机器学习导论
作业二

171860607, 白晋斌, 810594956@qq.com

2020 年 4 月 2 日

1 [15 pts] Linear Regression

给定数据集 $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$，最小二乘法试图学得一个线性函数 $y = \mathbf{w}^*\mathbf{x} + b^*$ 使得残差的平方和最小化，即

$$(\mathbf{w}^*, b^*) = \arg\min_{\mathbf{w},b} \sum_{i=1}^{m} [y_i - (\mathbf{w}\mathbf{x_i} + b)]^2. \tag{1.1}$$

“最小化残差的平方和”与“最小化数据集到线性模型的欧氏距离之和”或是“最小化数据集到线性模型的欧氏距离的平方和”一致吗？考虑下述例子

$$D = \{(-1, 0), (0, 0), (1, 1)\}, \tag{1.2}$$

并回答下列问题。

(1) **[5 pts]** 给出“最小化残差的平方和”在该例子中的解 (w^*, b^*)。

(2) **[5 pts]** 给出“最小化数据集到线性模型的欧氏距离的平方和”在该例子中的数学表达式，并给出其解 (w_E, b_E)，该解与 (w^*, b^*) 一致吗？

(3) **[5 pts]** 给出“最小化数据集到线性模型的欧氏距离之和”在该例子中的数学表达式，(w^*, b^*) 是该问题的解吗？

Solution. *此处用于写解答 (中英文均可)*

(1) 我们可将 $\sum_{i=1}^m [y_i - (\mathbf{wx_i} + b)]^2$ *分别对* $\mathbf{w}$ *和* b *求导, 得到*

$$\frac{\partial \sum_{i=1}^m [y_i - (\mathbf{wx_i} + b)]^2}{\partial \mathbf{w}} = 2\left(\mathbf{w}\sum_{i=1}^m x_i^2 - \sum_{i=1}^m (y_i - b)x_i\right) \tag{1.3}$$

$$\frac{\partial \sum_{i=1}^m [y_i - (\mathbf{wx_i} + b)]^2}{\partial b} = 2\left(mb - \sum_{i=1}^m (y_i - wx_i)\right) \tag{1.4}$$

然后令 (1.3) 和 (1.4) 为零, 可以得到 $\mathbf{w}$ *和* b *最优解的闭式解*

$$\mathbf{w} = \frac{\sum_{i=1}^m y_i(x_i - \bar{x})}{\sum_{i=1}^m x_i^2 - \frac{1}{m}\left(\sum_{i=1}^m x_i\right)^2} \tag{1.5}$$

$$b = \frac{1}{m}\sum_{i=1}^{m}(y_i - \mathbf{w}x_i) \tag{1.6}$$

其中 $\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$ 为 x 的均值. 将式 *(1.2)* 分别代入式 *(1.5)* 和 *(1.6)*, 可得

$$\mathbf{w} = \frac{1}{2} \tag{1.7}$$

$$b = \frac{1}{3} \tag{1.8}$$

此时取得最小值 $\frac{1}{6}$, 因此

$$(\mathbf{w}^*, b^*) = (\frac{1}{2}, \frac{1}{3}) \tag{1.9}$$

(2) 表达式为

$$(\mathbf{w_e}, b_e) = \arg\min_{\mathbf{w},b} \sum_{i=1}^{m} \left(\frac{|\mathbf{w}x_i - y_i + b|}{\sqrt{\mathbf{w}^2 + 1^2}} \right)^2. \tag{1.10}$$

当 $\mathbf{w} = \frac{\sqrt{13}-2}{3}$ 且 $b = \frac{1}{3}$ 时, 取到最小值 $\frac{4-\sqrt{13}}{3}$, 因此

$$(\mathbf{w_E}, b_E) = (\frac{\sqrt{13}-2}{3}, \frac{1}{3}) \tag{1.11}$$

显然, 该解与 (w^*, b^*) 不一致.

