姓名：方盛俊
学号：201300035

一. (20 points) **没有免费的午餐定理**

1. 根据教材 1.4 节“没有免费的午餐”定理, 所有学习算法的期望性能都和随机胡猜一样, 是否还有必要继续进行研究机器学习算法?
2. 教材 1.4 节在论述“没有免费的午餐”定理时, 默认使用了“分类错误率”作为性能度量来对分类器进行评估. 若换用其他性能度量 ℓ, 则教材中式 (1.1) 将改为

$$E_{ote}(\mathfrak{L}_a|X,f)=\sum_h\cdot\sum_{x\in\mathcal{X}-X}P(\boldsymbol{x})\ell(h(\boldsymbol{x}),f(\boldsymbol{x}))P(h|\mathcal{X},\mathfrak{L}_a)\quad(1)$$

试证明“没有免费的午餐定理”仍成立.

解:

1. 依然有必要继续研究机器学习算法.

   NFL 定理有一个重要的前提, 即所有的”问题”出现的机会相同, 即 f 是均匀分布的, 但是实际上并不是这样. 对于某一类具体的问题来说, f 一般都不是均匀分布的. 因此, 面对一个具体的问题, 一般都会有一个具体的机器学习算法, 能够取得比其他机器学习算法更好的结果.

2. 对于一个有 k 种分类结果的 k 分类器来说, 其真实目标函数可以是任何函数 $\mathcal{X}\mapsto\{1,2,\cdots,k\}$, 函数空间为 $\{1,2,\cdots,k\}^{|\mathcal{X}|}$.

   将 f 视为遵循均匀分布, 则我们有预测正确的概率 $\Pr(h(\boldsymbol{x})=f(\boldsymbol{x}))=\dfrac{1}{k}$, 是一个常数. 预测错误的概率是 $\Pr(h(\boldsymbol{x})\neq f(\boldsymbol{x}))=\dfrac{k-1}{k}$.

   对于分类问题, 任意一种性能度量 $\ell(h(\boldsymbol{x}),f(\boldsymbol{x}))$ 的输出只有两种结果, 设预测正确的结果为 c_1, 预测错误的结果为 c_0, 则有分布列 $\Pr(\ell(h(\boldsymbol{x}),f(\boldsymbol{x}))=c_1)=\dfrac{1}{k}$, $\Pr(\ell(h(\boldsymbol{x}),f(\boldsymbol{x}))=c_0)=\dfrac{k-1}{k}$.

那么有期望 $\mathbb{E}[\ell(h(\boldsymbol{x}), f(\boldsymbol{x}))] = \frac{1}{k} \cdot c_1 + \frac{k-1}{k} \cdot c_0$, 我们可以设 $c = \frac{1}{k} \cdot c_1 + \frac{k-1}{k} \cdot c_0$. 以二分类问题的错误率为例子, 此时 $c = \frac{1}{2} \cdot 0 + \frac{2-1}{2} \cdot 1 = \frac{1}{2}$.
则我们有

$$\begin{aligned}
\sum_f E_{ote}(\mathfrak{L}_a|X,f) &= \sum_f \sum_h \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x})\ell(h(\boldsymbol{x}),f(\boldsymbol{x}))P(h|X,\mathfrak{L}_a) \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x}) \sum_h P(h|X,\mathfrak{L}_a) \sum_f \ell(h(\boldsymbol{x}),f(\boldsymbol{x})) \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x}) \sum_h P(h|X,\mathfrak{L}_a) c k^{|\mathcal{X}|} \\
&= c k^{|\mathcal{X}|} \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x}) \sum_h P(h|X,\mathfrak{L}_a) \\
&= c k^{|\mathcal{X}|} \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x}) \cdot 1
\end{aligned}$$

可以看出, 总误差依然与学习算法无关, 因此对于其他的性能度量来说, NFL 定理依然成立.

