姓名：张凯然
学号：201300009

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. 答案如下:

我们有必要继续研究机器学习算法.

NFL 定理的重要前提是针对所有目标函数的总误差, 但是在实际的学习问题中我们只关心一个或者一类目标函数. NFL 定理告诉我们, 脱离了具体问题谈论学习算法的性能是毫无意义的.

2. 答案如下:

我们不妨假设新的性能度量函数 ℓ 是对于原来函数的重映射, 具体的

$$\mathbb{I}[h(x)\neq f(x)]=1\rightarrow\ell_{false}$$
$$\mathbb{I}[h(x)=f(x)]=0\rightarrow\ell_{true}$$

那么根据每个目标函数 f 的出现概率相等, 我们很容易得到如下

结果:

$$\begin{aligned}
\sum_f E_{ote}(\mathfrak{L}_a|X,f) &= \sum_f \cdot \sum_h \cdot \sum_{x\in\mathcal{X}-X} P(\boldsymbol{x})\ell(h(\boldsymbol{x}),f(\boldsymbol{x}))P(h|\mathcal{X},\mathfrak{L}_a) \\
&= \sum_{x\in\mathcal{X}-X} P(\boldsymbol{x}) \cdot \sum_h P(h|\mathcal{X},\mathfrak{L}_a) \cdot \sum_f \ell(h(\boldsymbol{x}),f(\boldsymbol{x})) \\
&= \sum_{x\in\mathcal{X}-X} P(\boldsymbol{x}) \cdot \sum_h P(h|\mathcal{X},\mathfrak{L}_a) \cdot \left(\frac{1}{2}2^{|\mathcal{X}|}\ell_{true} + \frac{1}{2}2^{|\mathcal{X}|}\ell_{false}\right) \\
&= \frac{1}{2}2^{|\mathcal{X}|}\left(\ell_{true}+\ell_{false}\right) \cdot \sum_{x\in\mathcal{X}-X} P(\boldsymbol{x}) \cdot \sum_h P(h|\mathcal{X},\mathfrak{L}_a) \\
&= \frac{1}{2}2^{|\mathcal{X}|}\left(\ell_{true}+\ell_{false}\right) \cdot \sum_{x\in\mathcal{X}-X} P(\boldsymbol{x}) \cdot 1
\end{aligned}$$

这个结果与学习算法 $\mathfrak{L}_a$ 本身无关, 因而完成了证明

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:

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

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 $y = [y_1; \ldots, y_m] \in \mathbb{R}^m$, 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:

[1] 公式 2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.

答案如下: 为了方便下述推导, 重新定义矩阵 $\boldsymbol{X}$

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^\top & 1 \\ \boldsymbol{x}_2^\top & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_m^\top & 1 \end{pmatrix}$$

将优化目标向量 $(\boldsymbol{w}, b)$ 合并为 $\hat{\boldsymbol{w}} = (\boldsymbol{w}, b)$, 于是原问题可以写为

$$\hat{\boldsymbol{w}}^* = \arg\min_{\hat{\boldsymbol{w}}} \left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}}\right)^\top \left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}}\right)$$

令 $E_{\hat{\boldsymbol{w}}} = \left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}}\right)^\top \left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}}\right)$, 对于 $\hat{\boldsymbol{w}}$ 求导得到

$$\frac{\partial E_{\hat{\boldsymbol{w}}}}{\partial \hat{\boldsymbol{w}}} = 2\ \boldsymbol{X}^\top \left(\boldsymbol{X}\hat{\boldsymbol{w}} - \boldsymbol{y}\right)$$

令上式为零可以得到 $\hat{\boldsymbol{w}}$ 的闭式解.
如果矩阵 $\boldsymbol{X}^\top\boldsymbol{X}$ 是满秩矩阵或者正定矩阵, 则可以得到

$$\hat{\boldsymbol{w}}^* = \left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top y$$

如果矩阵 $\boldsymbol{X}^\top\boldsymbol{X}$ 不满秩, 则可以得到多个最优解, 根据学习算法的归纳偏好可以得到最后的的结果.

