1}\}$, 其中类别 c_{N+1} 代表分类器拒绝预测, 定义判定准则 $h: \mathcal{X} \rightarrow \mathcal{Y}'$ 的条件风险为

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

根据最小化期望风险原则, 我们应当寻找一个 h 来最小化期望风险

$$R(h) = \mathbb{E}_{\mathbf{x}} [R(h(\mathbf{x}) | \mathbf{x})]$$

因而我们只需要对每一个样本 \mathbf{x} 最小化条件风险 $R(h(\mathbf{x}) | \mathbf{x})$, 就可以得到判定准则

$$h^*(\mathbf{x}) = \begin{cases} \arg \min_{c \in \mathcal{Y}} R(c | \mathbf{x}), & \min_{c \in \mathcal{Y}} R(c | \mathbf{x}) < \lambda_h \\ c_{N+1}, & \min_{c \in \mathcal{Y}} R(c | \mathbf{x}) \geq \lambda_h \end{cases}$$

2. 根据判定准则 h , 我们容易得出这样的结论:

- 对于任意给定样例 \mathbf{x} , $\lambda_h \leq \min_{c \in \mathcal{Y}} R(c | \mathbf{x})$, 则分类器会一直拒绝分类, 即满足条件

$$\lambda_h \leq \min_{\mathbf{x} \in \mathcal{X}} \left(\min_{c \in \mathcal{Y}} R(c | \mathbf{x}) \right)$$

- 对于任意给定样例 \mathbf{x} , $\lambda_h > \min_{c \in \mathcal{Y}} R(c | \mathbf{x})$, 则分类器不会拒绝分类, 即满足条件

$$\lambda_h > \max_{\mathbf{x} \in \mathcal{X}} \left(\min_{c \in \mathcal{Y}} R(c | \mathbf{x}) \right)$$

3. 首先我们算出各个类别的条件风险值

$$R(c_1 | \mathbf{x}) = 1 \cdot P(c_2 | \mathbf{x}) = 1 - p_1$$

$$R(c_2 | \mathbf{x}) = 1 \cdot P(c_1 | \mathbf{x}) = p_1$$

$$\lambda_h = 0.3$$

- 当 $p_1 < \theta_1$ 时, 则预测为第二类, 因而有

$$p_1 \leq 1 - p_1$$

$$p_1 \leq 0.3$$

因而 $\theta_1 \leq 0.3$

- 当 $\theta_1 < p_1 < \theta_2$ 时, 则, 因而有

$$0.3 \leq 1 - p_1$$

$$0.3 \leq p_1$$

即 $0.3 \leq p_1 \leq 0.7$ 于是 $\theta_1 \geq 0.3, \theta_2 \leq 0.7$

- 当 $p_1 < \theta_1$ 时, 则预测为第二类, 因而有

$$1 - p_1 \leq p_1$$

$$1 - p_1 \leq 0.3$$

因而 $\theta_2 \geq 0.7$

因而存在这样的 $\theta_1 = 0.3, \theta_2 = 0.7$, 使得题中要求成立.

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 $\mathbf{x} = \{x_1, x_2, \dots, x_{400}\}$, 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. 假设样本满足参数为 θ 的 0-1 分布, 我们需要根据已有的 99 次实验数据对参数 θ 进行极大似然估计.

容易写出似然和对数似然为

$$\begin{aligned} L(D | \theta) &= \prod_{x \in D} P(x | \theta) \\ &= \prod_{x \in D} \theta \\ &= \theta^{99} \\ \ln L(D | \theta) &= 99 \cdot \ln \theta \end{aligned}$$

对数似然函数显然是一个关于 θ 的增函数, 因而 θ 越大越好, 又由于 $0 \leq \theta \leq 1$, 所以 θ 的极大似然估计为

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

因而第 100 次投掷硬币获得正面朝上概率的极大似然估计为 1.

2. 专家的观点并不能给出一个含有参数 θ 的概率模型, 因而无法使用最大后验估计的方法.

反而, 我们从专家的观点中直接得到了参数本身. 假定事件 A 为抛掷了 99 次硬币并且均为正面朝上, 事件 B 为抛掷一次硬币并且正面朝上. 根据专家的观点, $P(B) = 0.5$. 我们需要估计

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

因而第 100 次抛掷硬币正面朝上的概率为 0.5.