二. (15 points) **线性回归**

给定包含 m 个样例的数据集 $\boldsymbol{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$, 其中 $\boldsymbol{x}_i = (x_{i1}; x_{i2}; \cdots; x_{id}) \in \mathbb{R}^d$, $y_i \in \mathbb{R}$ 为 $\boldsymbol{x}_i$ 的实数标记. 针对数据集 $\boldsymbol{D}$ 中的 m 个示例, 教材 3.2 节所介绍的 "线性回归" 模型要求该线性模型的预测结果和其对应的标记之间的误差之和最小:

$$\begin{aligned}
(\boldsymbol{w}^*, b^*) &= \frac{1}{2} \underset{(\boldsymbol{w},b)}{\arg\min} \sum_{i=1}^{m} \left(f(\boldsymbol{x}_i) - y_i\right)^2 \\
&= \frac{1}{2} \underset{(\boldsymbol{w},b)}{\arg\min} \sum_{i=1}^{m} \left(y_i - (\boldsymbol{w}^\top \boldsymbol{x}_i + b)\right)^2 . \qquad (2)
\end{aligned}$$

即寻找一组权重 $(\boldsymbol{w}, b)$, 使其对 $\boldsymbol{D}$ 中示例预测的整体误差最小.[1] 定义

[1]公式 **??**中系数 $\frac{1}{2}$ 是为了化简后续推导. 有时也会乘上 $\frac{1}{m}$ 以计算均方误差 (Mean Square Error), 由于平均误差和误差和在优化

$y = [y_1; \ldots, y_m] \in \mathbb{R}^m$, 且 $\boldsymbol{X} = \left[\boldsymbol{x}_1^\top; \boldsymbol{x}_2^\top; \cdots; \boldsymbol{x}_m^\top\right] \in \mathbb{R}^{m \times d}$, 请将线性回归的优化过程使用矩阵进行表示.

解:

原式可用矩阵表示为

$$(\boldsymbol{w}^*, b^*) = \arg\min_{(\boldsymbol{w},b)} \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - \mathbf{1}b)^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - \mathbf{1}b)$$

令 $\boldsymbol{E} = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - \mathbf{1}b)^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - \mathbf{1}b)$

对 $\boldsymbol{w}$ 求导得

$$\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{w}} = \boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{w} + \mathbf{1}b - \boldsymbol{y})$$

对 b 求导得

$$\frac{\partial \boldsymbol{E}}{\partial b} = \mathbf{1}^\top(\boldsymbol{X}\boldsymbol{w} + \mathbf{1}b - \boldsymbol{y})$$

令上面两式同时等于零即可得到 $\boldsymbol{w}$ 和 b 的最优解的闭式解.

当 $\boldsymbol{X}^\top\boldsymbol{X}$ 是满秩矩阵时, 则有

$$\boldsymbol{w}^* = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top(\boldsymbol{y} - \mathbf{1}b^*)$$

令 $\boldsymbol{T} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}$, 则

$$\boldsymbol{w}^* = \boldsymbol{T}\boldsymbol{X}^\top(\boldsymbol{y} - \mathbf{1}b^*)$$

代入有

$$\begin{aligned} b^* &= \mathbf{1}^\top\boldsymbol{y} - \mathbf{1}^\top\boldsymbol{X}\boldsymbol{w}^* \\ &= \mathbf{1}^\top\boldsymbol{y} - \mathbf{1}^\top\boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top(\boldsymbol{y} - \mathbf{1}b^*) \\ &= \mathbf{1}^\top\boldsymbol{y} - \mathbf{1}^\top\boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top\boldsymbol{y} + \mathbf{1}^\top\boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top\mathbf{1}b^* \end{aligned}$$

最后有

过程中只相差一个常数, 不影响优化结果, 因此在后续讨论中省略这一系数.

$$b^* = \frac{\mathbf{1}^\top \boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top \boldsymbol{y} - \mathbf{1}^\top \boldsymbol{y}}{\mathbf{1}^\top \boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top \mathbf{1} - 1}$$

最后将 b^* 带回即可求出

$$\boldsymbol{w}^* = \boldsymbol{T}\boldsymbol{X}^\top (\boldsymbol{y} - \mathbf{1}\frac{\mathbf{1}^\top \boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top \boldsymbol{y} - \mathbf{1}^\top \boldsymbol{y}}{\mathbf{1}^\top \boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top \mathbf{1} - 1})$$