(3) 表达式为

$$(\mathbf{w_e}, b_e) = \arg\min_{\mathbf{w},b} \sum_{i=1}^{m} \frac{|\mathbf{w}x_i - y_i + b|}{\sqrt{\mathbf{w}^2 + 1^2}}. \tag{1.12}$$

当 $\mathbf{w} = \frac{1}{2}$ 且 $b = \frac{1}{2}$ 时, 取到最小值 $\frac{\sqrt{5}}{5}$, 因此

$$(\mathbf{w_e}, b_e) = (\frac{1}{2}, \frac{1}{2}) \tag{1.13}$$

将式 *(1.2)*、*(1.9)* 代入 $\sum_{i=1}^{m} |y_i - (\mathbf{w}x_i + b)|$, 可得

$$\sum_{i=1}^{m} \frac{|\mathbf{w}x_i - y_i + b|}{\sqrt{\mathbf{w}^2 + 1^2}} = \frac{4\sqrt{5}}{15} > \frac{\sqrt{5}}{5} \tag{1.14}$$

显然,(w^*, b^*) 不是该问题的一个解.

2 [40+5 pts] 编程题, Logistic Regression

请结合编程题指南进行理解

试考虑对率回归与线性回归的关系。最简单的对率回归的所要学习的任务仅是根据训练数据学得一个 $\beta = (\omega; b)$，而学习 β 的方式将有下述两种不同的实现：

0. [闭式解] 直接将分类标记作为回归目标做线性回归，其闭式解为

$$\beta = (\hat{X}^T\hat{X})^{-1}\hat{X}^T y \tag{2.1}$$

, 其中 $\hat{X} = (X; \vec{1})$

1. [数值方法] 利用牛顿法或梯度下降法解数值问题

$$\min_{\beta} \sum_{i=1}^{m} (-y_i \beta^T \hat{x_i} + \ln(1 + e^{\beta^T \hat{x_i}})). \tag{2.2}$$

得到 β 后两个算法的决策过程是一致的，即：

(1) $z = \beta X_i$

(2) $f = \frac{1}{1+e^{-z}}$

(3) 决策函数

$$y_i = \begin{cases} 1, & \text{if } f > \theta \\ 0, & \text{else} \end{cases} \tag{2.3}$$

其中 θ 为分类阈值。回答下列问题:

(1) **[10 pts]** 试实现用闭式解方法训练分类器。若设分类阈值 $\theta = 0.5$，此分类器在 Validation sets 下的准确率、查准率、查全率是多少?

(2) **[10 pts]** 利用所学知识选择合适的分类阈值，并输出闭式解方法训练所得分类器在 test sets 下的预测结果。

(3) **[10 pts]** 利用数值方法重新训练一个新的分类器。若设分类阈值 $\theta = 0.5$，此分类器在 Validation sets 下的准确率、查准率、查全率是多少?

(4) **[10 pts]** 利用所学知识选择合适的分类阈值，并输出数值方法训练所得分类器在 test sets 下的预测结果。

(5) **[选做][Extra 5 pts]** 谈谈两种方法下分类阈值的变化对预测结果的影响，简要说明看法。

Solution. *此处用于写解答 (中英文均可)*

(1) 代码可见 171860607_0.py 文件, 若设分类阈值 $\theta = 0.5$, 此分类器在 Validation sets 下的准确率为 0.74、查准率为 0.667、查全率为 1.0.

(2) 代码可见 171860607_0.py 文件, 选用的分类阈值 $\theta = 0.52$, 程序输出的预测结果为:

1 1 0 0 0 0 1 0 1 1 1 1 0 0 0 1 0 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 1 1 0 1 1 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 0 1 1 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 0 1 0 0 1 1 1 0 0 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 0 1 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 0 1 0 1 1 1 1 0 1 0 0 1 1 0 0 1 0 0 0 0 1 1 1 0 1 1 0 0 1 0 0 0 1 0 0 1 1 0 0 1 1 1 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 1 1 0 1 0 1 0 1 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 1 1 0 0 0 1 0 1 0 0 1 1 1 0

(3) 代码可见 171860607_1.py 文件, 若设分类阈值 $\theta = 0.5$, 此分类器在 Validation sets 下的准确率为 1.0、查准率为 1.0、查全率为 1.0.