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:

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

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(3)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

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

# linear regression
def linReg(X_train:np.ndarray, y_train:np.ndarray) -> np.ndarray:
    pass

def linRegMSE(X_train:np.ndarray, y_train:np.ndarray, X_test:np.ndarray, y_test:np.
    ndarray) -> float:
    pass
reportLinRegMSE= lambda : linRegMSE(trainx,trainy,testx,testy)

# ridge regression
def ridgeReg(X_train:np.ndarray, y_train:np.ndarray, lmbd:float) -> np.ndarray:
    pass

def ridgeRegMSE(X_train:np.ndarray, y_train:np.ndarray, X_test:np.ndarray, y_test:np
    .ndarray, lmbd:float) -> float:
    pass
reportRidgeRegMSE= lambda lmbd : ridgeRegMSE(trainx,trainy,testx,testy,lmbd)
```

(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. 答案如下: 岭回归问题的优化目标可以写作:

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

令右式为 E , 将右式对于 $\boldsymbol{w}, b$ 求导得到

$$\frac{\partial E}{\partial \boldsymbol{w}} = \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w} + \boldsymbol{X}^\top (\mathbf{1}b - \mathbf{y}) + 2\lambda\boldsymbol{w} \tag{5}$$

$$\frac{\partial E}{\partial b} = \mathbf{1}^\top \boldsymbol{X}\boldsymbol{w} + \mathbf{1}^\top \mathbf{1}b - \mathbf{1}^\top \mathbf{y} \tag{6}$$

根据式 5 可以得到, 如果矩阵 $\boldsymbol{X}^\top\boldsymbol{X} + 2\lambda\boldsymbol{I}_d$ 可逆, 则得到

$$\boldsymbol{w}^*_{\mathbf{Ridge}} = \left(\boldsymbol{X}^\top\boldsymbol{X} + 2\lambda\boldsymbol{I}_d\right)^{-1} \boldsymbol{X}^\top (\mathbf{y} - \mathbf{1}b) \tag{7}$$

将该式带入式 6 可以解得

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

将 b 的结果带入式 7 即可得到 $\boldsymbol{w}^*_{\mathbf{Ridge}}$.

$$\boldsymbol{w}^*_{\mathbf{Ridge}} =$$

$$\left(\boldsymbol{X}^\top\boldsymbol{X} + 2\lambda\boldsymbol{I}_d\right)^{-1}\boldsymbol{X}^\top \left(\mathbf{y} - \mathbf{1}\frac{\mathbf{1}^\top \left(\boldsymbol{X}\left(\boldsymbol{X}^\top\boldsymbol{X} + 2\lambda\boldsymbol{I}_d\right)^{-1}\boldsymbol{X}^\top - \boldsymbol{I}_m\right)\mathbf{y}}{\mathbf{1}^\top \left(\boldsymbol{X}\left(\boldsymbol{X}^\top\boldsymbol{X} + 2\lambda\boldsymbol{I}_d\right)^{-1}\boldsymbol{X}^\top - \boldsymbol{I}_m\right)\mathbf{1}}\right)$$

类似第二题, 我们可以得到原始线性回归的最优解为

$$(\boldsymbol{w}^*_{\mathbf{LS}}, b^*_{\mathbf{LS}}) = \hat{\boldsymbol{w}}^* = \left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top y$$

对比两个闭式解, 区别在于求逆矩阵不同, 岭回归问题需要求解 $\left(\boldsymbol{X}^\top\boldsymbol{X} + 2\lambda\boldsymbol{I}_d\right)^{-1}$, 这一定是一个满秩矩阵, 但是线性回归问题需要求解 $\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}$, 这不一定是一个满秩矩阵.

2. 答案如下: 首先证明式 4 成立

Proof.

$$\begin{aligned} & \left(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}_m\right)^{-1} \boldsymbol{X} = \boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_d\right)^{-1} \\ \Longleftrightarrow\ & \boldsymbol{I}_m \cdot \boldsymbol{X} \cdot \left(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_d\right) = \left(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}_m\right) \cdot \boldsymbol{X} \cdot \boldsymbol{I}_d \\ \Longleftrightarrow\ & \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{X} = \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{X} \end{aligned}$$