三. (25 points) **正则化**

在实际问题中, 我们常常会遇到示例相对较少, 而特征很多的场景. 在这类情况中如果直接求解线性回归模型, 较少的示例无法获得唯一的模型参数, 会具有多个模型能够" 完美" 拟合训练集中的所有样例, 实现插值 (interpolation). 此外, 模型很容易过拟合. 为缓解这些问题, 常在线性回归的闭式解中引入正则化项 $\Omega(\boldsymbol{w})$, 通常形式如下:

$$\boldsymbol{w}^*_{\mathbf{Ridge}}, b^*_{\mathbf{Ridge}} = \arg\min_{\boldsymbol{w},b} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} + \mathbf{1}b - \mathbf{y}\|_2^2 + \lambda\Omega(\boldsymbol{w}) \ . \tag{3}$$

其中, $\lambda > 0$ 为正则化参数. 正则化表示了对模型的一种偏好, 例如 $\Omega(\boldsymbol{w})$ 一般对模型的复杂度进行约束, 因此相当于从多个在训练集上表现同等预测结果的模型中选出模型复杂度最低的一个.

考虑岭回归 (ridge regression) 问题, 即设置公式(**??**)中正则项 $\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_2^2$. 本题中将对岭回归的闭式解以及正则化的影响进行探讨.

1. 请给出岭回归的最优解 $\boldsymbol{w}^*_{\mathbf{Ridge}}$ 和 $b^*_{\mathbf{Ridge}}$ 的闭式解表达式, 并使用矩阵形式表示, 分析其最优解和原始线性回归最优解 $\boldsymbol{w}^*_{\mathbf{LS}}$ 和 $b^*_{\mathbf{LS}}$ 的区别;
2. 请证明对于任何矩阵 $\boldsymbol{X}$, 下式均成立

$$\left(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}_m\right)^{-1} \boldsymbol{X} = \boldsymbol{X} \left(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_d\right)^{-1} \ . \tag{4}$$

   请思考, 上述的结论是否能够帮助岭回归的计算, 在何种情况下能够带来帮助?
3. 针对波士顿房价预测数据 (`boston`), 编程实现原始线性回归模型和岭回归模型, 基于闭式解在训练集上构建模型, 计算测试集上的均方误差 (Mean Square Error, MSE). 请参考`LinearRegression.py`进行模型构造.

```python
from sklearn.datasets import load_boston
from sklearn.model_selection import train_test_split
import numpy as np
import matplotlib.pyplot as plt

X, y = load_boston(return_X_y=True)
trainx, testx, trainy, testy = train_test_split(
    X, y, test_size=0.33, random_state=42)


def linReg(X_train: np.ndarray, y_train: np.ndarray) -> np.ndarray:
    # linear regression
    ones = np.ones(X_train.shape[0]).reshape(-1, 1)
    X_splash = np.hstack((X_train, ones))
    T: np.ndarray = np.linalg.inv(X_splash.transpose() @ X_splash)
    return T @ X_splash.transpose() @ y_train


def linRegMSE(X_train: np.ndarray, y_train: np.ndarray, X_test: np.ndarray, y_test:
    np.ndarray) -> float:
    weights = linReg(X_train, y_train).reshape(-1, 1)
    ones = np.ones(X_test.shape[0]).reshape(-1, 1)
    X_splash = np.hstack((X_test, ones))
    E: np.ndarray = y_test.reshape(-1, 1) - X_splash @ weights
    # return E.transpose() @ E / X_test.shape[0]
    return np.mean(E ** 2)


def reportLinRegMSE():
    return linRegMSE(trainx, trainy, testx, testy)


def ridgeReg(X_train: np.ndarray, y_train: np.ndarray, lmbd: float) -> np.ndarray:
    # ridge regression
    ones = np.ones(X_train.shape[0])
    eye = np.eye(X_train.shape[1])
    T: np.ndarray = np.linalg.inv(
        X_train.transpose() @ X_train + 2 * lmbd * eye)
    S: np.ndarray = ones @ X_train @ T @ X_train.transpose() - ones
    b = (S @ y_train) / (S @ ones)
    w = T @ X_train.transpose() @ (y_train - b * ones)
    return np.hstack((w, b))


def ridgeRegMSE(X_train: np.ndarray, y_train: np.ndarray, X_test: np.ndarray, y_test
    : np.ndarray, lmbd: float) -> float:
    weights = ridgeReg(X_train, y_train, lmbd).reshape(-1, 1)
    ones = np.ones(X_test.shape[0]).reshape(-1, 1)
    X_splash = np.hstack((X_test, ones))
    E: np.ndarray = y_test.reshape(-1, 1) - X_splash @ weights
    # return E.transpose() @ E / X_test.shape[0]
    return np.mean(E ** 2)


def reportRidgeRegMSE(lmbd):
    return ridgeRegMSE(trainx, trainy, testx, testy, lmbd)

print(linReg(trainx, trainy))
print(ridgeReg(trainx, trainy, 0))
print(reportLinRegMSE())
print(reportRidgeRegMSE(0))
lmbds = np.arange(0, 2.1, 0.2)
mses = [reportRidgeRegMSE(lmbd) for lmbd in lmbds]
print(lmbds)
print(mses)
plt.plot(lmbds, mses)
```

```
plt.xlabel('lambda')
plt.ylabel('MSE')
plt.show()
```

