COMP9417 - 机器学习教程。分类

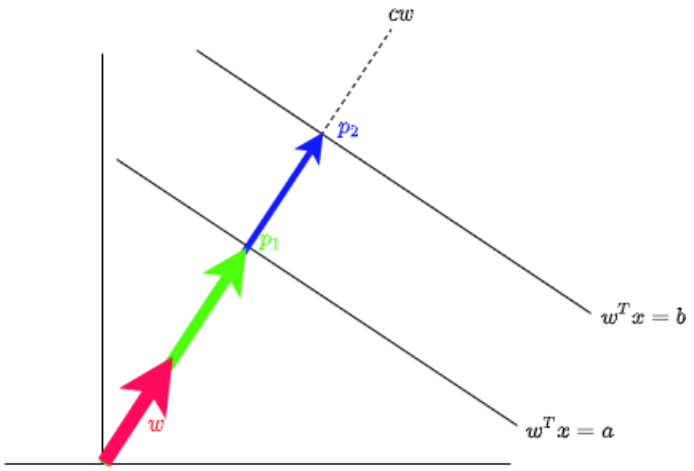
问题1.平行超平面之间的距离

在构建线性分类器时，一个常见的计算是计算两个平行超平面之间的距离。考虑两个平行超平面：$H_1 = \{x \in \mathbb{R}^n : w^T x = a\}$和$H_2 = \{x \in \mathbb{R}^n : w^T x = b\}$。证明$H_1$ 和H_2 之间的距离是由$\frac{|b-a|}{\|w\|_2}$。

提示：画一幅画。

解决方案。

考虑到以下对H_1 和 H_2 的图形描述。



两个超平面都垂直于w，w用红色描述。黑色虚线是对w的所有可能的倍数乘以一个非负常数c的追踪，因为一个向量乘以一个常数并不改变方向，只改变长度。这条线上的两个特定的向量在这里很重要，即p_1 和p_2，它们是与w方向相同的向量，分别位于H_1 和H_2 上。为了计算H_1 和H_2 之间的距离，我们只需要计算p_1 和p_2 之间的距离。

为此，我们注意到关于p_1 的两个信息：首先，对于某个常数d，$p_1 = dw$；其次，由于p_1 在H_1，我们还知道$w^T p_1 = a$。利用这两个事实，我们可以

断定

$$w^T p_1 = a \text{ and } p_1 = dw \Longrightarrow w^T (dw) = a$$
$$\Longrightarrow d = \frac{a}{\|w\|^2}$$
$$\Longrightarrow p_1 = \frac{a}{\|w\|^2} w.$$

对p_2进行类似的计算，可以得到

$$p_2 = \frac{b}{\|w\|^2} w.$$

现在，p_1和p_2之间的距离是。

$$\|p_2 - p_1\| = \left\| \frac{b}{\|w\|^2} w - \frac{a}{\|w\|^2} w \right\|$$
$$= (b - a) \frac{1}{\|w\|^2} \|w\|$$
$$|b-a| \frac{1}{\|w\|^2} \|w\|$$
$$\frac{|b - a|}{\|w\|^2}.$$

这个方程在整个课程中会出现无数次，确保你把它记在脑子里（当然也要理解其推导过程）。

问题2（感知器的训练和能力）

(a) 考虑以下训练数据。

x_1	x_2	y
-2	-1	-1
2	-1	1
1	1	1
-1	-1	-1
3	2	1

应用感知器学习算法，起始值$w_0 = 5$，$w_1 = 1$和$w_2 = 1$，学习率$\eta = 0.4$。请确保按照表格中的顺序，循环查看训练数据。以表格形式展示你的结果。

