机器学习：监督学习(Supervised Learning)之分类(Perceptron, Logistic, Softmax, Bayes)

---

1. 分类问题

结果输出为离散化的类别，而不是连续的值。

例：预测一个用户是否点击特定商品，是否会购买给定的品类，评论是正面还是负面等。

模型：输入 x ，标记结果： $y=\{0,1\}$ 。

预测输出：$0 \leq h(x) \leq 1$

$$if\ h(x_i) > 0.5, \quad y_i = 1$$
$$if\ h(x_i) \leqslant 0.5, \quad y_i = 0$$

2. 感知机($Perceptron$)

模型： $h(x)=g(w^Tx+b)$ 。$g(x)$ 为 $sign$ 符号函数。

$$g(x)=\begin{cases}+1, & x>0\\ -1, & x<0\end{cases}$$

感知机的目标是优化如下损失函数：

$$\min_{w,b}\sum_{i=1}^{m}-y_i(w^Tx_i+b)$$

对它求梯度，之后可以用梯度下降方法更新：

$$
\begin{aligned}
&while(until\ 所有样本正确分类 \vee 迭代次数到\ n)：\\
&\quad if\ sign(w^Tx_i+b)\neq y_i\\
&\quad\quad then\ w_{t+1}\leftarrow w_t+y_ix_i\\
&\quad else：continue
\end{aligned}
$$

3. $Fisher$ 线性判别分析(LDA)与有监督降维

LDA 的优化目标是求广义瑞利商的极大值。

3.1. 矩阵的特征值与特征向量($Eigvalue\&Eigvector$)

在Lesson 1 解方程一章中，我们已经了解了基本的矩阵特征值和特征向量求解，这里给出更详细的过程。对于 n 阶矩阵 A，其特征向量 x 表示，用 x 对 A 做线性变换后的向量，还是和 A 在同一条直线上。变化的比例是特征值 λ，表示新向量的方向和长度都可能会改变。数学上表示为，存在一个数 λ 和非 0 向量 x，使得：

$$
\begin{aligned}
&Ax=\lambda x\\
\Leftrightarrow&(A-\lambda I)x=0
\end{aligned}
$$

则称 λ 为矩阵 A 的特征值，x 为对应的特征向量。

根据Lesson 3 优化方法基础的基本概念部分，上面齐次方程有解的条件是方程的系数矩阵的行列式必须为 0 (或者系数矩阵不可逆，即其秩 $r(A-\lambda I)<n$)。

$$
|A-\lambda I|=0
$$

先求特征值：$|A-\lambda I|=0$，代回齐次方程求对应的特征向量，属于不同特征值的特征向量线性无关。Lesson 7 信息论与决策树会讲矩阵的特征值分解和奇异值分解算法。

对角矩阵，上三角矩阵，下三角矩阵的特征值为其主对角线元素。

n_i：代数重数，对应于特征多项式 $(\lambda-\lambda_1)^{n_1}\cdots(\lambda-\lambda_i)^{n_i}=0$ 的 λ_i 的阶数。

m_i：几何重数，对应于 $(A-\lambda_iI)x=0$ 的其次方程的线性无关的解的个数。

3.1.1 矩阵变换

**相似变换：对于 A,B 和可逆矩阵 P，满足：

$$
P^{-1}AP=B
$$

$A\sim B$，这一变换不改变矩阵的特征值，且 $r(A)=r(B),|A|=|B|$。

这一性质可用来求特征值，将其相似化为对角阵或三角阵，特征值即为主对角线上的元素。

矩阵相似对角化：

$$
P^{-1}AP=\Lambda
$$

前提是 P 可逆，即 A 有 n 个线性无关的特征向量。

**正交变换：对于正交矩阵 P ($P^{-1}=P^T$, 即$P^TP=I$)，满足：

$$
P^TAP=\Lambda
$$

通过正交变换可以构建对角阵。

实现时需要对同一个特征值的不同特征向量正交化，然后将所有正交化后的特征向量标准化即可。

3.2. 广义特征值与广义特征向量

广义特征值定义于矩阵 A,B 之上，对于方阵 A,B 存在一个数 λ 和非 0 向量 x，使得：

$$
Ax=\lambda Bx
$$

称 λ 为广义特征值，x 为对应的广义特征向量。类似有：

$$
|A-\lambda B|=0
$$

如果矩阵 B 可逆，上述问题也可以转为基本的特征值问题：

$$B^{-1}Ax = \lambda x$$

广义特征值在线性判别分析，以及Lesson 7 信息论与决策树的流形学习都会用到。

3.3. 拉格朗日乘数法

拉格朗日乘数法被广泛用于求解带约束条件的最优化问题，在Lesson 6 支撑向量机求解中有广泛应用。以一阶线性的优化函数为例，对于优化问题：

$$\begin{cases} \min\limits_{x} f(x) \\ s.t \quad h_i(x) \quad = 0 \quad i = 1, \cdots, p \end{cases}$$

可以构造拉格朗日乘子函数：

$$\mathcal{L}(x, \lambda) = f(x) + \sum_{i=1} p\lambda_i h_i(x)$$

对优化变量 $x, y, \cdots$，乘子变量 λ 分别求偏导并令其为 0，可以得到候选极值点。再求黑塞矩阵，判别它是否正定，判断该候选极值点是否是极值点。

3.4. 瑞利商($Rayleigh\ Quotient$)

瑞利商部分，在 LDA, PCA, LLE, SNE, LE 等降维算法中都需要应用它的一些结论和推导过程，所以理解瑞利商及其性质，对于降维算法的学习非常重要(更多降维算法在Lesson 7 信息论与决策树)。

方阵 A 和非 0 向量 x 的瑞利商定义为如下比值：

$$R(A, x) = \frac{x^T Ax}{x^T x}$$

对于任意的非 0 实数 k，有 $R(A, kx) = R(A, x)$，表明向量缩放后瑞利商不变，存在冗余。假设 $\lambda_{\min}, \lambda_{max}$ 是矩阵 A 的最小和最大特征值，则有：

$$\lambda_{\min} \leqslant R(A, x) \leqslant \lambda_{max}$$

当 x 分别为最小和最大特征值对应的特征向量时，$R(A, x)$ 取这两个值。

关注瑞利商的极大值求解的优化问题。这里首先和Lesson 6 支撑向量机 $primal\ SVM$ 的思路一样，我们添加约束来保证解的唯一性，即限定分母为 1 (x 为单位向量，$x^T x = 1$)，瑞利商变为：$R(A, x) = x^T Ax$ 。