(a) 对于线性回归模型, 请直接计算测试集上的 MSE;

(b) 对于岭回归问题, 请考察不同正则项权重 λ 的取值范围, 并观察训练集 MSE、测试集 MSE 和 λ 的取值的关系, 总结变化的规律;

除示例代码中使用到的 sklearn 库函数外, 不能使用其他的 sklearn 函数, 需要基于 numpy 实现线性回归模型和 MSE 的计算.

解:

1. 令 $\boldsymbol{E} = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} + \mathbf{1}b - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2$

   对 $\boldsymbol{w}$ 求导得

$$\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{w}} = \boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{w} + \mathbf{1}b - \boldsymbol{y}) + 2\lambda\boldsymbol{w}$$

   对 b 求导得

$$\frac{\partial \boldsymbol{E}}{\partial b} = \mathbf{1}^\top(\boldsymbol{X}\boldsymbol{w} + \mathbf{1}b - \boldsymbol{y})$$

   当 $(\boldsymbol{X}^\top\boldsymbol{X} + 2\lambda\boldsymbol{I}_d)$ 是满秩矩阵时, 令该式等于零即可得

$$\boldsymbol{w}^*_{\text{Ridge}} = (\boldsymbol{X}^\top\boldsymbol{X} + 2\lambda\boldsymbol{I}_d)^{-1}\boldsymbol{X}^\top(\boldsymbol{y} - \mathbf{1}b^*_{\text{Ridge}})$$

   令 $\boldsymbol{T} = (\boldsymbol{X}^\top\boldsymbol{X} + 2\lambda\boldsymbol{I}_d)^{-1}$, 则

$$\boldsymbol{w}^*_{\text{Ridge}} = \boldsymbol{T}\boldsymbol{X}^\top(\boldsymbol{y} - \mathbf{1}b^*_{\text{Ridge}})$$

   代入有

$$\begin{aligned} b^*_{\text{Ridge}} &= \mathbf{1}^\top\boldsymbol{y} - \mathbf{1}^\top\boldsymbol{X}\boldsymbol{w}^*_{\text{Ridge}} \\ &= \mathbf{1}^\top\boldsymbol{y} - \mathbf{1}^\top\boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top(\boldsymbol{y} - \mathbf{1}b^*_{\text{Ridge}}) \\ &= \mathbf{1}^\top\boldsymbol{y} - \mathbf{1}^\top\boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top\boldsymbol{y} + \mathbf{1}^\top\boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top\mathbf{1}b^*_{\text{Ridge}} \end{aligned}$$

最后有

$$b^*_{\textbf{Ridge}} = \frac{\mathbf{1}^\top \boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top \boldsymbol{y} - \mathbf{1}^\top \boldsymbol{y}}{\mathbf{1}^\top \boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top \mathbf{1} - 1}$$

最后将 $b^*_{\textbf{Ridge}}$ 带回即可求出

$$\boldsymbol{w}^*_{\textbf{Ridge}} = \boldsymbol{T}\boldsymbol{X}^\top (\boldsymbol{y} - \mathbf{1}\frac{\mathbf{1}^\top \boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top \boldsymbol{y} - \mathbf{1}^\top \boldsymbol{y}}{\mathbf{1}^\top \boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^\top \mathbf{1} - 1})$$

加上第二题的结果, 可以看出, 使用矩阵 $\boldsymbol{T}$ 抽象之后, 最优解和原始最优解的形式是一样的, 不同的是 $\boldsymbol{T}$ 的值, 二者的矩阵 $\boldsymbol{T}$ 括号内相差了一个 $2\lambda \boldsymbol{I}_d$ 项.