(4) 代码可见 171860607_1.py 文件, 选用的分类阈值 $\theta = 0.5$, 程序输出的预测结果为:

1 1 0 0 0 0 1 0 1 1 1 1 0 0 0 1 0 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 1 1 0 1 1 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 0 1 1 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 0 1 0 0 1 1 1 0 0 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 0 1 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 0 1 0 1 1 1 1 0 1 0 0 1 1 0 0 1 0 0 0 0 1 1 1 0 1 1 0 0 1 0 0 0 1 0 0 1 1 0 0 1 1 1 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 1 1 0 0 0 1 0 1 0 0 1 1 1 0

0 1 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 1 0 0
0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1
0 0 0 1 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 1 0 1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 1
1 0 1 0 1 0 0 0 1 0 1 1 1 0 1 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 1 0 0 1 0 0 0 1 0 0 1 1 0 0
1 1 1 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 1 1 0 1 0 1 0
1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 1 1 0 0 0 1 0 1 0 0 1 1 1 0

(5) 以 0.01 为分类阈值变化的单位, 经测试, 我们发现使用闭式解方法, 阈值在 0.52 到 0.71 之间 (包括首尾), 在评估集准确率可以到 100%. 使用数值方法, 阈值在 0.01 到 0.99 之间 (包括首尾), 在评估集准确率可以到 100%. 正是由于通过数值方法算出的 z 向量模长大于通过闭式解算出的 z 向量模长, 所以数值方法经 sigmoid 函数输出便越接近 0/1, 从而可以达到更高性能的阈值范围越广.

具体来说,Sigmoid 函数有个漂亮的 S 型, 当输入的值的取值离 $x = 0$ 越远, 函数的值会很快接近 0 或 1. 换句话说,z 向量模长越大, 经 sigmoid 函数输出的值越接近 0 或 1, 但是当 z 向量模的长很小时,sigmoid 函数输出的值便接近 $\frac{1}{2}$, 这时就需要我们手动调整分类阈值的大小, 当我们认为预测结果偏大时, 即 $|z| = |\beta X_i|$ 偏大时, 可以适当调高分类阈值, 反之则适当调低分类阈值, 从而得到较好的预测结果.

3 [10 pts] Linear Discriminant Analysis

在凸优化中，试考虑两个优化问题，如果第一个优化问题的解可以直接构造出第二个优化问题的解，第二个优化问题的解也可以直接构造出第一个优化问题的解，则我们称两个优化问题是等价的。基于此定义，试证明优化问题 **P1** 与优化问题 **P2** 是等价的。

$$\max_{\mathbf{w}} \frac{\mathbf{w}^\top S_b \mathbf{w}}{\mathbf{w}^\top S_w \mathbf{w}}. \tag{3.1}$$

$$\begin{aligned} \min_{\mathbf{w}} \quad & -\mathbf{w}^\top S_b \mathbf{w} \\ \text{s.t.} \quad & \mathbf{w}^\top S_w \mathbf{w} = 1. \end{aligned} \tag{3.2}$$

Solution. *此处用于写解答 (中英文均可)*

(1) 首先证明第一个优化问题的解可以直接构造出第二个优化问题的解.

不妨设第一个优化问题的解是 $\mathbf{w_{s1}}$, 令

$$M = \frac{\mathbf{w_{s1}}^\top S_b \mathbf{w_{s1}}}{\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}}}$$

取

$$\mathbf{w_{s2}} = \frac{\mathbf{w_{s1}}}{\sqrt{\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}}}}$$

特殊地, 当 $\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}} = 1$ 时, 我们可以直接认为

$$\mathbf{w_{s2}} = \mathbf{w_{s1}}$$

则

$$\mathbf{w_{s2}}^T S_w \mathbf{w_{s2}} = \left(\frac{\mathbf{w_{s1}}}{\sqrt{\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}}}}\right)^T S_w \left(\frac{\mathbf{w_{s1}}}{\sqrt{\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}}}}\right) = \frac{\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}}}{\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}}} = 1 \tag{3.3}$$

我们已成功构造出第二个优化问题的解, 接下来证明 $\mathbf{w_{s2}}$ 即第二个优化问题的解.
此时

$$\begin{aligned}-\mathbf{w_{s2}}^T S_b \mathbf{w_{s2}} &= -\left(\frac{\mathbf{w_{s1}}}{\sqrt{\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}}}}\right)^T S_b \left(\frac{\mathbf{w_{s1}}}{\sqrt{\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}}}}\right) \\ &= -\frac{\mathbf{w_{s1}}^\top S_b \mathbf{w_{s1}}}{\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}}} \\ &= -M\end{aligned}$$