迭代	(w, x)	$y(w, x)$	w

解决方案。

运行Perceptron训练算法的结果是以下迭代，其中高亮的列标志着权重向量在该特定迭代中的变化。迭代结果

因此，正确的权重是$w_0 = 3.8$，$w_1 = 2.6$，$w_2 = 2.2$。这些权重首次出现在迭代9中。

迭代	(w, x)	$y(w, x)$	w
1	2.00	-2.00	$[4.6, 1.8, 1.4]^T$
2	6.80	6.80	$[4.6, 1.8, 1.4]^T$
3	7.80	7.80	$[4.6, 1.8, 1.4]^T$
4	1.40	-1.40	$[4.2, 2.2, 1.8]^T$
5	14.40	14.40	$[4.2, 2.2, 1.8]^T$
6	-2.00	2.00	$[4.2, 2.2, 1.8]^T$
7	6.80	6.80	$[4.2, 2.2, 1.8]^T$
8	8.20	8.20	$[4.2, 2.2, 1.8]^T$
9	0.20	-0.20	$[3.8, 2.6, 2.2]^T$
10	16.00	16.00	$[3.8, 2.6, 2.2]^T$
11	-3.60	3.60	$[3.8, 2.6, 2.2]^T$
12	6.80	6.80	$[3.8, 2.6, 2.2]^T$
13	8.60	8.60	$[3.8, 2.6, 2.2]^T$
14	-1.00	1.00	$[3.8, 2.6, 2.2]^T$

(b) 请考虑以下三个逻辑函数。

1. $A \wedge \neg B$
2. $\neg A \vee B$
3. $(a \vee b) \wedge (\neg a \vee \neg b)$

觉察器可以学习其中的哪些功能？解释一下。

问题3.二元逻辑回归，两个角度

回顾前面，我们可以把最小二乘作为一种学习线性回归问题的函数，给出上述函数，我们看作是一个基础问题，使用MLE，我们现在讨论二元逻辑回归问题的两个角度。在这个问题中，我们得到了一个数据集$D = \{(x_i, y_i)\}$。^{n} $_{i=1}$

其中x_i's代表特征向量，就像在线性回归中一样，但y_i's现在是二进制。在这里

我们的目标是将我们的输出建模为一个特定数据点属于两个类别之一的概率。我们将这个预测的概率表示为

$$P(y = 1|x) = P(x)$$

我们将其建模为

$$\hat{p}(x) = \sigma(\vec{w}x)。 \qquad \sigma(z) = \frac{1}{1 + e^{-z}},$$

其中w是我们估计的权重向量。然后，我们可以构建一个分类器，将以下类别分配给

具有最大的概率，即：。

$$\hat{y} = \arg\max_{k=0,1} P(\hat{y} = k|x) = \begin{cases} \text{如果}\sigma(\vec{w}^T x) \geq 0.5, \text{则为1。} \\ \text{否则为0} \end{cases}$$

注意：不要把函数σ(z)与参数σ混淆，后者通常表示标准差。

(a) Logistic sigmoid函数σ()在logistic回归公式中的作用是什么？为什么我们不能在这里简单地使用线性回归？(σ(z)的图表在这里可能会有帮助）。

(b) 我们考虑逻辑回归的统计学观点。记得在线性回归的统计学观点中，我们假设$y|x \sim N(x\beta^{T*}, \sigma^2)$仅仅对$x$进行建模，因为$x$是一个已知的随机变量，所以我们假设$y|x \sim \text{Bernoulli}(p^*)$仅仅对$p^*$进行建模，因为$p^*$是一个已知的随机变量，所以我们假设$p^* = \sigma(x^T w^*)$。

~~在这里，我们只有处理的问题是，$\vec{w}^T x$是一个实数，这意味着$\vec{w}^T x$可以是实线上的任何数字。这没有意义，因为我们希望将$\hat{p}$建模为一个概率，而概率属于区间[0, 1]。Logistic sigmoid（也被称为*压扁函数*）的作用是将*原始分数*$w^T x$压扁到所需的区间。~~

~~其中$p^* = \sigma(x^T w^*)$是反应属于第1类的真实未知概率，我们假设这是由一些真实的权重向量w^*控制的。写下数据D的对数可能性（作为w的函数），并进一步写下MLE目标（但不要试图解决它）。~~

解决方案。

该假设意味着

$$P(y = 1|x) = p^y (1 - y)^{1-y}$$

。

所以我们可以把对数可能性写成

$$\begin{aligned} \ln L(w) &\equiv \ln P(y_1, \ldots, y_n | x_1, \ldots, x_n) \\ &= \ln \prod_{i=1}^{n} P(y_i | x_i) \\ &= \sum_{i=1}^{n} \ln P(y_i | x_i) \\ &= \sum_{i=1}^{n} \ln(p_i^{y_i}(1 - p_i)^{1-y_i}) \\ &= \sum_{i=1}^{n} [y_i \ln p_i + (1 - y_i) \ln(1 - p_i)] \end{aligned}$$