所以我们可以认为 $\boldsymbol{w}^*_{\textbf{LS}}$ 和 $b^*_{\textbf{LS}}$ 是 $\lambda = 0$ 的 $\boldsymbol{w}^*_{\textbf{Ridge}}$ 和 $b^*_{\textbf{Ridge}}$ 特殊情况.

2. 将恒等式

$$\boldsymbol{X} = \boldsymbol{X}$$

右端乘上 $(\boldsymbol{X}^\top \boldsymbol{X} + \lambda I_d)(\boldsymbol{X}^\top \boldsymbol{X} + \lambda I_d)^{-1}$ 即 I_d 则有

$$\boldsymbol{X} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \lambda I_d)(\boldsymbol{X}^\top \boldsymbol{X} + \lambda I_d)^{-1}$$

将 $\boldsymbol{X}$ 乘入括号内则有

$$\boldsymbol{X} = (\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{X})(\boldsymbol{X}^\top \boldsymbol{X} + \lambda I_d)^{-1}$$

再将 $\boldsymbol{X}$ 提出至右侧则有

$$\boldsymbol{X} = (\boldsymbol{X}\boldsymbol{X}^\top + \lambda I_m)\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \lambda I_d)^{-1}$$

最后则有

$$(\boldsymbol{X}\boldsymbol{X}^\top + \lambda I_m)^{-1}\boldsymbol{X} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \lambda I_d)^{-1}$$

式子成立.

这个结论能够帮助岭回归的计算.

当样例的维度 d 大于样例数目 m 的时候, 将 $\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X}+2\lambda I_d)^{-1}$ 转为 $(\boldsymbol{X}\boldsymbol{X}^\top+2\lambda I_m)^{-1}\boldsymbol{X}$ 能够将矩阵求逆求 $\boldsymbol{T}$ 的矩阵维度减少, 进而加快矩阵求逆的速度.

3. (a)

算出线性回归模型测试集上的 MSE 结果为 20.724023437336253.

(b)

取不同的权重所得结果为

λ	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
MSE	20.72	20.92	21.09	21.21	21.31	21.39	21.4	21.50	21.54	21.57	21.60

Table 1: lambda

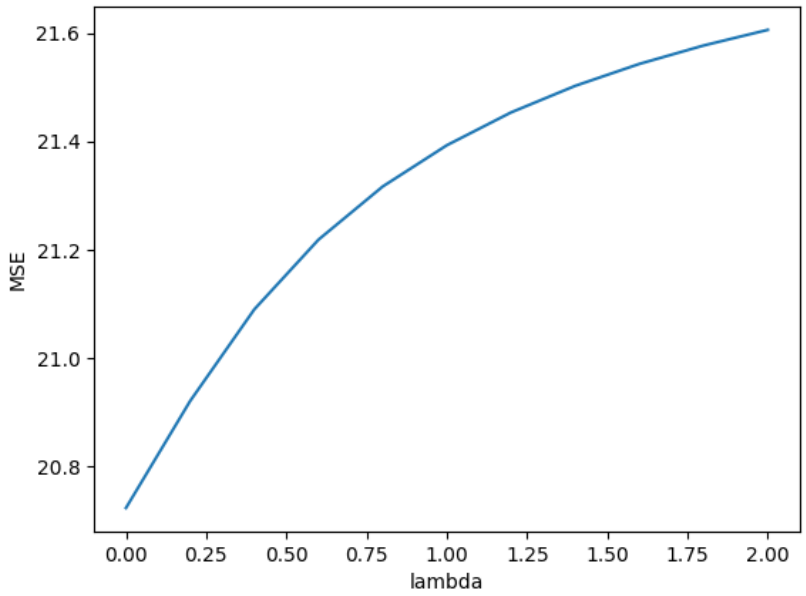

Figure 1: P-R 曲线

可以看出, MSE 随着 λ 的增大而缓慢增大, 所以我们可以选择一个相对较小但仍然为正数的 λ 值.