3. 使用极大似然估计:

这时我们只关注实验数据, 假设样本满足参数为 θ 的 0-1 分布, 我们需要根据已有的 400 次实验数据对参数 θ 进行极大似然估计.

容易写出似然和对数似然为

$$\begin{aligned} L(D | \theta) &= \prod_{x \in D} P(x | \theta) \\ &= \theta^{100} \cdot (1 - \theta)^{300} \\ \ln L(D | \theta) &= 100(\ln \theta + 3 \ln(1 - \theta)) \end{aligned}$$

对数似然函数对参数 θ 求导可以得到

$$\frac{\partial \ln L(D | \theta)}{\partial \theta} = 100 \left(\frac{1}{\theta} - \frac{3}{1 - \theta} \right) = 0$$

解得参数 θ 的极大似然估计为

$$\hat{\theta} = \frac{1}{4}$$

使用最大后验估计:

这时我们需要关注样本数据和先验知识两个方面.

容易写出后验函数和指数后验函数为

$$Pos(\theta | D) = L(D | \theta) \cdot f(\theta) \quad (2)$$

$$= \theta^{100} \cdot (1 - \theta)^{300} \cdot \frac{1}{\sqrt{2\pi} \cdot \frac{1}{30}} e^{-\frac{(\theta - \frac{1}{3})^2}{2 \cdot \frac{1}{900}}} \quad (3)$$

$$\ln Pos(\theta | D) = 100 (\ln \theta + 3 \ln(1 - \theta)) - 450(\theta - \frac{1}{2})^2 + \text{const} \quad (4)$$

对数后验函数对参数 θ 求偏导得到

$$2 \left(\frac{1}{\theta} - 3 \cdot \frac{1}{1 - \theta} \right) - 18 \left(\theta - \frac{1}{2} \right) = 0, 0 \leq \theta \leq 1$$

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

4. 答案如下:

- **极大似然估计**

极大似然估计 (MLE) 是频率主义学派的代表方法. 该学派认为参数虽然未知, 却是客观存在的固定值, 可以通过优化似然函数的方式求得. 这种方法完全依赖已知的样本数据, 通过调节未知参数的方式使得已有的数据出现概率最大.

- **最大后验估计**

最大后验估计 (MAP) 是贝叶斯学派的代表方法. 该学派认为参数是未观察到的随机变量, 其本身可能存在一个分布. 因而需要事先假设参数服从一个先验分布, 接着基于观测到的数据计算参数的后验分布. 这种方法不仅关注已知的样本数据, 还关注先验知识给出的概率分布, 通过调节未知参数的方式使得已有的数据出现概率最大.

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:

```
1 def load_data():
2     # 以feature, label 的形式返回数据集
3     feature, label = datasets.load_iris(return_X_y=True)
4     print(feature.shape) # (150, 4)
5     print(label.shape) # (150,)
6     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.

```
1 feature_train, feature_test, label_train, label_test = \
2     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:

$$\hat{y} = \arg \max_{y \in \{0,1,2\}} P(y) \prod_{i=1}^d P(x_i | y). \quad (5)$$

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 $P(x_i | y)$.

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 \mathbf{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 $P(x_i | y)$, 首先需要对 $P(x_i | y)$ 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. 答案如下:

在 NaiveBayes.py 文件中, 我们使用 class0, class1, class2 三个 numpy.array 收集训练集中分别属于三个类别的样本. 可以获得第 0 类样本数量 N_0 为 39, 第 1 类样本数量 N_1 为 37, 第二类样本数量 N_2 为 44. 假定以下参数: $P(y = 0) = p_0$, $P(y = 1) = p_1$, $P(y = 2) = p_2$. 似然函数需要让当前样本出现的概率最大, 因而可以写出如下的优化问题

$$\begin{aligned} \max_{p_0, p_1, p_2} \quad & \frac{120!}{N_0!N_1!N_2!} p_0^{N_0} p_1^{N_1} p_2^{N_2} \\ \text{s.t.} \quad & p_0 + p_1 + p_2 = 1 \\ & 0 \leq p_i \leq 1, i = 1, 2, 3 \end{aligned}$$

