第三章　线性模型

0.1　线性回归

0.1.1　最小二乘法 (least square method)

基于均方误差 (mean square error, MSE) 最小化的模型求解方法。均方误差定义为:

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}(y_i-f(x_i))^2=\frac{1}{m}\sum_{i=1}^{m}(y_i-wx_i-b)^2$$

通过对 E 分别求 w 和 b 的偏导，并令其等于 0 可得:

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

$$b=\frac{1}{m}\sum_{i=1}^{m}(y_i-wx_i)$$

其中 $\bar{x}=\frac{1}{m}\sum_{i=1}^{m}x_i$ 为 x 的均值。

为什么可以这样求解呢? 因为损失函数是一个**凸函数**（是向下凸，类似 U 型曲线），导数为 0 表示该函数曲线最低的一点，此时对应的参数值就是能使均方误差最小的参数值。特别地，要判断一个函数是否凸函数，可以求其**二阶导数**，若二阶导数在区间上**非负**则称其为凸函数，若在区间上恒大于零则称其为严格凸函数。

0.1.2　多元线性回归

令 $\hat{\mathbf{w}}=(\mathbf{w};b)$。把数据集表示为 $m\times(d+1)$ 大小的矩阵，每一行对应一个样例，前 d 列是样例的 d 个属性，**最后一列恒置为 1**，对应偏置项。把样例的真实标记也写作向量形式，记作 $\mathbf{y}$。则此时损失函数为:

$$E_{\hat{\mathbf{w}}}=(\mathbf{y}-X\hat{\mathbf{w}})^T(\mathbf{y}-X\hat{\mathbf{w}})$$

同样使用最小二乘法进行参数估计，首先对 $\hat{\mathbf{w}}$ 求导:

$$\begin{aligned}
dE_{\hat{\mathbf{w}}}&=(-Xd\hat{\mathbf{w}})^T(\mathbf{y}-X\hat{\mathbf{w}})+(\mathbf{y}-X\hat{\mathbf{w}})^T(-Xd\hat{\mathbf{w}})\\
&=(Xd\hat{\mathbf{w}})^T(X\hat{\mathbf{w}}-\mathbf{y})+(X\hat{\mathbf{w}}-\mathbf{y})^TXd\hat{\mathbf{w}}\\
&=(X\hat{\mathbf{w}}-\mathbf{y})^TXd\hat{\mathbf{w}}+(X\hat{\mathbf{w}}-\mathbf{y})^TXd\hat{\mathbf{w}}\\
&=2(X\hat{\mathbf{w}}-\mathbf{y})^TXd\hat{\mathbf{w}}\qquad(this\ is\ a\ scalar)\\
&=tr\left((\frac{\partial E_{\hat{\mathbf{w}}}}{\partial\hat{\mathbf{w}}})^Td\hat{\mathbf{w}}\right)\\
&=(\frac{\partial E_{\hat{\mathbf{w}}}}{\partial\hat{\mathbf{w}}})^Td\hat{\mathbf{w}}
\end{aligned}$$

所以有:

$$\frac{\partial E_{\hat{\mathbf{w}}}}{\partial\hat{\mathbf{w}}}=2X^T(X\hat{\mathbf{w}}-\mathbf{y})$$

当 X^TX 是可逆矩阵，也即**满秩矩阵**（full-rank matrix）或正定矩阵（positive definite matrix）时，令该式值为 0 可得到 $\hat{\mathbf{w}}$ 的闭式解：

$$\hat{\mathbf{w}}* = (X^TX)^{-1}X^T\mathbf{y}$$

现实任务中 X^TX 往往不是满秩的，常见的做法是引入**正则化（regularization）**项。

0.2 对数几率回归（逻辑回归）

对数几率函数（logistic function）：

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

该式可以改写为：

$$\ln \frac{y}{1-y} = \mathbf{w^T x} + b$$

其中，$\frac{y}{1-y}$ 称作几率（odds），把 y 理解为该样本是正例的概率，把 $1-y$ 理解为该样本是反例的概率，而几率表示的就是**该样本作为正例的相对可能性**。若几率大于 1，则表明该样本更可能是正例。对几率取对数就得到对数几率（log odds，也称为 logit）。几率大于 1 时，对数几率是正数。