使用拉格朗日乘子法，构造拉格朗日乘子函数：

$$L(x, \lambda) = x^T Ax + \lambda(x^T x - 1)$$

对 x 求梯度并令梯度为 0：$2Ax + 2\lambda x = 0$，即 $Ax = \lambda x$，这表示瑞利商的所有极值在矩阵 A 的特征值和特征向量处取得。假设 λ_i, x_i 带入瑞利商：

$$R(A, x_i) = \frac{x_i^T(Ax_i)}{x_i^T x_i} = \frac{x_i^T(\lambda_i x_i)}{x_i^T x_i} = \lambda_i$$

因此，在最大的特征值处，瑞利商有最大值，在最小的特征值处，瑞利商有最小值。Lesson 8 无监督学习(聚类, 信号分解, 流形降维)的主成分分析用到了这一结论。

3.5. 广义瑞利商

对瑞利商推广得到广义瑞利商：

$$R(A, B, x) = \frac{x^T Ax}{x^T Bx}$$

广义瑞利商也存在冗余，$R(A, B, kx) = R(A, B, x)$ 。如果对矩阵 B 做楚列斯基分解，即 $B = CC^T$，并令 $x = (C^T)^{-1}y$，可将广义瑞利商转为瑞利商形式：

$$\frac{x^T Ax}{x^T Bx} = \frac{((C^T)^{-1}y)^T A((C^T)^{-1}y)}{((C^T)^{-1}y)^T B((C^T)^{-1}y)} = \frac{y^T C^{-1} A(C^T)^{-1} y}{y^T y}$$

根据瑞利商的结论，广义瑞利商的最大值为矩阵 $C^{-1}A(C^T)^{-1}$ 的最大特征值，最小值为矩阵 $C^{-1}A(C^T)^{-1}$ 的最小特征值。也可以通过广义特征值直接求广义瑞利商的极大值，同样添加约束：$x^TBx=1$，构造拉格朗日乘子函数：

$$R(A,B,x)=x^TAx+\lambda(x^TBx-1)$$

对 x 求梯度并令梯度为 0：$2Ax+2\lambda Bx=0$，即 $Ax=\lambda Bx$，这是广义特征值问题。如果 B 可逆，则 $B^{-1}Ax=\lambda x$。这表示瑞利商的所有极值在矩阵 $B^{-1}A$ 的特征值和特征向量处取得。假设 λ_i,x_i 带入瑞利商：

$$R(A,B,x_i)=\frac{x_i^T(Ax_i)}{x_i^TBx_i}=\frac{x_i^T(\lambda_iBx_i)}{x_i^TBx_i}=\lambda_i$$

因此，在最大的广义特征值处，瑞利商有最大值，在最小的广义特征值处，瑞利商有最小值。最大化广义瑞利商也是线性判别分析的优化目标。

3.6. LDA 的二分类问题求解

菲谢尔线性判别算法的核心思想是：设法将样本点投影到一条直线 w 上，使得同类的样本投影点尽可能接近，异类的样本投影点尽可能远离。对于新加入的样本，先投影到这条直线上，再根据投影点的位置来确定新样本的类别。

给定数据集 $D=\{(x_i,y_i)\}_{i=1}^m,y_i\in\{0,1\}$。令 X_i,μ_i,Σ_i 分别表示第 $i\in\{0,1\}$ 类示例的集合，均值向量，协方差矩阵。若将数据投影到直线 w 上，则两类样本的中心在直线的投影 $E[w^Tx|y=y_i]$ 分别为 $w^T\mu_0,w^T\mu_1$，由此推导出，两类样本的协方差 $E[(w^Tx-E[w^Tx|y=y_i])^2]$ 分别为 $w^T\Sigma_0w,w^T\Sigma_1w$。

欲使同类样本投影点尽可能接近，可让同类样本投影点的协方差，即 $w^T\Sigma_0w+w^T\Sigma_1w$ 尽可能小。欲使异类样本投影店尽可能远离，可让类中心距离，即 $||w^T\mu_0-w^T\mu_1||_2^2$ 尽可能大，可得优化函数为：

$$\begin{aligned}J(w)&=\frac{||w^T\mu_0-w^T\mu_1||_2^2}{w^T\Sigma_0w+w^T\Sigma_1w}\\&=\frac{w^T(\mu_0-\mu_1)(\mu_0-\mu_1)^Tw}{w^T(\Sigma_0+\Sigma_1)w}\\w^*&=\arg\max_wJ(w)\end{aligned}$$

可以定义类内散度矩阵($within-class\ scatter\ matrix$)：

$$S_w=\Sigma_0+\Sigma_1=\sum_{x\in X_0}(x-\mu_0)(x-\mu_0)^T+\sum_{x\in X_1}(x-\mu_1)(x-\mu_1)^T$$

以及定义类间散度矩阵($between-class\ scatter\ matrix$)：

$$S_b=(\mu_0-\mu_1)(\mu_0-\mu_1)^T$$

则 $J(w)$ 可以写为如下广义瑞利商形式：

$$\begin{aligned}J(w)&=\frac{w^TS_bw}{w^TS_ww}\\w^*&=\arg\max_wJ(w)\end{aligned}$$

因此，LDA 线性判别分析的目标就是最大化这个广义瑞利商。上面我们说过，可以根据冗余添加约束条件，变号转为求最小值的最优化问题：

$$\begin{cases}\min_w-w^TS_bw\\s.t.\quad w^TS_ww=1\end{cases}$$

并使用拉格朗日乘子法，对 x 求梯度并令梯度为 0：

$$\begin{aligned}L(w,\lambda)&=-w^TS_bw+\lambda(w^TS_ww-1)\\2S_bw+2\lambda S_ww&=0\end{aligned}$$