任取 $\mathbf{w_{s3}}$, 使得 $\mathbf{w_{s3}}^T S_w \mathbf{w_{s3}} = \mathbf{1}$ 且 $\mathbf{w_{s3}} \neq \mathbf{w_{s2}}$, 即 $\mathbf{w_{s3}} \neq \frac{\mathbf{w_{s1}}}{\sqrt{\mathbf{w_{s1}}^T S_w \mathbf{w_{s1}}}}$, 显然若 $\mathbf{w_{s3}} = \mathbf{w_{s1}}$, 则 $\mathbf{w_{s1}}^T S_w \mathbf{w_{s1}} = 1$, 此时 $\mathbf{w_{s2}} = \mathbf{w_{s1}} = \mathbf{w_{s3}}$, 与 $\mathbf{w_{s3}} \neq \mathbf{w_{s2}}$ 矛盾, 所以同样地,$\mathbf{w_{s3}} \neq \mathbf{w_{s1}}$. 我们不妨设 $\mathbf{w_{s3}}$ 才是第二个优化问题的解, 则有

$$-\mathbf{w_{s3}}^T S_b \mathbf{w_{s3}} < -\mathbf{w_{s2}}^T S_b \mathbf{w_{s2}} = -M$$

即

$$\begin{aligned}&\mathbf{w_{s3}}^T S_b \mathbf{w_{s3}} > M \\ &\frac{\mathbf{w_{s3}}^T S_b \mathbf{w_{s3}}}{1} > M \\ &\frac{\mathbf{w_{s3}}^T S_b \mathbf{w_{s3}}}{\mathbf{w_{s3}}^T S_w \mathbf{w_{s3}}} > M \end{aligned} \tag{3.4}$$

式 *(3.4)* 显示对于优化问题 *P1*,$\mathbf{w_{s3}}$ 比 $\mathbf{w_{s1}}$ 更优, 这与我们之前的前提: 第一个优化问题的解是 w_{s1} 矛盾, 故假设不成立, 不存在这样的 $w_{s3} \neq w_{s2}$ 成为第二个优化问题的解. 换句话说,w_{s2} 即第二个优化问题的解.

(2) 然后证明第二个优化问题的解可以直接构造出第一个优化问题的解.
不妨设第二个优化问题的解是 $\mathbf{w_{S2}}$, 则

$$\mathbf{w_{s2}}^T S_w \mathbf{w_{s2}} = 1$$

令

$$M = -\mathbf{w_{s2}}^T S_b \mathbf{w_{s2}}$$

取

$$\mathbf{w_{s1}} = k\mathbf{w_{s2}}, (k = 1, 2, 3, ...)$$

特殊地, 当 $k = 1$ 时, 我们可以直接认为

$$\mathbf{w_{s1}} = \mathbf{w_{s2}}$$

我们已经成功构造出第一个优化问题的解, 接下来证明 $\mathbf{w_{s1}}$ 即第一个优化问题的解.
此时

$$\frac{\mathbf{w_{s1}}^\top S_b \mathbf{w_{s1}}}{\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}}} = \frac{k^2 \mathbf{w_{s2}}^\top S_b \mathbf{w_{s2}}}{k^2 \mathbf{w_{s2}}^\top S_w \mathbf{w_{s2}}} = \frac{-M}{1} = -M$$

任取 $\mathbf{w_{s3}}$, 使得 $\mathbf{w_{s3}} \neq \mathbf{w_{s1}}$, 即 $\mathbf{w_{s3}} \neq k\mathbf{w_{s2}}$, 即 $\mathbf{w_{s3}} \neq \mathbf{w_{s2}}$. 我们不妨设 $\mathbf{w_{s3}}$ 才是第一个优化问题的解, 则有

$$\frac{\mathbf{w_{s3}}^\top S_b \mathbf{w_{s3}}}{\mathbf{w_{s3}}^\top S_w \mathbf{w_{s3}}} > \frac{\mathbf{w_{s1}}^\top S_b \mathbf{w_{s1}}}{\mathbf{w_{s1}}^\top S_w \mathbf{w_{s1}}} = -M$$