0.2.1 极大似然法（maximum likelihood method）

所谓极大似然，就是最大化预测事件发生的概率，也即**最大化所有样本的预测概率之积**。令 $p(c = 1|\mathbf{x})$ 和 $p(c = 0|\mathbf{x})$ 分别代表 y 和 $1-y$。简单变换一下公式，可以得到：

$$p(c = 1|\mathbf{x}) = \frac{e^{\mathbf{w^T x}+b}}{1 + e^{\mathbf{w^T x}+b}}$$

$$p(c = 0|\mathbf{x}) = \frac{1}{1 + e^{\mathbf{w^T x}+b}}$$

由于预测概率都是小于 1 的，如果直接对所有样本的预测概率求积，所得的数会非常非常小，当样例数较多时，会超出精度限制。所以，一般来说会对概率去对数，得到**对数似然（log-likelihood）**，此时求所有样本的预测概率之积就变成了求所有样本的对数似然之和 *。对率回归模型的目标就是最大化对数似然，对应的似然函数是：

$$\ell(\mathbf{w}, b) = \sum_{i=1}^{m} \ln p(c_i|\mathbf{x_i}; \mathbf{w}; b) = \sum_{i=1}^{m} \ln(c_i p_1(\hat{\mathbf{x}}_\mathbf{i}; \beta) + (1 - c_i)p_0(\hat{\mathbf{x}}_\mathbf{i}; \beta))$$

可以理解为若标记为正例，则加上预测为正例的概率，否则加上预测为反例的概率。其中 $\beta = (\mathbf{w}; b)$。

对该式求导，令导数为 0 可以求出参数的最优解。特别地，此时发现似然函数的导数和损失函数是等价的，所以说**最大似然解等价于最小二乘解**。最大化似然函数等价于最小化损失函数：

$$E(\beta) = \sum_{i=1}^{m}(-y_i\beta^T\hat{x}_i + \ln(1 + e^{\beta^T\hat{\mathbf{x}}_\mathbf{i}}))$$

这是一个关于 β 的高阶可导连续凸函数，可以用最小二乘求（要求矩阵的逆，计算开销较大），也可以用数值优化算法如**梯度下降法**（gradient descent method）、**牛顿法**（Newton method）等逐步迭代来求最优解（可能陷入局部最优解）。

0.3 线性判别分析

在线性判别分析（Linear Discriminant Analysis，简称 LDA）中，不再是拟合数据分布的曲线，而是将所有的数据点投影到一条直线上，使得**同类点的投影尽可能近，不同类点的投影尽可能远**。

同类样例的投影值尽可能相近意味着**同类样例投影值的协方差应尽可能小**；然后，异类样例的投影值尽可能远离意味着**异类样例投影值的中心应尽可能大**。合起来，就等价于最大化：

$$J = \frac{\|\mathbf{w}^T\mu_0 - \mathbf{w}^T\mu_1\|_2^2}{\mathbf{w}^T\Sigma_0\mathbf{w} + \mathbf{w}^T\Sigma_1\mathbf{w}} = \frac{\mathbf{w}^T(\mu_0 - \mu_1)(\mu_0 - \mu_1)^T\mathbf{w}}{\mathbf{w}^T(\Sigma_0 + \Sigma_1)\mathbf{w}}$$

其中，分子的 μ_i 表示第 i 类样例的均值向量 * 即表示为向量形式后对各维求均值所得的向量）。分子表示的是两类样例的均值向量投影点（也即类中心）之差的 ℓ_2 范数的平方，这个值越大越好。分母中的 Σ_i 表示第 i 类样例的**协方差矩阵**。分母表示两类样例投影后的协方差之和，这个值越小越好。