等价于 $S_bw=\lambda S_ww$。注意到 S_bw 的方向恒为 $\mu_0-\mu_1$，只关注投影的方向，令 $S_bw=\lambda(\mu_0-\mu_1)$，可得：

$$w^*=S_w^{-1}(\mu_0-\mu_1)$$

对于 S_w^{-1} 的求解，考虑数值解的稳定性，实际中通常是先对 S_w 做奇异值分解($S_w=U\Sigma V^T$)，Σ 是一个实对角矩阵，其对角线上的元素是 S_w 的奇异值，然后再由 $S_w^{-1}=V\Sigma^{-1}U^T$ 得到 S_w^{-1}。

3.7. 算法过程

首先获取带标签的训练集，并计算两类样本的 μ_0, μ_1 和 Σ_0, Σ_1 。

计算类内散度矩阵：

$$S_w = \Sigma_0 + \Sigma_1$$

通过奇异值分解计算 S_w 的逆，然后计算最佳投影向量：

$$w^* = S_w^{-1}(\mu_0 - \mu_1)$$

找到投影向量后，对测试集的任意样本 x ，定义 s ：

$$s = w^{*T}x = (S_w^{-1}(\mu_0 - \mu_1))^T x$$

设判别门限 s' 为：

$$s' = \frac{w^{*T}(\mu_0 + \mu_1)}{2}$$

计算判别门限后，对测试集的任意样本 x 分类：

$$\begin{cases} y = 1,\ if\ s = w^{*T}x > s' \\ y = 0,\ if\ s = w^{*T}x < s' \end{cases}$$

3.8. 多分类 LDA 与有监督降维算法

假设存在 N 个类别，且第 i 类的样本数为 m_i ，定义全局散度矩阵：

$$S_t = S_b + S_w = \sum_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T$$

其中 μ 是所有样本的均值向量，将 S_w 重定义为每个类别的散度矩阵之和：

$$S_w = \sum_{i=1}^{N} S_{w_i}$$

其中，$S_{w_i} = \sum_{x \in X_i}(x - \mu_0)(x - \mu_0)^T$ 。因此可以得到：

$$S_b = S_t - S_w = \sum_{i=1}^{N} m_i(\mu_i - \mu)(\mu_i - \mu)^T$$

显然多分类有多种实现方法，使用 S_b, S_w, S_t 中的任意两个即可，常见的一种为：

$$\max_W \frac{tr(W^T S_b W)}{tr(W^T S_w W)}$$

其中 $W \in \mathbb{R}^{d \times (N-1)}$ ，$tr(\cdot)$ 表示矩阵的迹。同样根据瑞利商最大的推导，上式可以通过如下的广义特征值求解：

$$S_b W = \lambda S_w W$$

W 的闭式解为 $S_w^{-1}S_b$ 的 d' 个最大非零广义特征值对应的特征向量所组成的矩阵，$d' \leqslant N - 1$ 。

降维：在上面的二分类问题中，w^*x 实际上是将一个二维的样本点投影到了一条一维的直线上，因此 LDA 可以被用来线性降维。

对于多分类 LDA 也是同样的思路。若将 W 看为投影矩阵，则多分类 LDA 通过 $W^T X$ 将样本从 d 维空间投影到了 d' 维空间，d' 通常远小于样本本有的特征列数 d 。

因此，可以通过这个投影将样本从 $d \to d'$ ，从而减小样本点的维数。

投影过程中用到了类别信息，因此这也是一种典型的监督降维技术。

4. 逻辑斯蒂回归($Logistic\ Regression$)

其和感知机，线性回归的区别在于，激活函数不同。感知机是符号函数 $sign(w^Tx+b)$，线性回归是线性映射 w^Tx+b，而逻辑回归模型如下：

前向模型： $h(x)=\sigma(w^Tx+b)$：

$$\sigma(x)=\frac{1}{1+e^{-x}},\quad \sigma(x)\in(0,1)$$
$$\sigma'(x)=\sigma(x)(1-\sigma(x))$$

最终分类器的分类结果为： $y_{pred}=sign(h(x)-0.5)$。

其中 $\sigma(x)$ 为 $Sigmoid$ 函数，$sign$ 为符号函数。

$h(x)$ 表示概率的信息： $h(x)=p(y=1|x;w;b)$，且满足 $p(y=1|x;w;b)+p(y=0|x;w;b)=1$，即随机变量 y 服从 $p=0.5$ 伯努利分布($0-1$ 分布)。

4.1. $Sigmoid$ 函数的推导

为什么加上 $Sigmoid$ 函数后，模型就能从回归问题变为分类问题？这和 $Sigmoid$ 本身的推导过程有关。

首先构造分类规则，使用 $\ln$ 可以简化计算：下式大于 0 则分为第一类，小于 0 则分为第二类：

$$\ln\frac{p(y=+1|x)}{p(y=-1|x)}>0$$

还是通过贝叶斯规则来定义 $p(y|x)$：

$$p(y|x)=\frac{p(x|y)p(y)}{p(x)}$$

其中，$p(y)$ 服从伯努利分布，$p(x|y)$ 定义为任何我们想要的分布，比如说正态分布。

对于 $y\sim Bern(\pi)$ 和 $x|y\sim N(\mu_y,\sum)$，y 被分类为 1 如果：

$$\ln\frac{p(y=+1|x)}{p(y=-1|x)}>0$$

以二维正态分布为例，则可以将式子转乘法为加法：

$$\ln\frac{p(x|y=+1)p(y=+1)}{p(x|y=-1)p(y=-1)}$$
$$=\ln\frac{\pi_1}{\pi_0}-\frac{1}{2}(\mu_1+\mu_0)^T\Sigma^{-1}(\mu_1-\mu_0)+x^T\Sigma^{-1}(\mu_1-\mu_0)$$

令 $\ln\frac{\pi_1}{\pi_0}-\frac{1}{2}(\mu_1+\mu_0)^T\Sigma^{-1}(\mu_1-\mu_0)$ 表示偏置 b，$\Sigma^{-1}(\mu_1-\mu_0)$ 表示权重 w。