。

在这一点上，我们记得

$$p_i = \sigma(w^T x_i) = \frac{1}{1 + e^{-w^T x_i}},$$

于是

$$\ln L(w) = \sum_{i=1}^{n} [y_i \ln \sigma(w^T x_i) + (1 - y_i) \ln(1 - \sigma(w^T x_i))]$$
$$= \sum_{i=1}^{n} y_i \ln \frac{\sigma(w^T x_i)}{1 - \sigma(w^T x_i)} + \ln(1 - \sigma(w^T x_i))$$

然后，MLE优化是

$$\hat{w}_{MLE} = \arg\max_{w \in R^p} \ln L(w)$$

请注意，我们不能像在线性回归案例中那样用手解决这个问题（你应该自己尝试一下）。这是一个我们完全依靠数值方法来寻找解决方案的例子。

(c) 逻辑回归问题的另一种方法是纯粹从优化的角度来看待它。这就要求我们选择一个损失函数，并求出相应的最小化器。写下逻辑回归的MSE目标，并讨论你是否认为这个损失是合适的。

解决方案。
MSE目标简单来说就是

$$\arg\min_{w} \frac{1}{n} ||y - \sigma(Xw)||_2^2$$

其中$y = [y_1, \ldots, y_n]^T$，$\sigma(Xw)$是σ对向量Xw的元素顺向应用，或者我们可以等价地写成。

$$\text{最小值}_{w} \frac{1}{n} \sum_{i=1}^{n} (y_i - \sigma(w^T x_i))^2$$

这个目标在数学上是有意义的，但是平方误差似乎不是一个特别好的方法来衡量响应者y_i和我们的预测$\sigma(w^T x_i)$之间的距离。原因是y_i是二进制的，而我们的预测是实值的。不希望使用MSE还有更多的技术原因，特别是MSE在这个问题的参数向量w中是不凸的，这使得它更难优化，但这已经超出了本课程的范围。

(d) 考虑以下问题：给你两个离散的概率分布P和Q，并要求你量化Q离P有多远。 这是统计学和信息论中非常常见的任务。衡量两者之间差异的最常用方法是

来计算Kullback-Liebler（KL）分歧，也称为相对熵，其定义为：。

$$D_{\mathrm{KL}}(P\,/\!/Q) = \sum_{x\in X} P(x)\ln\frac{P(x)}{Q(x)}。$$

其中，我们对基础随机变量的所有可能值进行求和。一个好的方法是，我们有一个真实的分布P，一个估计值Q，我们试图弄清楚我们的估计值有多糟糕。写下两个伯努利分布P = Bernoulli(p)和Q = Bernoulli(q)之间的KL背离。

解决方案。
一个伯努利随机变量只能有两个值（即x=0，1），因此我们有。

$$\begin{aligned} D_{\text{KL(P//Q)}} &= \sum_{x=0}^{1} P(x)\ln\frac{P(x)}{Q(x)} \\ &= P(X=0)\ln\frac{P(X=0)}{Q(X=0)} + P(X=1)\ln\frac{P(X=1)}{Q(X=1)} \\ &= (1-p)\ln\frac{1-p}{1-q} + p\ln\frac{p}{q}. \end{aligned}$$

(e) 继续基于优化的观点。在我们的设置中，量化我们的预测$\hat{p}_i$ 和真实标签y_i 之间的差异的一种方法是看两个伯努利分布P_i = Bernoulli(y_i)和Q_i = Bernoulli($\hat{p}_i$)之间的KL分歧。用这个来写下逻辑回归问题的适当最小化。

解决方案。
我们可以简单地将KL损失定义为KL分歧之和。

$$\begin{aligned} L_{\text{KL(w)}} &= \sum_{i=1}^{n} D_{\text{KL(P}_i\text{//Q}_i\text{)}} \\ &= \sum_{i=1}^{n} D_{\mathrm{KL}}(\text{Bernoulli}(y_i)\,/\!/\,\text{Bernoulli}(\hat{p}_i)) \\ &= \sum_{i=1}^{n} (1-y_i)\ln\frac{1-y_i}{1-\hat{p}_i} + y_i\ln\frac{y_i}{\hat{p}_i}. \end{aligned}$$