取

$$\mathbf{w_{s4}} = \frac{\mathbf{w_{s3}}}{\sqrt{\mathbf{w_{s3}}^\top S_w \mathbf{w_{s3}}}}$$

则我们构造出了

$$\mathbf{w_{s4}}^T S_w \mathbf{w_{s4}} = 1$$

此时的

$$-\mathbf{w_{s4}}^T S_b \mathbf{w_{s4}} = -\left(\frac{\mathbf{w_{s3}}}{\sqrt{\mathbf{w_{s3}}^\top S_w \mathbf{w_{s3}}}}\right)^T S_b \left(\frac{\mathbf{w_{s3}}}{\sqrt{\mathbf{w_{s3}}^\top S_w \mathbf{w_{s3}}}}\right) = -\frac{\mathbf{w_{s3}}^\top S_b \mathbf{w_{s3}}}{\mathbf{w_{s3}}^\top S_w \mathbf{w_{s3}}} < M \tag{3.5}$$

式 *(3.5)* 显示对于优化问题 *P2*, $\mathbf{w_{s4}}$ 比 $\mathbf{w_{s2}}$ 更优, 这与我们之前的前提: 第二个优化问题的解是 $\mathbf{w_{s2}}$ 矛盾, 故假设不成立, 不存在这样的 $\mathbf{w_{s3}} \neq \mathbf{w_{s1}}$ 成为第一个优化问题的解, 换句话说,$\mathbf{w_{s1}}$ 即第一个优化问题的解.

得证.

4 [35 pts] Multiclass Learning

在处理多分类学习问题的时候，我们通常有两种处理思路：一是间接求解，利用一些基本策略 (OvO, OvR, MvM) 将多分类问题转换为二分类问题，进而利用二分类学习器进行求解。二是直接求解，将二分类学习器推广到多分类学习器。

4.1 问题转换

(1) **[5 pts]** 考虑如下多分类学习问题：假设样本数量为 n，类别数量为 C，二分类器对于大小为 m 的数据训练的时间复杂度为 $\mathcal{O}(m)$(比如利用最小二乘求解的线性模型) 时，试分别计算在 OvO、OvR 策略下训练的总时间复杂度。

(2) **[10 pts]** 当我们使用 MvM 处理多分类问题时，正、反类的构造必须有特殊的设计，一种最常用的技术为“纠错输出码”(ECOC)，根据阅读材料 (Error-Correcting Output Codes、Solving Multiclass Learning Problems via Error-Correcting Output Codes[1]；前者为简明版，后者为完整版) 回答下列问题：

1) 假设纠错码之间的最小海明距离为 n，请问该纠错码至少可以纠正几个分类器的错误? 对于图1所示的编码，请计算该纠错码的最小海明距离并分析当两个分类器出错时该编码的纠错情况。
2) 令码长为 8，类别数为 4，试给出海明距离意义下的最优 ECOC 编码，并简述构造思路。
3) 试简述好的纠错码应该满足什么条件? (请参考完整版阅读资料)
4) ECOC 编码能起到理想纠错作用的重要条件是：在每一位编码上出错的概率相当且独立，试分析多分类任务经 ECOC 编码后产生的二类分类器满足该条件的可能性及由此产生的影响。

(3) **[10 pts]** 使用 OvR 和 MvM 将多分类任务分解为二分类任务求解时，试论述为何无需专门这对类别不平衡进行处理。

Class	Code Word							
	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
c_0	0	0	0	0	1	1	1	1
c_1	0	0	1	1	0	0	1	1
c_2	0	1	0	1	0	1	0	1

图 1: 3 类 8 位编码

4.2 模型推广

[10 pts] 对数几率回归是一种简单的求解二分类问题的广义线性模型，试将其推广到多分类问题上，其中标记为 $y \in \{1, 2 \ldots, K\}$。

提示：考虑如下 $K-1$ 个对数几率

$$\ln \frac{p(y=1|\mathbf{x})}{p(y=K|\mathbf{x})}, \ln \frac{p(y=2|\mathbf{x})}{p(y=K|\mathbf{x})}, \cdots, \ln \frac{p(y=K-1|\mathbf{x})}{p(y=K|\mathbf{x})}$$

Solution. 此处用于写解答 (中英文均可)

1 (1) *OvO* 策略:n 个样本,C 种类别，每次挑出两种类别，两两结合，每种类别要与其他 *C-1* 种类别分别结合，也就是每种类别的样本要出现 *C-1* 次，所有样本一共出现了 $n(C-1)$ 次，所以 *OvO* 策略下训练的总时间复杂度为 $\mathcal{O}(n(C-1))$.