则分类面为如下函数：

$$L=\ln\frac{p(y=+1|x)}{p(y=-1|x)}=w^Tx+b$$

$L>0\rightarrow y=+1;\ L<0\rightarrow y=-1$，注意 $L=0$ 能分给任何一类。

根据伯努利分布规则，有 $p(y=-1|x)=1-p(y=+1|x)$。又因为有 $\ln\frac{p(y=+1|x)}{p(y=-1|x)}=w^Tx+b$，则可通过如下求解 $p(y=+1|x)$，并推导出 $Sigmoid$ 函数：

$$p(y=+1|x)=\frac{\exp\{wx^T+b\}}{1+\exp\{wx^T+b\}}=\sigma(wx^T+b)$$

4.2. 交叉熵损失的推导(最大似然估计)

在Lesson 7 信息论与决策树我们将会对交叉熵有更深入的了解。首先定义模型训练的标签为：

$$y_i\in R\Rightarrow y_i\in\{0,1\}$$

$Logistic$ 回归拟合的就是伯努利分布，即 $P(y_i=1)=p, P(y_i=0)=1-p$。

设 y_i 满足伯努利分布，μ_i 满足高斯分布。

根据伯努利分布，假设有 n 个样本，每个样本属于每个类的概率可以写成：

$$p(y_i|x_i,w)=\sigma(x_i)^{y_i}(1-\sigma(x_i))^{1-y_i}$$

$$\text{其中，}\sigma(x)=\frac{1}{1+\exp(-w^Tx+b)}$$

由于样本独立同分布，根据Lesson 3.5 参数估计(MLE, MAP, Bayes, KNN, Parzen, GMM, EM算法)似然函数参数估计，训练集的似然函数为：

$$\begin{aligned}p(y|X,w)&=Ber(y|\mu(X,w))\\&=\prod_{i=1}^{n}Ber(y_i|\sigma(w^Tx_i+b))\\&=\prod_{i=1}^{n}(\sigma(x_i)^{y_i}(1-\sigma(x_i))^{1-y_i})\end{aligned}$$

而逻辑回归算法的目标就是最大化这个似然函数，取对数得到：

$$\begin{aligned}L(w)&\triangleq\ln p(y|X,w)\\&=\sum_{i=1}^{n}\left[y_i\ln\sigma(x_i)+(1-y_i)\ln(1-\sigma(x_i))\right]\\&=\sum_{i=1}^{n}\left[\ln(\mu_i)^{C(y_i=1)}(1-\mu_i^{C(y_i=0)})\right]\end{aligned}$$

其中，$\mu_i\triangleq\sigma(x_i)$ 。定义随机变量 y 和 μ 之间的关系为：

$$C(y_i=1)=\begin{cases}1, & y_i=1\\0, & \mu_i=0\end{cases}$$

$$C(y_i=0)=\begin{cases}1, & y_i=0\\0, & \mu_i=1\end{cases}$$

因此，原式 $=$

$$\begin{aligned}&=\max\sum_{i=1}^{n}[y_i\ln\mu_i+(1-y_i)\ln(1-\mu_i)]\\&=\min\sum_{i=1}^{n}-[y_i\ln\mu_i+(1-y_i)\ln(1-\mu_i)]\end{aligned}$$

其中，$-\sum_{i=1}^{n}[y_i\ln\mu_i+(1-y_i)\ln(1-\mu_i)]$ 也就是交叉熵损失 $J(w)$ ，衡量的是变量 y_i 和 μ_i(即$\sigma(x_i)$) 之间的差异，如果值越大，说明两个概率分布的差异越大。越小，说明两个概率分布的差异越小，即样本集的预测结果和标签之间的误差越小。当两个的概率分布相等时($y_{pred}=y_{label}$)，交叉熵有极小值，而 $Logistic$ 回归的目标函数，就是求这个损失的极小值。

在Lesson 7 信息论与决策树中会详细讨论这个部分。

我们对标签 y_i 从 $0,1$ 推广到 $-1,1$ ，则交叉熵损失可以表示如下：

$$y_i\in\{0,1\}\rightarrow\tilde{y}_i\in\{-1,1\}$$

$$\therefore-\min\sum_{i=1}^{n}[y_i\ln\mu_i+(1-y_i)\ln(1-\mu_i)]=\min\sum_{i=1}^{n}\ln(1+\exp(-\tilde{y}_iw^Tx_i))$$

因此：

$$\begin{aligned}J(w)&\triangleq\sum_{i=1}^{n}\ln(1+\exp(-\tilde{y}_iw^Tx_i))\\\nabla J(w)&=X^T(\mu-\tilde{y})\\\nabla^2J(w)&=X^Tdiag(\mu_i(1-\mu_i))\end{aligned}$$

这样，我们就把最大似然估计转为了最小化交叉熵损失的优化问题。之后就可以用常规的随机梯度下降方法求解。

4.3. 交叉熵和均方误差的随机梯度下降优化

注意：根据Lesson 9 凸优化内容，交叉熵误差可以确保最优化问题为凸优化问题，找到全局最优解；而均方误差(欧氏距离)不一定会是凸函数，可能存在多个局部极小值，寻找全局最优解存在困难。

$Logistic$ 回归的目标是优化如下交叉熵损失函数(损失表达公式的推导在前文)。

$$L = \min_{w,b} \sum_{i=1}^{n} \ln(1 + \frac{1}{-y_i w^T x_i}) = \min_{w,b} \sum_{i=1}^{n} \ln(1 + \exp(-ys))$$

对 w 求梯度：

$$\nabla L(w) = \frac{\partial \sigma(w,x)}{\partial w_i} = \frac{\exp(y_i w^T x_i)}{1 + \exp(y_i w^T x_i)}(-y_i x_i) = \sigma(-yw^T x)(-yx)$$