四. (20 points) **线性判别分析**

教材 3.4 节介绍了 “线性判别分析” 模型 LDA (Linear Discriminative Analysis), 本题首先针对 LDA 从分布假设的角度进行推导和分析. 考虑 N 分类问题, 训练集 $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$, 其中, 第 n 类样例从高斯分布 $\mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ 中独立同分布采样得到 (其中, $n = 1, 2, \cdots, N$). 记该类样例数量为 m_n. 类别先验为 $p(y = n) = \pi_n$, 反映了各类别出现的概率. 若 $\boldsymbol{x} \in \mathbb{R}^d \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, 则其概率密度函数为

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right). \tag{5}$$

假设不同类别的条件概率为高斯分布, 当不同类别的协方差矩阵 $\boldsymbol{\Sigma}_n$ 相同时, 对于类别的预测转化为类别中心之间的线性问题, 下面对这一模型进行进一步分析. 假设 $\boldsymbol{\Sigma}_n = \boldsymbol{\Sigma}$, 分析 LDA 的分类方式以及参数估计步骤.

1. 样例 $\boldsymbol{x}$ 的后验概率 $p(y = n \mid \boldsymbol{x})$ 表示了样例属于第 n 类的可能性, 当计算样例针对 N 个类别的后验概率后, 找出后验概率最大的类别对样例的标记进行预测, 即 $\arg\max_n p(y = n \mid \boldsymbol{x})$. 等价于考察 $\ln p(y = n \mid \boldsymbol{x})$ 的大小, 请证明在此假设下,

$$\arg\max_y p(y \mid \boldsymbol{x}) = \arg\max_n \underbrace{\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_n - \frac{1}{2} \boldsymbol{\mu}_n^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_n + \ln \pi_n}_{\delta_n(\boldsymbol{x})}. \tag{6}$$

   其中 $\delta_n(\boldsymbol{x})$ 为 LDA 在分类时的判别函数.

2. 在 LDA 模型中, 需要估计各类别的先验概率, 以及条件概率中高斯分布的参数. 针对二分类问题 ($N = 2$), 使用如下方式估计类别先验、均值与协方差矩阵:

$$\hat{\pi}_n = \frac{m_n}{m}; \quad \hat{\boldsymbol{\mu}}_n = \frac{1}{m_n} \sum_{y_i = n} \boldsymbol{x}_i, \tag{7}$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{m - N} \sum_{n=1}^{N} \sum_{y_i = n} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_n)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_n)^\top. \tag{8}$$

   LDA 使用这些经验量替代真实参数, 计算判别式 $\delta_n(\boldsymbol{x})$ 并按照第**??**问中的准则做出预测. 请证明:

$$\boldsymbol{x}^\top \hat{\boldsymbol{\Sigma}}^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) > \frac{1}{2} (\hat{\boldsymbol{\mu}}_2 + \hat{\boldsymbol{\mu}}_1)^\top \hat{\boldsymbol{\Sigma}}^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) - \ln(m_2/m_1) \tag{9}$$

时 LDA 将样例预测为第 2 类. 请分析这一判别方式的几何意义.

3. 在 LDA 中, 对样例 $\boldsymbol{x}$ 的判别可视为在投影的空间中和某个阈值进行比较. 上述推导通过最大后验概率的方法得到对投影后样例分布的需求, 而 Fisher 判别分析 (Fisher Discriminant Analysis, FDA) 也是一种常见的线性判别分析方法, 直接对样例投影后数据的分布情况进行约束. FDA 一般通过广义瑞利商进行求解, 请基于教材 3.4 节对“线性判别分析” 的介绍, 对广义瑞利商的性质进行分析, 探讨 FDA 多分类推广的性质. 下面请说明对于 N 类分类问题, FDA 投影的维度最多为 $N-1$, 即投影矩阵 $\boldsymbol{W} \in \mathbb{R}^{d\times(N-1)}$.
提示: 矩阵的秩具有如下性质: 对于矩阵 $\mathbf{A} \in \mathbb{R}^{m\times n}$, 矩阵 $\mathbf{B} \in \mathbb{R}^{n\times r}$, 则

$$\operatorname{rank}(\mathbf{A})+\operatorname{rank}(\mathbf{B})-n \leq \operatorname{rank}(\mathbf{AB}) \leq \min\{\operatorname{rank}(\mathbf{A}), \operatorname{rank}(\mathbf{B})\} . \quad (10)$$

对于任意矩阵 $\boldsymbol{A}$, 以下公式成立

$$\operatorname{rank}(\boldsymbol{A}) = \operatorname{rank}\left(\mathbf{A}^{\top}\right) = \operatorname{rank}\left(\mathbf{A}\mathbf{A}^{\top}\right) = \operatorname{rank}\left(\mathbf{A}^{\top}\mathbf{A}\right) . \quad (11)$$