请注意，我们可以将各个条款扩展为：。

$$\ln\frac{1-y_i}{1-\hat{p}_i} + y_i\ln\frac{y_i}{\hat{p}_i} = (1-y_i)\ln(1-y_i) - (1-y_i)\ln(1-\hat{p}_i) + y_i\ln y_i - y_i\ln$$
$$\overset{w}{\propto} -(1-y_i)\ln(1-\hat{p}_i) - y_i\ln\hat{p}_i。$$

在上面的最后一步中，我们已经注意到只有涉及$\hat{p}_i$ *的*条款是w的函数，因此我们可以安全地从考虑中删除所有其他条款。这是因为我们正在优化

因此，只需考虑涉及w的条款即可。

$$L_{\text{隆坡}}(w) \propto -\sum_{i=1}^{n} [(1-y_i)\ln(1-\hat{p}_i) + y_i \ln \hat{p}_i] \quad i$$

$$= -\sum_{i=1}^{n} y_i \ln \frac{p}{1-\hat{p}_i} + \ln(1-\hat{p}_i)$$

$$= -\sum_{i=1}^{n} y_i \ln \frac{\sigma(w^T x_i)}{1-\sigma(w^T x_i)} + \ln(1-\sigma(w^T x_i)) \circ$$

(f) 在逻辑回归（和其他二元分类问题）中，我们通常使用交叉熵损失，定义为

$$L_{XE}(a, b) = -a \ln b - (1-a)\ln(1-b)。$$

利用你在上一部分的结果，讨论为什么XE损失是一个好的选择，并在逻辑回归的统计和优化观点之间建立联系。

解决方案。

我们可以很容易地看到，最小化交叉熵损失等同于最小化前一个问题中的KL损失，所以XE损失是一个有效的目标函数。为了具体化，我们有

$$L_{KL}(w) \propto L_{XE}(w)。$$

因此，立即可以得出

$$\arg\min_w L_{KL}(w) = \arg\min_w L_{XE}(w),$$

所以XE给出的解决方案与KL发散最小化的解决方案相同。现在，利用我们在上一个问题中的结果，并将其与我们上面的MLE推导进行比较，我们可以看到

$$\hat{w}_{MLE} = \arg\max_{w \in Rp} \ln L(w)$$

$$= \arg\max_{w \in Rp} \sum_{i=1}^{n} y_i \ln \frac{\sigma(w^T x_i)}{1-\sigma(w^T x_i)} + \ln(1-\sigma(w^T x_i))$$

$$= \arg\min_{w \in Rp} -\sum_{i=1}^{n} y_i \ln \frac{\sigma(w^T x_i)}{1-\sigma(w^T x_i)} + \ln(1-\sigma(w^T x_i))$$

$$= \arg\min_{w \in Rp} L_{KL}(w)$$

$$= \arg\min_{w \in Rp} L_{XE}(w).$$

因此，我们看到的是，最小化XE损失与最大化对数似然得到的解决方案是一样的。这与我们在线性回归案例中看到的情况类似：使用MSE等同于在高斯假设下最大化似然。我们可以把这看作是。MSE之于高斯，如同XE之于伯努利。

问题4.数值求解逻辑回归问题

在前面的问题中，我们表明，为了解决逻辑回归问题，我们必须解决以下优化问题。

$$\hat{w} = \arg\min_w L(w)$$
$$= \arg\min_w -\sum_{i=1}^{n} y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \quad 。$$

其中

$$p_i = \sigma(w^T x_i) = \frac{1}{1 + e^{-w^T x_i}}.$$

不幸的是，在这种情况下，我们不能以封闭形式求解$\hat{w}$。换句话说，我们不能像线性回归那样简单地求导、等价于零和求解来得到一个好的解决方案。相反，我们必须依靠数字技术，如梯度下降。在这个问题中，我们将通过并推导出逻辑回归问题的梯度下降更新。