则随机梯度下降算法过程如下：

$$\begin{aligned} &while(until\ \nabla L(w) = 0 \vee 迭代次数到\ n):\\ &\quad for\ batch\ in\ dataloader(shuffle):\\ &\qquad \nabla L(w) = \frac{1}{batch\_size} \sum_{i=1}^{batch\_size} \sigma(-y_i w_t^T x_i)(-y_i x_i)\\ &\qquad w_{t+1} = w_t - \eta \cdot \nabla L(w) \end{aligned}$$

如果是均方误差，其目标是优化如下损失函数：

$$L = \min_{w,b} \sum_{i=1}^{n} (\frac{1}{1 + e^{-y_i w^T x_i}} - y_i)^2$$

对 w 求梯度：

$$\nabla L(w) = \frac{\partial \sigma(w,x)}{\partial w_i} = 2(\sigma(yw^T x) - 1)\sigma(yw^T x)(1 - \sigma(yw^T x))yx$$

则随机梯度下降算法过程如下：

$$\begin{aligned} &while(until\ \nabla L(w) = 0 \vee 迭代次数到\ n):\\ &\quad for\ batch\ in\ dataloader(shuffle):\\ &\qquad \nabla L(w) = \frac{2}{batch\_size} \sum_{i=1}^{batch\_size} (\sigma(y_i w_t^T x_i) - 1)\sigma(y_i w_t^T x_i)(1 - \sigma(y_i w_t^T x_i))y_i x_i\\ &\qquad w_{t+1} = w_t - \eta \cdot \nabla L(w) \end{aligned}$$

4.4. $Logistic$ 回归中的分治法

这里主要是用了Lesson 3 优化方法基础介绍的分治法中的坐标下降法的思想。

给定了 l 个训练样本 $(x_i, y_i), i = 1, \cdots, l$，其中 $x_i \in \mathbb{R}^n$ 为特征向量，$y_i = \pm 1$ 为标签。根据Lesson 4 监督学习之回归(Linear, NonLinear, Lasso, Ridge, Generalization)和Lesson 6 支撑向量机的内容，类比定义带 L_2 正则化的 $Logistic$ 的拉格朗日对偶问题为：

$$\min_{\alpha} D_{LR}(\alpha) = \frac{1}{2}\alpha^T Q \alpha + \sum_{i:\alpha_i > 0} \alpha_i \ln \alpha_i + \sum_{i:\alpha_i < C} (C - \alpha_i) \ln(C - \alpha_i)$$

其中 C 为惩罚因子，矩阵 Q 定义为：

$$Q_{ij} = y_i y_j x_i^T x_j$$

如果定义 $0 \log 0 = 0$，它与该极限是一致的：$\lim_{x \to 0^+} x \ln x = 0$。

上式可以简化为：

$$\begin{aligned} &\min_{\alpha} D_{LR}(\alpha) = \frac{1}{2}\alpha^T Q \alpha + \sum_{i=1}^{l} \alpha_i \ln \alpha_i + (C - \alpha_i) \ln(C - \alpha_i)\\ &0 \leqslant \alpha_i \leqslant C, i = 1, \cdots, l \end{aligned}$$

目标函数中带有对数函数可以采用坐标下降法求解，即每次只针对一个分量进行优化，而让其他的分量固定不定，算法依次优化每一个变量，直至收敛。与Lesson 3 优化方法基础中的其他最优化方法相比，坐标下降法有更快的迭代速度，更适合大规模问题的求解。

在用坐标下降法求解时，一个技巧是，不直接优化一个分量，而是优化该分量的增量，这和 $ResNet$ 中的残差思想其实是一致的。假设本次迭代时要优化 α_i ，其他的 $\alpha_j, j \neq i$ 固定不动。假设本次迭代后的 α_i 的值为 $\alpha_i + z$ ，即 $\alpha_{i+1} = \alpha_i + z$ ，则上式中的目标函数和不等式约束可以写成 z 的优化函数：

$$\min_z g(z) = (c_1 + z)\ln(c_1 + z) + (c_2 - z)\ln(c_2 - z) + \frac{a}{2}z^2 + bz$$
$$-c_1 \leqslant z \leqslant c_2$$

其中所有常数定义为：

$$c_1 = \alpha_i,\ c_2 = C - \alpha_i,\ a = Q_{ii},\ b = (Q\alpha)_i$$

因为目标函数有对数函数，该优化函数是一个超越函数，无法给出极值的解析解。因此，采用Lesson 3 优化方法基础的牛顿法解决上面问题，迭代公式为：

$$\begin{aligned}
&init\ z_0, k = 0\\
&while\ k < N:\\
&\quad if\ ||g'(x_k)|| < eps\ then\\
&\quad\quad break\\
&\quad end\ if\\
&\quad d_k = -\frac{g'(z_k)}{g''(z_k)}\\
&\quad z_{k+1} = z_k + d_k\\
&\quad k = k + 1\\
&end\ while
\end{aligned}$$

梯度为一阶导数，黑塞矩阵为二阶导数，在上面的算法中，它们分别为：

$$g'(z) = az + b + \ln\frac{c_1 + z}{c_2 - z}$$
$$g''(z) = a + \frac{c_1 + c_2}{(c_1 + z)(c_2 - z)}$$

为了保证牛顿法收敛，还需要利用Lesson 3 优化方法基础讲述的直线搜索技术，检查迭代之后的函数值是否充分下降。

5. $Softmax$ 回归

$Softmax$ 回归是 $Logistic$ 回归的拓展，实际上就是把逻辑回归的思想扩展到多分类问题上。它是一个多分类的算法，也常用来作为深度神经网络的最后一层分类层。

假设有 l 个训练样本 (x_i, y_i) ，其中 x_i 为 n 维特征向量，y_i 为类别标签，其取值为 $1-k$ 之间的整数。$Softmax$ 回归用下式来求样本 x 属于每个类别的概率：

$$h_\theta(x) = \frac{1}{\sum_{i=1}^{k} e^{\theta_i^T x}} \begin{pmatrix} e^{\theta_1^T x} \\ \vdots \\ e^{\theta_k^T x} \end{pmatrix}$$