LDA 可从贝叶斯决策理论的角度阐释，并可证明：当两类数据同先验，满足高斯分布且协方差相等时，LDA 可达到最优分类。

0.4 多分类学习

基于一些策略，把多分类任务分解为多个二分类任务，利用二分类模型来解决问题。有三种最经典的拆分策略，分别是一对一，一对其余，和多对多。

一对一（One vs. One，简称 OvO）的意思是把所有类别两两配对。假设样例有 N 个类别，OvO 会产生 $\frac{N(N-1)}{2}$ 个子任务，每个子任务只使用两个类别的样例，并产生一个对应的二分类模型。测试时，新样本输入到这些模型，产生 $\frac{N(N-1)}{2}$ 个分类结果，最终预测的标记由投票产生，也即把被预测得最多的类别作为该样本的类别。

一对其余（One vs. Rest，简称 OvR）产生 N 个二分类模型，测试时，新样本输入到这些模型，产生 N 个分类结果，若只有一个模型预测为正例，则对应的类别就是该样本的类别；若有多个模型预测为正例，则选择置信度最大的类别（参考模型评估与选择中的**比较检验**）。

多对多（Many vs. Many，简称 MvM）是每次将多个类作为正例，其他的多个类作为反例。OvO 和 OvR 都是 MvM 的特例。书中介绍的是一种比较常用的 MvM 技术——**纠错输出码**（Error Correcting Outputs Codes，简称 ECOC）。

MvM 的正反例划分不是任意的，必须有特殊的构造，否则组合起来时可能就无法定位到预测为哪一类了。ECOC 的工作过程分两步：

- 编码：对应于训练。假设有 N 个类别，计划做 M 次划分，每次划分把一部分类别划为正类，一部分类别划分为反类，最终训练出 M 个模型。而每个类别在 M 次划分中，被划为正类则记作 +1，被划为负类则记作-1，于是可以表示为一个 M 维的编码。
- 解码：对应于预测。把新样本输入 M 个模型，所得的 M 个预测结果组成一个预测编码。把这个预测编码和各个类别的编码进行比较，跟哪个类别的编码**距离**最近就预测为哪个类别。

此处的距离有汉明距离，欧式距离等。

为什么称这种方法为**纠错**输出码呢？因为 ECOC 编码对分类器的错误有一定的容忍和修正能力。即使预测时某个分类器预测成了错误的编码，在解码时仍然有机会产生正确的最终结果。具体来说，对同一个学习任务，**编码越长，纠错能力越强**。但是相应地也需要训练更多分类器，增大了计算和存储的开销。对同等长度的编码来说，理论上任意两个类别之间的编码距离越远，纠错能力越强。

0.5 类别不平衡问题

欠采样（undersampling）针对的是负类，也即移取训练集的部分反例，使得正类和负类的样例数目相当。由于丢掉了大量反例，所以时间开销也大大减少。但是带来一个问题就是，随机丢弃反例可能会丢失一些重要信息。书中提到一种解决方法是利用**集成学习机制**，将反例划分为多个集合，用于训练不同的模型，从而使得对每个模型来说都进行了欠采样，但全局上并无丢失重要信息。

过采样（oversampling）针对的是正类，也即增加训练集的正例，使得正类和负类的样例数目相当。过采样的时间开销会增大很多，因为需要引入很多正例。注意！过采样不能简单地通过重复正例来增加正例的比例，这样会引起严重的过拟合问题。一种较为常见的做法是对已有正例进行**插值**来产生新的正例。

阈值移动（threshold-moving）利用的是**再缩放**思想。前面对数几率回归中，几率 $\frac{y}{1-y}$ 表示正例的相对可能性，默认以 1 作为阈值，其实是假设了样本的真实分布为正例反例各一半。但这可能不是真相，假设我们有一个存在类别不平衡问题的训练集，正例数目为 m^+，反例数目为 m^-，可以重定义：

$$\frac{y'}{1-y'} = \frac{y}{1-y} \times \frac{m^-}{m^+}$$

这就是再缩放（rescaling）。当几率大于 $\frac{m^+}{m^-}$ 时就预测为正例。但必须注意，这种思想是基于观测几率近似真实几率这一假设的，现实任务中这一点未必成立。