OvR 策略:C 种类别的样本，每次取一种作为一大类，将剩余的 *C-1* 种样本看做另一大类，这样每种类别的样本要出现 C 次，所有样本一共出现了 nC 次，所以 *OvR* 策略下训练的总时间复杂度为 $\mathcal{O}(nC)$.

(2) 1) 至少可以纠正 $\lfloor \frac{n-1}{2} \rfloor$ 个分类器的错误.

该纠错码的最小海明距离是 *4*，因为 $\lfloor \frac{4-1}{2} \rfloor = 1$，所以当两个分类器出错时，该编码器将无法纠错，例如编码 *00111111*，编码器无法判断是 f_2, f_3 还是 f_4, f_5 出错，故无法正确分类到 c_0 或 c_1.

2) 如表*1*所示.

基本思路是: 当需要对 k 个类别构造 *ECOC* 编码时，纠错码之间的最小海明距离要尽

表 1: 4 类 8 位编码

Class	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
c_0	1	1	1	1	1	1	1	1
c_1	0	0	0	0	1	1	1	1
c_2	0	0	1	1	0	0	1	1
c_3	0	1	0	1	0	1	0	1

可能的大. 每个代码的长度为 2^{k-1}. 不妨设第一类全部为 *1*，则第二类要想与第一类海明距离最大，则有 2^{k-2} 个位与第一类不同，不妨设 $[0, 2^{k-2})$ 位为 *0*,$[2^{k-2}, 2^{k-1})$ 位为 *1*.

第三类既要与第一类海明距离最大，又要与第二类海明距离最大，即要有 2^{k-2} 个位与第一类不同，2^{k-2} 个位与第二类不同. 不难想到 2^{k-2} 个与第一类不同的位中要有 2^{k-3} 个与第二类相同的位，2^{k-2} 个与第二类不同的位中要有 2^{k-3} 个与第一类相同的位，因此第三类可以是 $[0, 2^{k-3}), [2^{k-2}, 2^{k-3} + 2^{k-2})$ 位为 *0*，$[2^{k-3}, 2^{k-2}), [2^{k-3} + 2^{k-2}, 2^{k-1})$ 位为 *1*. 以此类推，即可得到海明距离意义下的最优 *ECOC* 编码.

3) *1.* 行分离：对于海明距离，每个 *ECOC* 编码序列应该与其它 *ECOC* 编码序列良好可分 *2.* 列分离：每一位的分类函数 f_i 应该与其他位的分类函数 f_j 无相关性，这可以通过要求第 i 列与其他每个列之间的海明距离较大以及第 i 列与其他每个列的补码之间的海明距离也较大来实现.

4) *1.* 每一位编码上出错的概率相当，也就是每一位的分类函数产生误差的概率相当，这实际上取决于类别直接的区分难度，课本中提到，“不同拆解方式所形成的两个类别子集的区分难度往往不同，即其导致的二分类问题的难度不同”，换句说话，很难做到每个二分类问题难度相当，也就是很难满足每一位编码上出错的概率相当.

2. 每一位编码上出错的概率独立，这取决于 *ECOC* 编码的构造，因为好的纠错码可以满足列分离的条件，即每一位的分类函数与其他位的分类函数无相关性，当类别越多时，满足这个列分离条件的可能性越大，当类别较少时，很难满足每一位编码上出错的概率独立.

3. 影响：显然，类别越多，我们越容易满足独立的条件，越难满足相当的条件，反之越容易满足相当的条件，越难满足独立的条件. 这需要我们通过实验或其他手段进行取舍，找到一个均衡，课本中也提到，“一个理论纠错性能很好，但导致的二分类问题较难的编码，与另一个理论性能差一些，但导致的二分类问题较简单的编码，最终产生的模型性能孰强孰弱很难说.”