其中，k 个向量 θ_i 为模型要学习的参数。简化概率表示，令 $S_i = \theta_i^T x$ ，得到 k 个标量，然后用 $Softmax$ 对它们归一化，就得到了概率值：

$$p_i = \frac{e^{S_i}}{\sum_k e^{S_k}}$$

模型的输出就是 $p_1 - p_k$ 组成的 k 维的概率向量 $\hat{y} = (p_i, \cdots, p_k)$，元素之和为 1 ，每个分量就是样本被判为该类的概率，是一个多项分布。使用指数变换是因为指数函数值大于 0 ，能保证概率非负。

最终分类就是求概率向量中，最大的概率对应的下标值，即： $y_{pred} = \arg\max(\hat{y})$ 。

5.1. 交叉熵损失的推导(最大似然估计)

$Softmax$ 回归要优化的参数为：

$$\theta = (\theta_1 \quad \theta_2 \quad \cdots \quad \theta_k)$$

θ_i 是一个列向量，因此 θ 是个 $n \times k$ 的矩阵。根据上面算法的定义，预测概率图 $\hat{y}$ 可以写为：

$$\hat{y} = \frac{1}{\sum\limits_{i=1}^{k} e^{\theta_i^T x}} \begin{pmatrix} e^{\theta_1^T x} \\ \vdots \\ e^{\theta_k^T x} \end{pmatrix}$$

训练集的真实标签向量 y 使用 $One-Hot$ 独热编码编码为向量。如果样本属于第 j 类，则向量的第 j 个分量为 1，其他为 0。比如说，如果 y 是第二类，则其表示为 $(0,1,0,\cdots,0)_{1\times k}$。

根据最大似然估计，样本的概率质量函数可以写成：

$$p_1^{y_1} \cdot p_2^{y_2} \cdots p_k^{y_k} = \prod_{i=1}^{k} (\hat{y}_i)^{y_i}$$

当矩阵 y 的取值确定属于第 j 类，即只有 y_j 为 1，其他都是 0，则上式的值为 $\hat{y}_j$。训练集的大小为 l，则似然函数可以写成：

$$J(\theta) = \prod_{i=1}^{l} \left(\prod_{j=1}^{k} \left(\frac{\exp(\theta_j^T x_i)}{\sum\limits_{t=1}^{k} \exp(\theta_t^T x_i)} \right)^{y_{ij}} \right)$$

其中 y_{ij} 为第 i 个训练样本标签向量的第 j 个分量。取对数得到对数似然函数为：

$$\ln J(\theta) = \sum_{i=1}^{l} \sum_{j=1}^{k} \left(y_{ij} \ln \frac{\exp(\theta_j^T x_i)}{\sum\limits_{t=1}^{k} \exp(\theta_t^T x_i)} \right)$$

$Softmax$ 回归目标是让这个对数似然函数极大化，转为最优化问题，等价于下面的损失函数取最小值：

$$L(\theta) = -\sum_{i=1}^{l} \sum_{j=1}^{k} \left(y_{ij} \ln \frac{\exp(\theta_j^T x_i)}{\sum\limits_{t=1}^{k} \exp(\theta_t^T x_i)} \right) = -\sum_{i=1}^{l} \sum_{j=1}^{k} \left(y_{ij} \ln \hat{y}_{ij} \right)$$

而对于单个样本 (x_i, y_i)，其损失为：

$$L(\theta) = -\sum_{j=1}^{k} y_j \ln \hat{y}_j$$

这就是多类别的交叉熵损失，反映了预测值 $\hat{y}$ 和标签值之间的差距，二者均为多项分布。在随机梯度下降法中，可以不对所有样本的交叉熵求和，而是对部分的样本求均值。

最大化对数似然函数就是最小化交叉熵损失函数，这也是 $Softmax$ 回归的目标，可以证明这个交叉熵损失函数是凸函数。

5.2. 交叉熵的随机梯度下降优化

对于单个样本 (x_i, y_i)，定义 k 为单个样本的特征维数，$\hat{y}_j$ 表示经过模型前向过程中，计算样本属于第 j 类的概率。

$$\hat{y}_j = \frac{\exp(\theta_j^T x_i)}{\sum\limits_{t=1}^{k} \exp(\theta_t^T x_i)}$$

假设样本 (x_i, y_i) 的实际标签是属于第 t 类，根据上文可以列出损失函数表达式：

$$L(\theta) = -\sum_{j=1}^{k} y_j \ln \hat{y}_j = -\ln \hat{y}_t$$

最后一步中去掉了 $\sum$，原因是根据独热编码，只有 y_t 标为 1，其余的 y_j 都是 0。

令 $s_j=\theta_j^Tx_i$ 简化计算，则 $\hat{y}_j=\frac{\exp(s_j)}{\sum_{t=1}^k\exp(s_t)}$ 。为了使用梯度下降，需要求 L 关于 θ 的偏导。

因此，将损失函数对 θ_j 求偏导，根据复合函数的偏导得到：

$$\frac{\partial L}{\partial\theta_j}=\frac{\partial L}{\partial\hat{y}_t}\frac{\partial\hat{y}_t}{\partial s_j}\frac{\partial s_j}{\partial\theta_j}=-\frac{1}{\hat{y}_k}\frac{\partial\hat{y}_t}{\partial s_j}x_i^T$$

因此只需要求解 $\frac{\partial\hat{y}_t}{\partial s_j}$ 即可，需要分情况讨论，因为只有 y 的第 t 个元素为 1，其他的都是 0 。

$$\hat{y}_t=\frac{\exp(s_t)}{\sum\limits_{t=1}^k\exp(s_t)}\quad\therefore\frac{\partial\hat{y}_t}{\partial s_j}=\frac{(e^{s_t})'\sum_k e^{s_t}-e^{s_t}(\sum_k e^{s_t})'}{(\sum_k e^{s_t})^2}$$

$$=\begin{cases}\frac{e^{s_j}\sum_k e^{s_t}-e^{s_j}e^{s_j}}{(\sum_k e^{s_t})^2}=\frac{e^{s_j}}{\sum_k e^{s_t}}-(\frac{e^{s_j}}{\sum_k e^{s_t}})^2=\hat{y}_j(1-\hat{y}_j) & j=t\\ \frac{0\sum_k e^{s_t}-e^{s_t}e^{s_j}}{(\sum_k e^{s_t})^2}=0-\frac{e^{s_j}}{\sum_k e^{s_t}}\frac{e^{s_t}}{\sum_k e^{s_t}}=-\hat{y}_j\hat{y}_t & j\neq t\end{cases}$$