对数似然可以写作

$$\ln LL = N_0 \ln p_0 + N_1 \ln p_1 + N_2 \ln(1 - p_0 - p_1)$$

对 p_0, p_1 求偏导可以得到

$$\begin{aligned} \frac{\partial \ln LL}{\partial p_0} &= \frac{N_0}{p_0} - \frac{N_2}{1 - p_0 - p_1} = 0 \\ \frac{\partial \ln LL}{\partial p_1} &= \frac{N_1}{p_1} - \frac{N_2}{1 - p_0 - p_1} = 0 \end{aligned}$$

从而解得

$$\begin{cases} p_0 = \frac{N_0}{N_0 + N_1 + N_2} = \frac{39}{120} \\ p_1 = \frac{N_1}{N_0 + N_1 + N_2} = \frac{37}{120} \\ p_2 = \frac{N_2}{N_0 + N_1 + N_2} = \frac{44}{120} \end{cases}$$

2. 答案如下:

sklearn 中的 GaussianNB 类认为类条件概率满足正态分布, 并且根据某类样本在一个确定特征上的取值来确定其均值和方差. 在 NaiveBayes.py 文件中使用 GaussianNB 类构建了分类器 my_nb, 使用 GaussianNB 类的 score 属性检验其在测试集上性能, 得到的准确率为 0.9667

3. 答案如下:

构建了一个新的分类器 pro_nb, 指定了类别先验为均匀分布, 在 NaiveBayes.py 文件中使用 GaussianNB 类的 score 属性检验其在测试集上性能, 得到的准确率为 0.9667

4. 答案如下:

我们可以画出各个类别下不同属性值的频率分布图: Figure 1. 根据图像观察, 大部分属性值的频率分布呈现了“中间高, 两侧低”的图形, 因而可以选用概率密度满足“中间高, 两侧低”性质的分布, 例如正态分布.

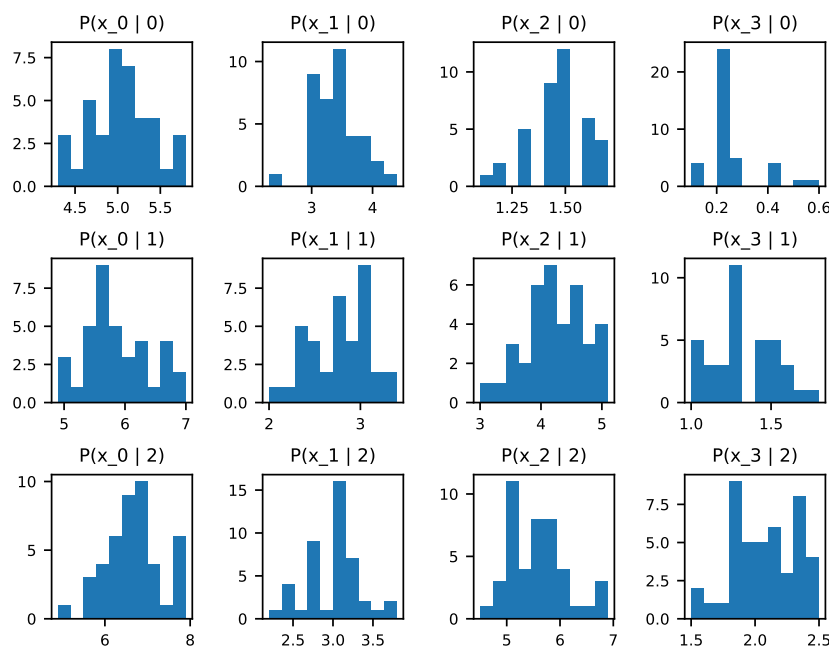


Figure 1: 各个类别下不同属性值的概率分布图

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 \mathbf{x} 对应的标记 $y(\mathbf{x}) \in \{-1, +1\}$. 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 $H(\mathbf{x}) = \sum_{t=1}^T \alpha_t h_t(\mathbf{x})$. AdaBoost 最小化指数损失: $\ell_{\text{exp}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} [e^{-y(\mathbf{x})H(\mathbf{x})}]$.

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

$$\ell_{\text{exp}, t} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} [e^{-y(\mathbf{x})H_t(\mathbf{x})}]. \quad (6)$$

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 \mathbf{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 $\sum_{n=1}^N \mathbf{y}_n = 0$, 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: $\sum_{n=1}^N [h_t(\mathbf{x})]_n = 0$. $[h_t(\mathbf{x})]_n$ 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:

$$\ell_{\text{multi-exp}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[e^{-\frac{1}{N} \sum_{n=1}^N \mathbf{y}_n [H(\mathbf{x})]_n} \right] = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[e^{-\frac{1}{N} \mathbf{y}^\top H(\mathbf{x})} \right]. \quad (7)$$

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 $\text{sign}(H(\mathbf{x}))$ 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(7)的合理性.