(3) 对于 *OvR*,*MvM* 来说，由于对每个类进行了相同的处理，在汇总时，其拆解出的二分类任务中类别不平衡的影响会相互抵消，因此通常不需专门处理.

2 对于标记为 $y \in \{1, 2 \ldots, K\}$ 的多分类问题，我们把其中一个类别看成是主类别，将其它 *K−1* 个类别和我们所选择的主类别分别建立对数几率回归模型.

以 *y=1* 和 *y=K* 两种标记为例. 在对这两种标记进行二分类时，已知

$$P(y = 1|x) + P(y = K|x) = 1 \tag{4.1}$$

又已知对数几率回归函数

$$y = \frac{1}{1 + e^{-(w^T x + b)}} \tag{4.2}$$

代入 (4.1)，可得

$$\ln \frac{P(y = 1|x)}{P(y = K|x)} = \ln \frac{1 - P(y = K|x)}{P(y = K|x)} = \ln \frac{1 - \frac{1}{1+e^{-w_1^T x + b_1}}}{\frac{1}{1+e^{-w_1^T x + b_1}}} = \ln e^{w_1^T x + b_1} = w_1^T x + b_1 \tag{4.3}$$

同理，

$$\ln \frac{P(y = 2|x)}{P(y = K|x)} = w_2^T x + b_2$$

$$\ln \frac{P(y = 3|x)}{P(y = K|x)} = w_3^T x + b_3$$

$$\cdots$$

$$\ln \frac{P(y=K-1|x)}{P(y=K|x)} = w_{K-1}^T x + b_{K-1}$$

同时可以得到

$$P(y=1|x) = P(y=K|x)e^{w_1^T x+b_1}$$
$$P(y=2|x) = P(y=K|x)e^{w_2^T x+b_2}$$

$$\cdots$$

$$P(y=K-1|x) = P(y=K|x)e^{w_{K-1}^T x+b_{K-1}}$$

我们已知

$$\sum_{i=1}^{K} P(y=i|x) = 1 \tag{4.4}$$

故

$$P(y=K|x)\sum_{i=1}^{K-1} e^{w_i^T x+b_i} + P(y=K|x) = 1$$

$$P(y=K|x) = \frac{1}{1+\sum_{i=1}^{K-1} e^{w_i^T x+b_i}} \tag{4.5}$$

通过式 *(4.5)*, 其他标记的概率也可以计算出来

$$P(y=1|x) = \frac{e^{w_1^T x+b_1}}{1+\sum_{i=1}^{K-1} e^{w_i^T x+b_i}} \tag{4.6}$$

$$P(y=2|x) = \frac{e^{w_2^T x+b_2}}{1+\sum_{i=1}^{K-1} e^{w_i^T x+b_i}} \tag{4.7}$$

$$\cdots$$

$$P(y=K-1|x) = \frac{e^{w_{K-1}^T x+b_{K-1}}}{1+\sum_{i=1}^{K-1} e^{w_i^T x+b_i}} \tag{4.8}$$

假设有 N 个样本, 这里可以应用极大似然估计法估计模型参数, 从而得到对数几率回归模型. 似然函数为

$$L(w,b) = \prod_{i=1}^{N} P(y=y_i|x) = \frac{e^{\sum_{i=1}^{N} w_{y_i}^T x+\sum_{i=1}^{N} b_{y_i}}}{\left(1+\sum_{i=1}^{K-1} e^{w_i^T x+b_i}\right)^N}$$

对数似然函数为

$$l(w,b) = \ln L(w,b) = \sum_{i=1}^{N} w_{y_i}^T x + \sum_{i=1}^{N} b_{y_i} - N\ln\left(1+\sum_{i=1}^{K-1} e^{w_i^T x+b_i}\right)$$

对对数似然函数的变量求一阶导数, 并令一阶导数等于 *0*, 求解方程组即可得到极大似然估计量 $\hat{w}$, $\hat{b}$, 将其代入式 *(4,6),(4,7),(4.8)* 即可得到对数几率回归的多分类模型.

参考文献

[1] Thomas G Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, 2:263–286, 1994.