所以最后得到：

$$\frac{\partial L}{\partial\theta_j}=\frac{\partial L}{\partial\hat{y}_t}\frac{\partial\hat{y}_t}{\partial s_j}\frac{\partial s_j}{\partial\theta_j}=-\frac{1}{\hat{y}_t}\frac{\partial\hat{y}_t}{\partial s_j}x_i^T=\begin{cases}(\hat{y}_j-1)x_i^T & j=t\\ \hat{y}_jx_i^T & j\neq t\end{cases}$$

即对于要求解的 θ_j 的梯度，其下标 j (对应的 $\hat{y}_j$)等于 y 对应类别标签的下标 t ，和 j 不等于对应类别标签 t 时，梯度是不一样的。这也就是多分类的交叉熵推导，其本质上衡量的就是，第 j 个样本计算得到的 $\hat{y}_j$，和实际标签下标 y_j 之间的差异。

算法过程可以表示为：首先使用独热编码，如果 y 属于第 t 类，则让矩阵下标为 t 的元素对应的值为 1 。之后正常使用随机梯度下降即可。

$$\begin{aligned}&One-Hot\ Encoding:init\ y=(0,\cdots,0,1_t,0,\cdots,0)\\&while(until\ \nabla L(w)=0\vee\text{迭代次数到 }n):\\&\quad for\ batch\ in\ dataloader(shuffle):\\&\qquad\nabla L(w)=\frac{2}{batch\_size}\sum_{i=1}^{batch\_size}\sum_{j=1}^{k}\begin{cases}(\hat{y}_j-1)x_i^T & j=t\\ \hat{y}_jx_i^T & j\neq t\end{cases}\\&\qquad w_{t+1}=w_t-\eta\cdot\nabla L(w)\end{aligned}$$

6. 贝叶斯决策($Bayes$)

它是一种基于统计学思想的方法，由未知向已知，由先验向后验转化的方法。

对于数据 x ，其最优的贝叶斯分类器为：

$$f^*(x)=\arg\max_{c\in[C]}p(c|x)$$

即预测具有最大条件概率的类别。p 是未知的，但我们可以估计它，朴素贝叶斯本质上也是一个参数密度估计问题。

朴素贝叶斯常用来处理离散型的数据。如果是连续型的数据，我们可以用 GMM 或者核密度估计 KDE 。

朴素贝叶斯思想： 先验概率 $+$ 数据 $=$ 后验概率 。

设 (Ω,F,P) 是一个概率空间，$B\in F$ ，且 $P(B)>0$ 。对 $\forall A\in F$ ，记：

$$P(B|A)=\frac{P(AB)}{P(A)}=\frac{P(A|B)P(B)}{P(A)}$$

这个公式可以看做，事件 B 是因，事件 A 是果。$P(B)$ 为先验概率($piror$)，是随机事件发生前观察/分析到的概率。$P(B|A)$ 为后验概率($posterior$)，事件发生后才知道。$P(A|B)$ 为似然函数。

如果事件 $A_1,\cdots,A_m$ 构成一个完备事件组，且 $P(A_i),P(B)>0$ ，则根据全概率公式和贝叶斯公式：

$$P(A_m|B)=\frac{P(B|A_m)P(A_m)}{\sum\limits_{i=1}^nP(A_i)P(B|A_i)}$$

根据贝叶斯公式和上述条件概率公式，例如：

$$P(Y=1|X_1=0,X_2=1,X_3=1)$$

$$=\frac{P(Y=1,X_1=0,X_2=1,X_3=1)}{P(X_1=0,X_2=1,X_3=1)}$$
$$=\frac{P(Y=1)P(X_1=0,X_2=1,X_3=1|Y=1)}{P(X_1=0,X_2=1,X_3=1)}$$

分母为全概率公式，分子为似然函数 × 先验概率。

朴素贝叶斯的朴素($Naive$)体现在算法使用**条件独立性假设**，即假设 $P(A|B,Y)=P(A|Y)$ ，A,B 关于事件 Y 独立，可记为 $A\perp B|Y$ ，则根据条件概率公式可得：

$$P(A,B|Y)=P(A|B,Y)P(B|Y)=P(A|Y)\times P(B|Y)$$
$$\text{即 } P(X|y)=\prod_{j=1}^{P}P(x_i|y)$$

因此原式 $P(Y=1|X_1=0,X_2=1,X_3=1)$ 为：

$$\frac{P(Y=1)P(X_1=0,X_2=1,X_3=1|Y=1)}{P(X_1=0,X_2=1,X_3=1)}$$
$$=\frac{P(Y=1)P(X_1=0|Y=1)P(X_2=1|Y=1)P(X_3=1|Y=1)}{P(X_1=0,X_2=1,X_3=1)}$$
$$=\frac{P(Y=1)P(X_1=0|Y=1)P(X_2=1|Y=1)P(X_3=1|Y=1)}{P(Y=1)P(X_1=0|Y=1)P(X_2=1|Y=1)P(X_3=1|Y=1)+P(Y=0)P(X_1=0|Y=0)P(X_2=1|Y=0)P(X_3=1|Y=0}$$

6.1. 朴素贝叶斯分类器原理

上面的贝叶斯公式可以变为如下形式：

$$p(y|x)=\frac{p(x|y)p(y)}{p(x)}$$

其中，x 为特征向量，y 为所属类别。$p(x)$ 表示特征向量的概率分布，$p(y)$ 表示每个类别出现的概率，即类先验概率。$p(x|y)$ 表示每个类样本的条件概率(类条件概率)，也被称为似然概率。通过这三个概率，就能计算出某样本 x_i 属于某类别 y_i 的后验概率 $p(y|x)$ 。