解:

1. 由贝叶斯公式可知

$$\begin{aligned}
& p(y=n|\boldsymbol{x}) \\
=& \frac{p(y=n)p(\boldsymbol{x}|y=n)}{p(\boldsymbol{x})} \\
=& \frac{\pi_n((2\pi)^{\frac{d}{2}}\det(\boldsymbol{\Sigma})^{\frac{1}{2}})^{-1}\exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_n)^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_n))}{p(\boldsymbol{x})} \\
=& \frac{\pi_n((2\pi)^{\frac{d}{2}}\det(\boldsymbol{\Sigma})^{\frac{1}{2}})^{-1}\exp(\boldsymbol{x}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_n-\frac{1}{2}\boldsymbol{\mu}_n^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_n-\frac{1}{2}\boldsymbol{x}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{x})}{p(\boldsymbol{x})}
\end{aligned}$$

并且 $\arg\max_n p(y=m|\boldsymbol{x})$ 可以转化为

$$\arg\max_n \ln p(y=m|\boldsymbol{x})$$

即有

$$\begin{aligned}\arg\max_n \ln \pi_n + \ln((2\pi)^{\frac{d}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}})^{-1} + \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_n \\ - \frac{1}{2} \boldsymbol{\mu}_n^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_n - \frac{1}{2} \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - \ln p(\boldsymbol{x})\end{aligned}$$

去除与 n 无关的项即可得

$$\arg\max_n \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_n - \frac{1}{2} \boldsymbol{\mu}_n^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_n + \ln \pi_n$$

2. 将式子

$$\boldsymbol{x}^\top \hat{\boldsymbol{\Sigma}}^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) > \frac{1}{2} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^\top \hat{\boldsymbol{\Sigma}}^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) - \ln \frac{m_2}{m_1}$$

拆开可得

$$\boldsymbol{x}^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_2 - \boldsymbol{x}^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_1 > \frac{1}{2} \hat{\boldsymbol{\mu}}_2^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_2 - \frac{1}{2} \hat{\boldsymbol{\mu}}_1^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_1 - \ln \frac{m_2}{m_1}$$

进行移项

$$\boldsymbol{x}^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_2 - \frac{1}{2} \hat{\boldsymbol{\mu}}_2^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_2 + \ln \frac{m_2}{m} > \boldsymbol{x}^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_1 - \frac{1}{2} \hat{\boldsymbol{\mu}}_1^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_1 + \ln \frac{m_1}{m}$$

将 $\ln \frac{m_n}{m}$ 替换为 $\ln \hat{\pi}_n$ 可得

$$\boldsymbol{x}^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_2 - \frac{1}{2} \hat{\boldsymbol{\mu}}_2^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_2 + \ln \hat{\pi}_2 > \boldsymbol{x}^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_1 - \frac{1}{2} \hat{\boldsymbol{\mu}}_1^\top \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_1 + \ln \hat{\pi}_1$$

即有 $n^* = \arg\max_n \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_n - \frac{1}{2} \boldsymbol{\mu}_n^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_n + \ln \pi_n = 2$

所以此时 LDA 将样例预测为第 2 类.

几何意义:

$\frac{1}{2}(\hat{\boldsymbol{\mu}}_2 + \hat{\boldsymbol{\mu}}_1)$ 是 $\hat{\boldsymbol{\mu}}_1$ 和 $\hat{\boldsymbol{\mu}}_2$ 这两个类别中心的中点.

而 $x^\top \hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)$ 可以化为 $(\hat{\boldsymbol{\Sigma}}^{-\frac{1}{2}}x)^\top(\hat{\boldsymbol{\Sigma}}^{-\frac{1}{2}}(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1))$

即将特殊的多维正态分布缩放旋转成了标准多维正态分布, 然后衡量离第 1 类的类别中心 $\hat{\boldsymbol{\mu}}_1$ 的远近.

若大于号成立, 说明 $\boldsymbol{x}$ 比中点 $\frac{1}{2}(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2)$ 离 $\hat{\boldsymbol{\mu}}_1$ 更远, 也就是离 $\hat{\boldsymbol{\mu}}_2$ 更近.