□

这个等式的意义在于转变需要求逆的矩阵, 矩阵 $\left(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}_m\right)$ 的秩为 m, 然而矩阵 $\left(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_d\right)$ 的秩为 d, 这使得我们可以选取一个秩小的矩阵求逆, 从而降低计算开销. 当特征维度数大于样本数 $(d > m)$ 时, 可以求解 $\left(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}_m\right)$ 的逆, 当样本数特别多时 $(m > d)$, 则可以求解 $\left(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_d\right)$ 的逆.

3. 答案如下:

(a) 线性回归模型在测试集上的 MSE 为 20.72402343734099

(b) 岭回归问题的 MSE 如下表

λ	MSE in Training Set	MSE in Test set
0.000000	22.985016	20.724023
0.000100	22.985016	20.724112
0.001000	22.985019	20.724914
0.010000	22.985316	20.733068
0.100000	23.006671	20.822143
1.000000	23.334838	21.393405
10.000000	23.948194	21.947534
100.000000	26.449625	23.831913
1000.000000	32.094287	28.831454

可以看到, 随着 λ 的增大, 训练集和测试集的 MSE 都在增大.

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

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right). \tag{8}$$

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,

$$\arg\max_y p(y \mid \boldsymbol{x}) = \arg\max_n \underbrace{\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_n - \frac{1}{2}\boldsymbol{\mu}_n^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_n + \ln \pi_n}_{\delta_n(\boldsymbol{x})}. \tag{9}$$

   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:

$$\hat{\pi}_n = \frac{m_n}{m}; \quad \hat{\boldsymbol{\mu}}_n = \frac{1}{m_n}\sum_{y_i = n} \boldsymbol{x}_i, \tag{10}$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{m - N}\sum_{n=1}^{N}\sum_{y_i = n} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_n)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_n)^\top. \tag{11}$$

   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1问中的准则做出预测. 请证明:

$$\boldsymbol{x}^\top \hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) > \frac{1}{2}(\hat{\boldsymbol{\mu}}_2 + \hat{\boldsymbol{\mu}}_1)^\top \hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) - \ln(m_2/m_1) \tag{12}$$

   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

$$\operatorname{rank}(\mathbf{A})+\operatorname{rank}(\mathbf{B})-n \leq \operatorname{rank}(\mathbf{AB}) \leq \min\{\operatorname{rank}(\mathbf{A}), \operatorname{rank}(\mathbf{B})\} . \quad (13)$$

对于任意矩阵 $\boldsymbol{A}$, 以下公式成立

$$\operatorname{rank}(\boldsymbol{A}) = \operatorname{rank}\left(\mathbf{A}^{\top}\right) = \operatorname{rank}\left(\mathbf{A}\mathbf{A}^{\top}\right) = \operatorname{rank}\left(\mathbf{A}^{\top}\mathbf{A}\right) . \quad (14)$$

解:

1. 答案如下: 这是一行字.

2. 答案如下: 这是一行字.

3. 答案如下: 这是一行字.

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. 答案如下:

OvO 优点 训练每一个二分类器时, 只使用两类的样例, 因而在类别很多时, 时间开销更小.

OvO 缺点 训练出的二分类器数量较多 ($N(N-1)/2$ 个) , 因而存储模型使用的空间开销和测试时间更大.

OvR 优点 训练出的二分类器数量较少 (N 个) , 节省了模型使用的空间开销和测试时间.

OvR 缺点 训练每一个二分类器都要用到全部样例, 在类别很多的情况下时间开销大.

OvO 难以处理的情况 OvO 策略的集成方法是投票, 默认认为每一个二分类器给出的结果拥有相同的权重, 这只能适用于数据集中每一个类的样例数目相近的情况, 如果类之间的样例数目差距很大, 那么这种策略就难以给出精确的结果.

OvR 难以处理的情况

2. 答案如下: 不需要

对于每一个类进行了相似的划分, 获得的二分类任务中类别不均衡的影响会相互抵消, 因而不需要专门处理.