公式化定义：

设定输入空间为 $x\subseteq R^n$ 为 n 维向量的集合，输出空间为 $y=\{c_1,c_2,\cdots,c_k\}$ 一共 k 个类别。

训练数据集为：$T\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$ 。并根据先验分布和条件分布可得联合概率分布。

公式化推导：

$$P(Y=c_k|X=x)=\frac{P(X=x|Y=c_k)P(Y=c_k)}{P(X=x)}$$
$$\text{其中，}P(X=x)=\sum_k P(Y=c_k)\sum_j P(X^{(j)}|Y=c_k)$$
$$\text{其中，}P(X=x|Y=c_k)=\sum_j P(X^{(j)}|Y=c_k)$$
$$\text{带入得，}P(Y=c_k|X=x)=\frac{P(Y=c_k)\sum_j P(X^{(j)}|Y=c_k)}{\sum_k P(Y=c_k)\sum_j P(X^{(j)}|Y=c_k)}$$

目标函数可定义为：

$$y=\arg\max_{c_k}P(Y=c_k|X=x)$$
$$=\arg\max_{c_k}\frac{P(Y=c_k)\prod_j P(X^{(j)}=x^j|Y=c_k)}{\sum_k P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)}|Y=c_k)}$$

对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率。哪个最大，就认为此分类项属于哪个类别。所以可以忽略分母的全概率项 ，简化判别函数。

所以原目标函数等价为：

$$y = \arg\max_{c_k} P(Y = c_k) \prod_j P(X^{(j)} = x^j | Y = c_k)$$

通过上面推导也能看出，实现贝叶斯分类器需要知道每类样本的特征向量所服从的概率分布，只要知道样本的概率分布，则可以通过贝叶斯分类器求解。

比如说，如果样本分布服从正态分布，则该朴素贝叶斯分类器被称为高斯贝叶斯分类器。

6.2. 最大化后验概率

Y 为标签，f 为模型，X 为特征。

假设 $0-1$ 损失函数：

$$L(Y, f(X)) = \begin{cases} 1, Y \neq f(X) \\ 0, Y = f(X) \end{cases}$$

则期望损失函数为(希望尽可能对，故希望 $R_{\exp}$ 越小越好)：

$$\begin{aligned} R_{\exp}(f) &= E[L(Y, f(X))] \\ &= E_X \sum_{k=1}^{K} [L(Y, f(X))] P(c_k|X) \\ &= \int_{X \times Y} L(y, f(x)) P(x, y) \mathrm{d}x \mathrm{d}y \\ &= \int_{X \times Y} L(y, f(x) P(y|x) \mathrm{d}y \, P(X) \mathrm{d}x \\ &= \int_X \int_Y L(y, f(x) P(y|x)) \mathrm{d}y \, P(X) \mathrm{d}x \end{aligned}$$

后验概率最大化推导：

$$\begin{aligned} f(x) &= \arg\max_{y \in Y} \sum_{k=1}^{K} L(c_k, y) P(c_k | X = x) \\ &= \arg\min_{y \in Y} \sum_{k=1}^{K} P(y \neq c_k | X = x) \\ &= \arg\min_{y \in Y} (1 - P(y = c_k | X = x)) \\ &= \arg\max_{y \in Y} P(y = c_k | X = x) \end{aligned}$$

算法流程：

- 计算先验概率及其条件概率。

$$\begin{aligned} P(Y = c_k) &= \frac{\sum_{i=1}^{N} l(y_i = c_k)}{N}, \quad k = 1, 2, \cdots, K \\ P(X^{(j)} = x^{(j)} | Y = c_k) &= \frac{\sum_{i=1}^{N} l(X^{(j)} = x^j | Y = c_k)}{\sum_{i=1}^{N} l(Y = c_k)} \end{aligned}$$

- 对于给定的实例 $x = (x^{(1)}, x^{(2)}, \cdots x^{(n)})^T$，计算目标函数的等价概率积

$$P(Y = c_k) \prod_j P(X^{(j)} = x^{(j)} | Y = c_k), \quad k = 1, 2, \cdots, K$$

- 确定实例 x 的类别。

$$y = \arg\max_{c_k} P(Y = c_k) \prod_j P(X^{(j)} = x^{(j)} | Y = c_k)$$

性能分析：

优点：对小规模的数据表现很好，适合多分类任务，适合增量式训练。

缺点：对输入数据的表达形式很敏感。

6.3. 新冠患者死亡率预测

从朴素贝叶斯角度出发分析...

6.4. 高斯贝叶斯分类模型

假设每个类的样本的 n 维特征向量 x 都服从正态分布，此时的类条件概率密度函数为：

一维：

$$p(x|c) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

多维：

$$p(x|c) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_c|^{\frac{1}{2}}}\exp\Big(-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)\Big)$$

其中 $c = 1, 2, \cdots, k$ 为类别标签，μ_c 为类别标签 c 的均值向量，$\sum_c$ 为协方差矩阵，用每个类的训练样本通过最大似然得到。

$$p(c|x) = \frac{p(c)p(x|c)}{p(x)}$$

根据朴素贝叶斯中的推导，实际分类器为 $\arg\max_c p(c)p(x|c)$ 。如果每个类出现的概率 $p(c)$ 相等，则该分类器简化为：

$$\arg\max_c p(x|c)$$

即计算每个类的改密函数值 $p(x|c)$ 并取其中极大值对应的类，作为分类结果。

接下来进一步简化，对 $p(x|c)$ 取对数，并变形后得到：

$$\begin{aligned}\ln(p(x|c)) &= \ln\Big(\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_c|^{\frac{1}{2}}}\Big) - \frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c) \\ &= -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_c| - \frac{1}{2}((x-\mu_c)\Sigma_c^{-1}(x-\mu_c))\end{aligned}$$

其中，$-\frac{n}{2}\ln(2\pi)$ 为一个常数。原问题 $\arg\max_c p(x|c)$ 实际上变为求下面的极小值：

$$\arg\min_c \frac{1}{2}\ln|\Sigma_c| + \frac{1}{2}((x-\mu_c)\Sigma_c^{-1}(x-\mu_c))$$

该值最小的哪一类即为最后的分类结果。