但是还要考虑一个因素, 如果第 2 类的样例比第 1 类的样例数目多, 就要加入一定的偏好, 比如这里的对数几率 $\ln(m_2/m_1)$, 当 $m_2 > m_1$ 时是正数, 就能更容易判断为第 2 类.

3. 求解

$$\max_{\boldsymbol{W}} \frac{\mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{S}_b \boldsymbol{W})}{\mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{S}_w \boldsymbol{W})}$$

最后可以转化为广义特征值问题求解

$$\boldsymbol{S}_b \boldsymbol{W} = \lambda \boldsymbol{S}_w \boldsymbol{W}$$

即求解

$$\boldsymbol{S}_w^{-1} \boldsymbol{S}_b \boldsymbol{W} = \lambda \boldsymbol{W}$$

也就是求 $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ 特征值最大的前几个的特征向量.

而我们知道

$$\boldsymbol{S}_b = \sum_{i=1}^{N} m_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^\top$$

其中我们知道

$$\sum_{i=1}^{N} m_i(\boldsymbol{\mu}_i - \boldsymbol{\mu}) = \sum_{i=1}^{N} m_i\boldsymbol{\mu}_i - \boldsymbol{\mu}\sum_{i=1}^{N} m_i = \mathbf{0}$$

也就是

$$\boldsymbol{\mu}_N - \boldsymbol{\mu} = -\frac{1}{m_N}\sum_{i=1}^{N-1} m_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})$$

即 $(\boldsymbol{\mu}_N - \boldsymbol{\mu})$ 可以由 $(\boldsymbol{\mu}_i - \boldsymbol{\mu})$ 线性表示出来.
因此我们有

$$\begin{aligned}
\operatorname{rank}(\boldsymbol{S}_w^{-1}\boldsymbol{S}_b) &\leq \min\{\operatorname{rank}(\boldsymbol{S}_w^{-1}), \operatorname{rank}(\boldsymbol{S}_b)\} \\
&\leq \operatorname{rank}(\boldsymbol{S}_b) \\
&= \operatorname{rank}(\sum_{i=1}^{N} m_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^\top) \\
&= \operatorname{rank}(\sum_{i=1}^{N-1} m_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^\top) \\
&\leq \sum_{i=1}^{N-1} \operatorname{rank}((\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^\top) \\
&= \sum_{i=1}^{N-1} \operatorname{rank}(\boldsymbol{\mu}_i - \boldsymbol{\mu}) \\
&= N - 1
\end{aligned}$$

因此 $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ 最多有 $N-1$ 个非零特征值, 对应 $N-1$ 个特征向量. 而 $\boldsymbol{W}$ 的每个列向量都是 $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ 的线性无关的特征向量. 所以 FDA 投影的维度最多为 $N-1$.

五. (20 points) **多分类学习**

教材 3.5 节介绍了“多分类学习”的多种方式, 本题针对 OvO 和 OvR 两种多分类学习方法进行分析:

1. 分析两种多分类方法的优劣. 思考这两种多分类推广方式是否存在难以处理的情况?
2. 在 OvR 的每一个二分类子任务中, 目标类别作为正类, 而其余所有类别作为负类. 此时, 是否需要显式考虑正负类别的不平衡带来的影响?

解：

1. OvO 的优势: OvO 的每个分类器仅用到两个类的样例, 在类别很多的时候, OvO 的训练时间开销通常比 OvR 小.

   OvO 的劣势: OvO 需要训练 $N(N-1)/2$ 个分类器, 所以存储开销和训练时间开销通常比较大.

   OvR 的优势: OvR 只需要训练 N 个分类器.

   OvR 的劣势: 每次训练时都用到了所有的数据, 在类别很多的时候, 训练时间开销更大.

   这两种多分类推广方式都可能存在难以处理的情况.

   例如对 OvR 来说, 如果类别特别多的时候, 每次都要使用全部的数据来处理, 时间开销就接近 $O(n^2)$ 了, 几乎是不可接受的.

   即使是对于 OvO 来说, 我们也能构造出一个类似的例子, 例如有一半的样例都是属于第 0 类, 但是剩下的一半几乎每个样例都对应一个类别, 这样时间开销和空间开销也都到达了 $O(n^2)$ 了, 也几乎是不可接受的.

2. 对 OvR 来说, 由于对每一个类进行了相同的处理, 其拆解出来的二分类任务中类别不平衡的影响会相互抵消, 因此通常不需要显式考虑正负类别不平衡带来的影响.