解:

1. 答案如下:

理想的基学习器 h_t 可以纠正以往加权分类器的 H_{t-1} 的错误, 即最小化 $\ell_{\text{exp}}(H_{t-1} + h_t | \mathcal{D})$. 根据课本推导, 可以得到

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

我们需要接着生成新的数据分布, 并使得 $h_t(\mathbf{x})$ 的生成基于这个分布, 即

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

令 \mathcal{D}_t 表示一个分布

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

利用这个关系, 可以推出 \mathcal{D}_t 与 \mathcal{D}_{t-1} 的关系.

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

2. 答案如下:

- 首先

该损失函数可以惩罚和答案 \mathbf{y} 相差甚远的分类结果. 内积 $\mathbf{y}^\top H(\mathbf{x})$ 可以度量两个向量之间的相似程度, 内积值越大, 则两向量越接近, 此时损失函数的惩罚值越小; 而内积值越小, 则两向量越分离, 此时损失函数的惩罚值越大.

- 其次

最小化该损失函数可以获得一个等效的贝叶斯分类器 (第三问中会证明), 因而达到了理论最优解

- 最后

加入系数 $-\frac{1}{N}$ 可以标准化内积值, 避免分类数目 N 对惩罚函数的值造成影响.

3. 答案如下:

理想的基学习器需要最小化多元指数损失函数

$$\begin{aligned} \arg \min_{H(\mathbf{x})} \quad & \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[e^{-\frac{1}{N} \sum_{n=1}^N \mathbf{y}_n [H(\mathbf{x})]_n} \right] \\ \text{s.t.} \quad & \sum_{n=1}^N [H(\mathbf{x})]_n = 0 \end{aligned}$$

考虑原问题的拉格朗日函数

$$\begin{aligned} & \exp \left(-\frac{[H(\mathbf{x})]_1}{N-1} \right) \Pr(\mathbf{y}_1 | \mathbf{x}) \\ & + \dots \\ & + \exp \left(-\frac{[H(\mathbf{x})]_N}{N-1} \right) \Pr(\mathbf{y}_N | \mathbf{x}) \\ & - \lambda \left(\sum_{n=1}^N [H(\mathbf{x})]_n \right) \end{aligned}$$

针对 $[H(\mathbf{x})]_n$ 和 λ 分别求导得到

$$\begin{aligned} & -\frac{1}{N-1} \exp \left(-\frac{[H(\mathbf{x})]_1}{N-1} \right) \Pr(\mathbf{y}_1 | \mathbf{x}) - \lambda = 0 \\ & \dots \\ & -\frac{1}{N-1} \exp \left(-\frac{[H(\mathbf{x})]_N}{N-1} \right) \Pr(\mathbf{y}_N | \mathbf{x}) - \lambda = 0 \\ & \sum_{n=1}^N [H(\mathbf{x})]_n = 0 \end{aligned}$$

求解上述方程组可以得到

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

因而

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

这就是一个 Bayes 分类器, 因为它最大化后验概率.

五. (20 points) **Bagging**

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

$$h_t(\mathbf{x}) = y(\mathbf{x}) + \epsilon_t(\mathbf{x}). \quad (8)$$

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

$$E_{av} = \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{\mathbf{x}}[\epsilon_t(\mathbf{x})^2]. \quad (9)$$

T 个学习器得到的 Bagging 模型为:

$$H_{bag}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T h_t(\mathbf{x}). \quad (10)$$

Bagging 模型的误差为:

$$\epsilon_{bag}(\mathbf{x}) = H_{bag}(\mathbf{x}) - y(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T \epsilon_t(\mathbf{x}), \quad (11)$$

其期望平均误差为:

$$E_{bag} = \mathbb{E}_{\mathbf{x}}[\epsilon_{bag}(\mathbf{x})^2]. \quad (12)$$

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

$$E_{bag} = E_{av}. \quad (13)$$

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

解:

1. 答案如下:

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

2. 答案如下:

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

要证明 $\mathbb{E}_{bag} \leq \mathbb{E}_{av}$, 只要证明

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

只要证明

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

由于平方和概率密度的非负性, 该式成立, 证毕.