(a) 为了进行任何形式的梯度下降，我们将需要求导数。说明

$$\sigma'(z) = \sigma(z)(1 - \sigma(z))。$$

然后用这个结果来证明

$$\frac{dp_i}{dw} = p_i(1 - p_i)x_i 。$$

其中$p_i = \sigma(w^T x_i)$。

解决方案。
第一个结果是通过链式规则的标准应用得出的，更多细节见HW0。第二个结果也是通过连锁规则立即得出的。

$$\frac{dp_i}{dw} = \frac{d\sigma(w^T x_i)}{d(w^T x_i)} \frac{d(w^T x_i)}{dw} = \sigma(w^T x_i)(1 - \sigma(w^T x_i))x_i = p_i(1 - p_i)x_i 。$$

(b) 用前面的结果来说明

$$\frac{d \ln p_i}{dw} = (1 - p_i)x_i 。$$

解决方案。
再次使用链式规则，我们可以得到

$$\frac{d \ln p_i}{dw} = \frac{d \ln p_i}{dp_i} \frac{dp_i}{dw} = \frac{1}{p_i} p_i(1-p_i)x_i = (1-p_i)x_i$$

。

(c) 利用前几部分的结果，计算出

$$\frac{dL(w)}{dw}$$

并写出w的梯度下降更新，步长为η。

解决方案

$$\begin{aligned}
\frac{dL(w)}{dw} &= \frac{d}{dw}\left(-\sum_{i=1}^{n} y_i \ln \hat{p}_i + (1-y_i)\ln(1-\hat{p}_i)\right) \\
&= -\sum_{i=1}^{n} y_i \frac{d}{dw}\ln \hat{p}_i + (1-y_i)\frac{d}{dw}\ln(1-\hat{p}_i) \\
&= -\sum_{i=1}^{n} y_i (1-\hat{p}_i)x_i + (1-y_i)\left(-\frac{1}{1-\hat{p}_i}\hat{p}_i(1-\hat{p}_i)x_i\right) \\
&= -\sum_{i=1}^{n} [y_i(1-\hat{p}_i)x_i - (1-y_i)\hat{p}_i x_i] \\
&= -\sum_{i=1}^{n} (y_i - \hat{p}_i)x_i
\end{aligned}$$

因此，在迭代$k+1$的梯度下降更新为

$$w^{(k+1)} = w^{(k)} + \eta \sum_{i=1}^{n} (y_i - \hat{p}_i)x_i$$

(d) 凸函数没有任何局部最小值，因此当我们对凸函数进行梯度下降时，无论我们的初始化如何，都能保证收敛到一个全局最小值。

$w^{(0)}$. Prove that $L(w)$ is convex.

解决方案。

回顾一下，一个函数的Hessian，$H \in \mathbb{R}^{p \times p}$，只是该函数的二阶偏导数的矩阵，所以有（$k$，$l$）个元素。

$$\begin{aligned}
H_{k,l} &= \frac{\partial^2 L(w)}{\partial w_k \partial w_l} \\
&= \frac{\partial}{\partial w_k}\left(-\sum_{i=1}^{n}(y_i - \hat{p}_i)x_{i,l}\right) \\
&= \sum_{i=1}^{n} \hat{p}_i(1-\hat{p}_i)x_{i,k}x_{i,l}
\end{aligned}$$

其中x_{il}是*第i个*输入向量x_i的*第l个*特征。从这里可以直接得出结论：

$$H = \sum_{i=1}^{n} \hat{p}_i(1-\hat{p}_i)x_i x_i^T$$

现在，回顾一下前面的教程，如果一个函数的Hessian是正半定的，那么它就是凸的，也就是说，对于任何非零矢量u。

$$u^T Hu \geq 0。$$

在我们的案例中，我们有

$$\begin{aligned} u^T Hu &= \sum_{i=1}^{n} \hat{p}_i(1-\hat{p}_i)(u^T x_i)(x_i^T u) \\ &= \sum_{i=1}^{n} \hat{p}_i(1-\hat{p}_i)(u^T x_i)^2 \\ &\geq 0, \end{aligned}$$

其中最后一行成立，因为$\hat{p}_i \in (0, 1)$和$(u^T x_i)^2 \geq 0$。