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 $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$, 其中 k 是聚类簇的数目, k 均值算法希望获得簇划分 $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ 使得教材式 (9.24) 最小化, 目标函数如下:

$$E = \sum_{i=1}^m \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{u}_i\|^2. \quad (14)$$

其中 μ_1, \dots, μ_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

$$E(\mathbf{\Gamma}, \mu_1, \dots, \mu_k) = \sum_{i=1}^m \sum_{j=1}^k \Gamma_{ij} \|\mathbf{x}_i - \mu_j\|^2. \quad (15)$$

其中 $\mathbf{\Gamma} \in \mathbb{R}^{m \times k}$ 为指示矩阵 (indicator matrix) 定义如下: 若 \mathbf{x}_i 属于第 j 个簇, 即 $\mathbf{x}_i \in C_j$, 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 μ_1, \dots, μ_k ;
- 2: **repeat**
- 3: **Step 1:** 确定 $\{\mathbf{x}_i\}_{i=1}^m$ 所属的簇, 将它们分配到最近的簇中心所在的簇.

$$\Gamma_{ij} = \begin{cases} 1, & \|\mathbf{x}_i - \mu_j\|^2 \leq \|\mathbf{x}_i - \mu_{j'}\|^2, \forall j' \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

- 4: **Step 2:** 对所有的簇 $j \in \{1, \dots, k\}$, 重新计算簇内所有样本的均值, 得到新的簇中心 μ_j :

$$\mu_j = \frac{\sum_{i=1}^m \Gamma_{ij} \mathbf{x}_i}{\sum_{i=1}^m \Gamma_{ij}} \quad (17)$$

- 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. 答案如下:

证明 Step 1 会使目标函数 J 的值降低 (或不增加);

假设使用了 Step 1 会使目标函数 J 的值增加, 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

$$\exists i, j, j', \|\mathbf{x}_i - \mu_j\| > \|\mathbf{x}_i - \mu_{j'}\|$$

这违背了算法的划分依据, 引发矛盾, 因而假设不成立.

证明 Step 2 会使目标函数 J 的值降低 (或不增加);

证明新的簇中心是到簇内所有点距离和最小的点即可. 针对式 15 对 μ_j 求偏导得到

$$\sum_{i=1}^m \Gamma_{ij}(\mu_j - \mathbf{x}_i) = 0$$

$$\mu_j = \frac{\sum_{i=1}^m \Gamma_{ij} \mathbf{x}_i}{\sum_{i=1}^m \Gamma_{ij}}$$

因而新的中心点就是到簇内所有点距离和最小的点.

2. 答案如下:

根据题意, 我们有 m 个样本和 k 个簇, k -means 算法至多执行 $k^m + 1$ 次后会停止.

对于所有的样本点它们最多都有 k 种分类情况, 因而最多有 k^m 种可能的分类情况, 根据鸽笼原理, 执行 $k^m + 1$ 次循环后, 第 $k^m + 1$ 次循环造成的分类情况一定在前 k^m 种情况中出现过. 假定第 i 次循环后目标函数值为 SSE_i .

由于第 $k^m + 1$ 次循环造成的分类情况一定在前 k^m 种情况中出现过, 而且 SSE 一定非递增, 所以 $SSE_{k^m+1} \geq SSE_{k^m}$. 又由于 SSE 一定非递增, 所以 $SSE_{k^m+1} \leq SSE_{k^m}$, 所以 $SSE_{k^m+1} = SSE_{k^m}$. 算法一定终止了

3. 答案如下:

假定我们已经得到了目标函数 E 在聚类簇数目为 k 情况下的最小值 E_k .

现在随机增加一个新的 (不重复的) 聚类簇中心, 使得聚类簇数目变为 $k + 1$. 此时重复执行 k -means 算法的 Step1 和 Step2 直到终止, 可以得到算法的返回值 E'_{k+1} , 根据第一问的结论, $E'_{k+1} \leq E_k$.

而在聚类簇数目为 $k + 1$ 情况下的目标函数理论最小值 $E_{k+1} \leq E'_{k+1}$, 因而目标函数 E 在聚类簇数目为 $k + 1$ 情况下的最小值 E_{k+1} 小于等于目标函数 E 在聚类簇数目为 k 情况下的最小值 E_k . 即目标函数 E 最小值是关于 k 的非增函数.